Errata 4 to TS 102 796 V1.4.1 and Errata 1 to TS 102 796 V1.5.1

2020-07-31

Table of Contents

1 Introduction

This document contains the currently identified and resolved errata to ETSI TS 102 796 v1.4.1 and V1.5.1. It is a living document which will be updated based on experience of implementing receivers, services and tests. Versions of this document will periodically be made publicly available via the www.hbbtv.org web site.

The contents of this document will be included in subsequent errata documents making this document obselete.

Feedback is welcome. HbbTV members should provide this through the HbbTV internal issue tracking system. Non-members may provide it by email to info@hbbtv.org.

For avoidance of doubt, the contents of this document have not been reviewed or approved by ETSI.

2 Conventions

In this document, text quoted from other documents or to be added to other documents is indented except where it appears in a table. Fine-grained changes in text from other documents are shown using the underline and strikethrough convention.

3 Summary

The following table summarises the changes included in this document that shall be applied to TS 102 796 V1.4.1. These changes were included in TS 102 796 V1.5.1 during the process of creating that document.

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
3535, 8576	XML parsing	Other	2.1, 7.2.3.2, 7.2.4, 8.2.1.1, 10.2.4, A.1, A.2.6.2, E.2.1	4.2.4, 4.5.5, 4.6.4, 4.8.2, 4.13.5
4913	Clarify resource usage by HTML5 media elements	Ambiguity	9.6.2, A.2.1	4.7.2, 4.13.3
5097	Frame rate support with UHD content via DASH	Ambiguity	New clause E.4.6	4.15.2
5307	Clarifying media synchroniser error codes	Ambiguity	8.2.3.2	4.6.2
5314	Update inter-device sync spec from DVB blue book back to ETSI	Other	2.1, 8.2.3, 10.2.8, 13, G.1	4.2.1, 4.6.5, 4.8.4, 4.11.3, 4.16.1
5324	Clarify URL scheme and MPD anchor rules	Ambiguity	9.2, E.4.5	4.7.1, 4.15.1

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
5385	Removing support for CSS3 navigation	Other	10.2.2.1, new annex L	4.8.1, 4.18.1
5499	Timing requirements for end of mid roll adverts	Ambiguity	New clause 9.6.12	4.7.4
5500	Confusion over deviceId class	Ambiguity	12.1.5	4.10.1
5575	Running apps from previous spec versions	Editorial	1	4.1.1
5582	Two setChannel methods of video/broadcast defined with same signature	Error	A.2.4.7.4	4.13.10
5583	Ignoring unsupported AIT descriptors	Ambiguity	2.2, 7.2.3.1	4.5.4, 4.2.3
5639	Issue with companion launcher API	Error	8.2.6.1	4.6.3
5738	errorCode(s) for onCSLaunch	Ambiguity	8.2.6.1	4.6.7
5824	Key "Label" should be "label"	Ambiguity	A.2.5.3	4.13.7
5862	Running a regular HbbTV app on a channel not in the terminal channel list	Editorial	6.2.2	4.4.4
5864	text in wrong location in annex A - referring to MetadataSearch	Editorial	A.1	4.13.1
5872	Clarification to 6.2.2.9 re key events	Ambiguity	6.2.2.9	4.4.2
5899	Reference to NI clause 10.1.1 of the DAE specification	Inconsistency	A.1	4.13.16
5910	UPnP version to be used with DIAL	Implementation	2.1, 14.7	4.2.2, 4.12.1
5919	VK_RECORD key is not in any set of keys	Error	A.1	4.13.4
5972	PVR API issues	Error	A.1, A.2.4.7.3, A.2.4.8.1, A.2.4.8.2	4.13.11
5984	Scope of AVAudio and AVVideoComponents	Ambiguity	A.2.4.6	4.13.12
6047	Update key event text in clause 5.2	Editorial	5.2	4.3.1
6049	Avoid MediaSynchroniser API leaking "secret" stream URLs	Other	8.2.3.2.1, 13.6.2, 13.8.2.2	4.6.1, 4.11.1

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
6211	video/broadcast object issues when playing A/V from broadband	Ambiguity	6.2.2.7, A.2.4.1, A.2.4.5, new clause A.2.26	4.4.3, 4.13.8
6215, 8576	Content supporting both Clear Key and conventional DRM	Ambiguity	9.6.7, A.2.27, B.3, new clause B.4	4.7.6, 4.13.13, 4.14.1
6422	Clarification of TEMI timeline requirements	Ambiguity	13.4.2	4.11.2
6480	Keeping "supportsPointer" read only and false for not-activated applications	Ambiguty	10.2.2.2	4.8.7
6482	Incoherent AVSubtitleComponent language with TTML tracks	Inconsistency	A.1	4.13.17
6495	Additional cases when 250ms ad insertion transition may not be possible to meet	Other	9.6.3	4.7.3
6498	Subtitle track selection - ability for an app to suppress subtitles reliably	Ambiguity	10.2.7.1, 10.2.7.3	4.8.3
6544	Missing XML declaration in example XML AIT	Error	7.2.3.2	4.5.3
6550	Clarification on XML AIT example	Error	7.2.3.2	4.5.2, 4.12.2
6560	Channel change key behaviour with broadcast independent applications	Ambiguity	6.2.2.2	4.4.1
6564	bindToCurrentChannel() while an A/V Control object is presenting	Ambiguity	A.2.1	4.13.6
6591	"channels" should be "audioChannels"	Editorial	A.2.4.6	4.13.2
6610	Update reference to TS 102 809	Other	2.1, 7.2, 11.1	4.2.1, 4.5.1, 4.9.1
6741	Unclear language in A 2.20.1 - Extensions to Represent Subtitle Presentation	Ambiguity	A.2.20.1	4.13.14
6914	URL used in in DIAL example should be changed	Other	14.7.3.1, 14.7.3.2	4.12.3
6915	Lack of linkage between spec version in AIT signalling and version in DOCTYPE in	Other	7.2.3.1, A.2.6.2	4.5.6, 4.13.9

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
	individual pages of app			
6971	Ambiguous use of "this"	Ambiguity	7.3.2.6	4.5.8
7045	Including or not including data services in the service list based on HbbTV version	Ambiguity	7.2.6	4.5.7
7046	Require the same DASH player for HTML5 video element and A/V control object?	Other	New clause 9.1.1.3	4.7.5
7088	Allow applications to prioritise DRM to use	Other	A.1, new clause A.2.27	4.13.13
7128	Clarification on FSA: Is group priority value 0x00 valid?	Ambiguity	7.2.7.2	4.5.9
7146	Pending requests for FSA files when carousel is unmounted	Ambiguity	New clause 7.2.7.5	4.5.11
7162	SHA-1 sunset date has passed	Other	11.2.4	4.9.2
7196, 7697	Carousel access following channel change	Ambiguity	7.2.5.3	4.5.10
7323	Web Storage following channel change	Other	6.3.2	4.4.5
7338	Cookie writing to persistent storage time requirement	Ambiguity	10.2.1	4.8.5
7364	Clarification on value for property "lastError" if no error occured	Ambiguity	8.2.3.2.1	4.6.6
7439	Clarification to parental rating	Ambiguity	A.1, new clause A.2.28	4.13.15
7598	deviceID	Other	12.1.5, A.2.20.5	4.10.1, 4.13.20
7607	Encrypted Media Extensions Recommendation	Other	2.1	4.2.7
7636	Clarify what timelineSelector to convey in CII message	Ambiguity	13.6.2	4.11.4
7775	Web Storage minimum storage requirement	Ambiguity	10.2.1	4.8.6
7801	Errors in mapping from DASH roles to HTML5 audio/video track kind values	Error	A.2.12.3	4.13.19

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
7970	Approval and pre-approval to launch an HbbTV app on request by a CS app	Ambiguity	14.6.2	4.12.4
8015	Clarifications relating to origins for application boundary	Ambiguity	6.3.2, 6.3.3	4.4.7
8029	Possible ambiguity in table A.1 re subtitlesEnabled	Ambiguity	A.1	4.13.18
8038	WebStorage behaviour if user has disabled persistent storage	Other	12.1.4	4.10.2
8049	URL prefixes for DVB URLs in App Boundary Descriptor	Error	7.2.3.1, 7.2.3.2	4.5.12
8068	HTML5 media element audio resume position	Error	9.6.3	4.7.8
8070	RFC for web origin allows it to be null in "privacy-sensitive" contexts	Other	2.1, 6.3.2	4.2.6, 4.4.6
8074	No requirements on the DASH player to adapt	Ambiguity	New clause E.4.7	4.15.3
8103	Media synchronization informative examples don't construct CorrelationTimestamp objects	Ambiguity	8.2.3.3.1	4.6.8
8119	Use of current.ait for starting a b-r app in the time between successful service selection and AIT acquisition	Ambiguity	6.2.2.6.1, 9.2	4.4.8, 4.7.7
8158	Issues with HbbTV apps selecting inband TTML subtitles	Ambiguity	10.2.7.3	4.8.8
8184	Warning to app developers that some values/properties are insecure	Other	New clause A.2.29, A.1, 8.2.1.2, 8.2.2	4.13.21, 4.6.9
8232	Broken cross-reference to 14.5.6	Editorial	14.5.1	4.12.5

The following table summarises the changes included in this document that shall be applied to both TS 102 796 V1.4.1 and TS 102 796 V1.5.1.

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
5062	Undefined MediaSynchroniser behaviour for HTML5 media elements if source is changed	Ambiguity	9.7.1.2	4.7.16
6566	DSM-CC cache validity	Other	7.2.2	4.5.16
7047	Minimum performance requirements for DSM-CC OC	Other	New Clause 7.2.5.5	4.5.13
7893	Equivalence between an InbandEventStream and an EventStream	Ambiguity	9.3.2.2	4.7.13
8322	Clarification of when the Media Synchroniser raises error code 18	Ambiguity	8.2.3.2.2	4.6.10
8612	selectComponent/unselectComponent when v/b playstate is other than PRESENTING	Ambiguity	A.2.4.5	4.13.31
8681	HBBTV and W3C/DASH-IF text track kind calculation inconsistency	Other	A.2.12.3	4.13.24
8712	Security level for videobroadcast currentChannel property : remove Broadcast-related	Error	A.1	4.13.22
8714	Alpha blending in HbbTV	Other	2.2, 10.1	4.2.8, 4.8.9
8718	initMediaSynchroniser	Error	8.2.3.2.2, 13.11.3	4.6.10, 4.11.5
8810 10719 10722	behaviour of multi-stream sync API at times when no content exists in a slave media synchroniser stopping multi-stream sync and disposing of a MediaSynchroniser object successful completion of initMediaSynchroniser and addMediaObject	Ambiguity, Error	8.2.3 9.7.1.2 10.2.7.1 13.3.1 13.3.8 new Annex M	4.6.11 4.7.17 4.8.11 4.11.6 4.19
8859	Invalid reference	Editorial	A.2.20.5	4.13.23
9285	OSD display when returning to broadcast from broadband	Other	A.2.4.5	4.13.29
9288	Crashes after a number of transitions from broadcast to broadband and back again	Other	9.8	4.7.12

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
9290	Processing time for stream events varies a lot (even on the same device)	Editorial	7.2.4	4.5.15
9298 9497	Reconciling user understanding of what terminal model they have with entries in a broadcaster log file, Clarify interpretation of family name in HTTP UA	Other	7.3.2.4	4.5.17
9309	setActiveDRM : need to specify how/when the default behavior is restored	Ambiguity	A.2.27	4.13.27
9315	DASH - MPD events	Ambiguity	2.1, 2.2 and 9.3.2.2	4.2.11 and 4.7.9
9325	Unreasonably demanding a/v sync timing requirement	Other	9.7.4	4.7.11
9445	Behaviour of html5 media when HbbTV app is hidden	Ambiguity	A.2.14	4.13.28
9481, 9623 10181 10195	potential conflict between HbbTV 9.6.2 and HTML5 re HTML5 load() method taking video & audio decoders hardware video resource management and HTML5 video elements Releasing resources from HTML media elements media decoders and the seeked event	Ambiguity	9.6.2	4.7.10
9487 10141	error in XML capabilities example and XSD A.2.1 XML Namespace issues	Editorial / Error	10.2.4 A.2.15	4.8.12 4.13.34
9509	Confusing language added to 6.2.2.2 re channel changing mechanisms	Editorial	6.2.2.2	4.4.9
9629	firing playStateChange when the state does not change	Ambiguity	A.2.4.1	4.13.30
9840	Clarification required on expected state when video/broadcast object stops	Ambiguity	A.2.4.1	4.13.26
10007	Incompatibility with modern soft input (virtual keyboards) that operate on words or phrases	Other	10.2.1	4.8.10
10230	Inconsistency on support for encrypted	Ambiguity	11.5	4.9.3

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
	content over broadband			
10256	Incorrect reference to organisation_id definition	Editorial	H.4.2	4.17.1
10439	AdaptationSet@contentType attribute optionality	Ambiguity	New clause E.2.7.	4.15.6
10447	Errata to DVB-DASH in HbbTV 2.0.1/2	Ambiguity	2.1, E.4.1	4.2.10 and
10455	Additional web security APIs	Other	2.2 and new clause A.3.14	4.2.9 and 4.13.25
10531	TTML, mismatch between HbbTV and ETSI EN 303 560	Other	7.3.1.5.1	4.5.18
10535	Mixing audio codecs within a single AdaptationSet	Ambiguity	E.3.2	4.15.4
10649	Multiple video decoders with differing capabilities	Other	9.6.2	4.7.14
10723	Consistency between HbbTV-TA and HbbTV 2.0.1/2/3 re multiple decoders and ad insertion/replacement	Other	9.6.3	4.7.15
10731	Backporting XML AIT schema changes from OpApp into core spec	Other	7.2.3.2	4.5.19
10748	Distinctive identifiers: requestAccessToDeviceID method	Error	12.1.5	4.10.3
10797	bindToCurrentChannel from STOPPED state when resources are not available	Ambiguity	A.2.4.1	4.13.32
10876	AudioTrack and VideoTrack creation with EssentialProperty descriptors	Ambiguity	A.2.12.1	4.13.33

The following table summarises the changes included in this document that shall be applied to TS 102 796 V1.5.1 and that are not applicable to TS 102 796 V1.4.1..

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
#10451	Table reference in clause 7.3.1.1	Editorial	7.3.1.1	4.5.14

Key to categories

Ambiguity	Feature where different implementations may behave in different ways. This includes under-specified features as well as inconsistencies within the specification.
Editorial	Purely editorial change
Error	Clear technical error in the specification. Cannot be implemented as written.
Implementation	Feature removed, simplified or modified in order to simplify implementation and testing.

4 Changes to TS 102 796 v1.4.1

4.1 Clause 1 – Scope

4.1.1 Running apps from previous spec versions

In the following paragraph, "the two" is removed as shown with strike-through notation;

> The present document requires terminals to run applications signalled as conforming to ~~the two~~ previous revisions. This allows for smooth transitions where the previous revisions have been deployed.

4.2 Clause 2 – References

4.2.1 Update reference to TS 102 809

Normative reference 3, TS 102 809, is updated from version 1.2.1 to version 1.3.1.

4.2.2 UPnP version to be used with DIAL

The following normative references are added.

[67]   ISO/IEC 29341-1:2011: "Information technology -- UPnP Device Architecture -- Part 1: UPnP Device Architecture Version 1.0".

NOTE: This specification was first published by UPnP™ in 2008, and an equivalent version is available from http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0.pdf

[68]   ISO/IEC 29341-1-1:2011: Information technology -- UPnP Device Architecture -- Part 1-1: UPnP Device Architecture Version 1.1".

NOTE: This specification was first published by UPnP™ in 2008, and an equivalent version is available from http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf

4.2.3 Ignoring unsupported AIT descriptors

The following additional informative reference is added:

[i.21]   DVB Services, "MHP & GEM | MHP AIT Descriptor"

NOTE: Located at http://www.dvbservices.com/identifiers/mhp_ait_descriptor

4.2.4   XML Parsing

The following normative reference is added;

> [69] W3C Recommendation (4 February 2004): "Extensible Markup Language (XML) 1.0 (Third Edition)"

> NOTE: Available at http://www.w3.org/TR/2004/REC-xml-20040204/

4.2.5   Update inter-device sync spec from DVB blue book back to ETSI

In clause 2.1, reference 47 is changed from:

> DVB Bluebook A167-2 [06/2016]: "Digital Video Broadcasting (DVB); Companion Screens and Streams; Part 2: Content Identification and Media Synchronisation".

> NOTE: Available at http://www.dvb.org/resources/public/standards/a167-2_dvb-css_part_2_content-ID-and-media-sync_June2016.pdf

To

> ETSI TS 103 286-2 (V1.2.1): "Digital Video Broadcasting (DVB); Companion Screens and Streams; Part 2: Content Identification and Media Synchronisation".

4.2.6   RFC for web origin allows it to be null in "privacy-sensitive" contexts

In clause 2.1, reference 25 is changed to:

> IETF RFC 6454: "The Web Origin Concept"

4.2.7   Encrypted Media Extensions Recommendation

In clause 2.1, reference 66 is changed from

> W3C Working Draft (04 February 2016) : "Encrypted Media Extensions".

to

> W3C Recommendation (18 September 2017) : "Encrypted Media Extensions".

4.2.8   Alpha blending in HbbTV

In clause 2.2, the following informative references are added.

> [i.22]   Tantek Çelik; Chris Lilley; L. David Baron. CSS Color Module Level 3. 07 June 2011. W3C Recommendation. URL: http://www.w3.org/TR/2011/REC-css3-color-20110607/

> [i.23]   Rik Cabanier; Nikos Andronikos. Compositing and Blending Level 1. 13 January 2015. W3C Candidate Recommendation. URL: https://www.w3.org/TR/compositing-1/

4.2.9   Web security improvements

In clause 2.2, the following informative reference is added.

> [i.24]   WHATWG: "Fetch Living Standard"

NOTE: Located at https://fetch.spec.whatwg.org/

4.2.10 Errata to DVB-DASH in HbbTV 2.0.1/2

In clause 2.1, the following changes are made;

- The reference to MPEG DASH is changed from 2014 (2nd Edition) to 2019 (4th Edition).

- The reference to DVB-DASH / TS 103 285 is changed from 1.2.1 to 1.3.1

4.2.11 DASH - MPD events

In clause 2.1, the following normative references are added.

[1]   IETF RFC 4648 "The Base16, Base32, and Base64 Data Encodings"

[2]   W3C Recommendation (2 May 2008): "Canonical XML Version 1.1"

NOTE: Available at https://www.w3.org/TR/xml-c14n11/

In clause 2.2, the following informative reference is added.

[i.1]   ANSI/SCTE 214-1 "MPEG DASH for IP-Based Cable Services Part 1: MPD Constraints and Extensions"

4.3 Clause 5 – User experience

4.3.1 Update key event text in clause 5.2

In clause 5.2, two rows are added to Table 2: "Relevant remote control buttons or key events for the end user when using interactive applications" as shown underlined.

BACK button	Variable usage as defined by the application (typically going back one step in the application flow).
Number keys	Variable usage as defined by the application (typically used for numeric input or channel selection).
Transport keys (play, pause, stop, FF, FR)	Variable usage as defined by the application (typically used to control media playback).
2 program selection buttons (e.g. P+ and P-)	If available: selects the next or previous broadcast service in the internal channel list which may lead to the termination of the running application as described in clause Error: Reference source not found. These functions remain active at all times while broadcast-related applications are running – see clause 6.2.2.2.

The following text is added immediately after the table.

Some input devices may provide the user with a combined play/pause function instead of separate play and pause functions. Applications should be written to cater for both cases.

4.4 Clause 6 – Service and application model

4.4.1 Channel change key behaviour with broadcast independent applications

In clause 6.2.2.2, the following paragraph is extended with the sentence shown underlined.

The channel change mechanisms offered by the terminal (e.g. P+/P- keys, number keys)

shall remain functional at all times while broadcast related applications are running, regardless of whether media is being presented and whether that originates from broadcast or broadband. <u>The behaviour of these channel change mechanisms is implementation-dependent when a broadcast-independent application is running (and hence no broadcast channel is selected).</u>

4.4.2 Clarification to 6.2.2.9 re key events

The following paragraph is added at the end of clause 6.2.2.9.

In all cases, an application launched from a non-HbbTV application environment shall be "activated" for the purposes of receiving key events (see clause 10.2.2.1).

4.4.3 video/broadcast object issues when playing A/V from broadband

The following text is added at the end of clause 6.2.2.7:

Broadcast related applications that wish to access information from the video/broadcast object, e.g. channelchange succeeded events or stream events, while playing broadband content, should put the video/broadcast object into the stopped state. When an application survives a channel change, e.g. caused by P+/P-, the video/broadcast object transitions from the stopped state into the connecting state and into the presenting state if available resources permit (e.g. if additional video and audio decoders are available beyond those used for presenting the broadband content). The application is responsible to put it back into the stopped state.

4.4.4 Running a regular HbbTV app on a channel not in the terminal channel list

The following text is added to clause 6.2.2 immediately before the paragraph starting "Figure 13 shall not apply when selecting an MPEG program which is not a broadcast DVB service."

Applications may select services using a mechanism called "locally defined channels" (see clauses 7.13.1.3 and 7.13.11 of the OIPF DAE specification [1]). These may refer to regular broadcast DVB services (whether found by a channel scan or not), in which case figure 13 and the rest of the application lifecycle shall apply. These may also refer to MPEG programs that are not broadcast DVB services.

4.4.5 Web Storage following channel change

In clause 6.3.2, the second bullet point is modified as shown by underline / strikethrough markup.

For resources loaded via DSM-CC object carousel, the origin shall be ~~the DVB~~<u>a</u> URI in the form ~~(as defined in ETSI TS 102 851 [10] clause 6.3.1)~~:

~~- "dvb" ":" "//" original_network_id "." transport_stream_id "." service_id "." component_tag.~~

- "hbbtv-carousel" ":" "//" organisation_id ":" carousel_id

where organisation_id is the organisation id associated with the currently-running broadcast related application, and carousel_id is the ID of the carousel from which the resource was loaded, both encoded in decimal with no leading zeros.

NOTE 1:   Only broadcast related applications have access to broadcast carousels, and only an application that has a defined organisation_id can be broadcast related.

NOTE 2:   URLs using the hbbtv-carousel: scheme cannot be used to access files from the carousel. This scheme is used solely as the origin associated with resources accessed using a dvb: URI.

NOTE: In this case, the "host" is the DVB triplet plus the component_tag.

Hexadecimal digits in the DVB triplet and the component_tag shall be encoded using lower case characters.

4.4.6 RFC for web origin allows it to be null in "privacy-sensitive" contexts

The following text is added to clause 6.3.2 after the text "For resources loaded via HTTP and HTTPS, the origin shall be as defined in clause 5.3 of the HTML5 Recommendation [54]".

An HbbTV application shall not be considered a "privacy-sensitive" context for the purposes of clause 7.3 of RFC 6454 [25] and an Origin header shall be included in HTTP requests made on behalf of an HbbTV® application, and during the process of launching an HbbTV® application.

4.4.7 Clarifications relating to origins for application boundary

In clause 6.3.2, the text shown underlined is added to note 2.

NOTE 2:   URLs using the hbbtv-carousel: scheme cannot be used to access files from the carousel or for signalling in a simple_application_boundary_descriptor or an <applicationBoundary> element of an XML AIT. This scheme is used solely as the origin associated with resources accessed or referenced using a dvb: URI.

In clause 6.3.3, a new note 4 is added as shown underlined and the existing note 4 renumbered accordingly.

one or more dvb: URL prefixes. The application boundary shall be extended to include also object carousels referenced by such prefixes.

NOTE 4:   As defined above, the application boundary is held by the terminal as a set of origins. Clause 6.3.2 defines how to obtain the origin of a dvb: URL. The resulting origins will use the hbbtv-carousel: scheme as defined in clause 6.3.2.

4.4.8 Use of current.ait for starting a b-r app in the time between successful service selection and AIT acquisition

In clause 6.2.2.6.1, the paragraph shown is modified with the underlined text.

If these conditions are met, the application shall transition to be a broadcast-related application as defined in clause Error: Reference source not found and hence the broadcast signalling in the successfully selected service shall be obeyed thereafter. The application should be authored to follow the behaviour defined in clause Error: Reference source not found. These conditions shall apply regardless of whether an application was originally launched as broadcast-related or as broadcast-independent and regardless of how many times an application may have previously transitioned from broadcast-related to broadcast-

independent or vice-versa. <u>The application shall remain broadcast-independent until it has been determined whether it will transition to be broadcast-related or be killed. Hence there will be a short period while the terminal is acquiring the AIT when the broadcast service is selected but a broadcast-independent application is running.</u>

<u>NOTE: Applications need to wait for the the completion of the transition to broadcast-related before calling any APIs that are not available to broadcast-independent applications.</u>

4.4.9 Confusing language added to 6.2.2.2 re channel changing mechanisms

In clause 6.2.2.2, the following text is modified as shown with underline / strikethrough..

The channel change mechanisms offered by the terminal ~~(e.g. P+/P- keys, number keys)~~ shall remain functional ~~at all times while broadcast related applications are running,~~ regardless of whether media is being presented and whether that originates from broadcast or broadband <u>with the following clarifications</u>.

- <u>Mechanisms that depend on key events that are available to HbbTV applications (e.g. number keys) shall remain functional at all times while broadcast related applications are running and the running application has not successfully requested the key events concerned. While a running HbbTV application has claimed the keys concerned then a channel change mechanism depending on those keys shall be disabled.</u>

- <u>Mechanisms that do not depend on key events that are available to HbbTV applications (e.g. P+/P- keys, voice) shall remain functional at all times while broadcast related applications are running.</u>

- The behaviour of these channel change mechanisms is implementation dependent when a broadcast-independent application is running (and hence no broadcast channel is selected).

4.5 Clause 7 – Formats and protocols

4.5.1 Update reference to TS 102 809

In clause 7.2.2, the first paragraph is extended with the text shown underlined.

DSM-CC object carousel as defined in clause 7 of TS 102 809 [3] shall be supported. <u>The present document does not require the use of the protection mechanism described in clause 9 of TS 102 809 [3] by either broadcasters or terminals. Requirements for the use of this mechanism may be defined by the appropriate specifications for each market where the terminals are to be deployed.</u>

In clause 7.2.3.1, the first paragraph is extended with the text shown underlined.

Table 5 identifies the descriptors and other signalling entities whose MPEG-2 encoding shall be supported. Clause numbers and page numbers refer to TS 102 809 [3]. <u>The present document does not require the use of the protection mechanism described in clause 9 of TS 102 809 [3] by either broadcasters or terminals. Requirements for the use of this mechanism may be defined by the appropriate specifications for each market where the terminals are to be deployed.</u>

4.5.2 Clarification on XML AIT example

In table 7, "Contents of XML AIT for Broadcast-independent applications", in the row for the applicationTransport element, the cell in the column "Requirement on XML AIT file" is extended with the following;

The URLBase element shall be a URL ending with a slash ("/") character. No URLExtension elements shall be present. Only one applicationTransport element with type HTTPTransportType shall be present in the scope of the application.

4.5.3 Missing XML declaration in example XML AIT

In clause 7.2.3.2, "xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" is added to the example XML AIT as shown underlined.

```
<?xml version="1.0" encoding="UTF-8"?>
<mhp:ServiceDiscovery
    xmlns:mhp="urn:dvb:mhp:2009"
    xmlns:hbb="urn:hbbtv:application_descriptor:2014">
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    <mhp:ApplicationDiscovery DomainName="example.com">
        <mhp:ApplicationList>
            <mhp:Application>
                <mhp:appName Language="eng">Whizzo Play Along Quiz</mhp:appName>
                    <mhp:applicationIdentifier>
                    <mhp:orgId>123</mhp:orgId>
                    <mhp:appId>456</mhp:appId>
                </mhp:applicationIdentifier>
```

4.5.4 Ignoring unsupported AIT descriptors

Text is added to clause 7.2.3.1 as shown underlined.

Elementary streams that are used to carry an application information table may additionally carry information using other table_ids. When acquiring and monitoring for AIT elementary streams, terminals shall silently ignore table_ids not supported for carriage of AIT information.

NOTE: The present document only requires support for table_id 0x74 as defined in ETSI TS 102 809 [3].

<u>AIT subtables for HbbTV applications may include descriptors that are not required to be supported by the present document (see DVB services - i.21). Terminals should not support these AIT descriptors unless required by another specification. Terminals shall ignore AIT descriptors that they do not support.</u>

Table 5: Supported application signalling features

4.5.5 XML Parsing

The following is added to clause 7.2.3.2 between the first and second paragraphs (before "The semantics of the fields ..."):

The XML AIT shall not contain an XML Document Type Definition ("<!DOCTYPE ...>").

The following is added to clause 7.2.4 at the end of the clause;

The XML event description file defined in clause 8.2 of TS 102 809[3] shall not contain an XML Document Type Definition ("<!DOCTYPE ...>").

4.5.6 Lack of linkage between spec version in AIT signalling and version in DOCTYPE in individual pages of app

The following is added to clause 7.2.3.1 in Table 5 "Supported application signalling features", in the row for "5.2.5 Platform profiles", at the end of the cell in the "Notes" column:

For example, an application signalled as requiring [1,1,1] is able to detect that it's running on a [1.4.1] terminal and take advantage of the additional features defined in the present document.

4.5.7 Including or not including data services in the service list based on HbbTV version

In clause 7.2.6, changes are made as indicated using underline / strike-through markup.

Terminals shall process the data_broadcast_descriptor in the SDT. ~~and~~ Terminals shall include, in the terminal's service list all those~~,~~ data services that:

~~signal applications that are supported. If the selector_bytes are not present, the service shall not be included in the terminals service list.~~

• carry a data_broadcast_descriptor that indicates the HbbTV data_broadcast_id and have selector_byte present and

• signal an HbbTV application that is supported by the terminal

The present document is intentionally silent about data services that signal application(s) that are not supported. There are a number of reasons and/or circumstances why it may be appropriate to still include these in the terminal channel list.

4.5.8 Ambiguous use of "this"

In clause 7.3.2.6, "this" in the following note is replaced by "The preceding paragraph".

NOTE: This does not apply to HTTP requests made by the MPEG DASH player or the DRM agent.

4.5.9 Clarification on FSA: Is group priority value 0x00 valid?

In clause 7.2.7.2, the bulleted list introduced by "The semantics are as defined in ETSI ES 202 184 [36] with the following exceptions:" is extended with the following item:

• group_priority: priority 0 is the highest priority and increasing group_priority value means decreasing actual priority. In the present document there is no allocation of priority values. The system group has no specific meaning.

4.5.10 Carousel access following channel change

The following text is added at the end of clause 7.2.5.3.

Applications that have HTML documents loaded from a carousel and wish to access equivalent files in a different carousel after a channel change need to take care to ensure that valid file references are used. Without action by the application, relative URLs would still resolve to the original carousel. Applications can use absolute dvb: URLs to explicitly reference the new carousel. Alternatively, if it is desireable to use relative file references, the HTML5 `<base>` element may be used to update the base path used for the resolution of future relative URLs.

Where the same carousel (as defined in B.2.10 of TS 102 809) is signalled in a new service that is on the same multiplex, the terminal shall be able to successfully resolve relative URLs after a channel change since the carousel remains accessible in this case (see clause 9.2).

An example of how the `<base>` element may be used to change the document's base URL is shown below:

```
if (document.getElementById('myBase')) {
    document.getElementById('myBase').href = newBaseUrl;
} else {
    var newBase = document.createElement('base');
    newBase.setAttribute('id', 'myBase');
    newBase.setAttribute('href', newBaseUrl);
    document.getElementByTagName('head')[0].appendChild(newBase);

        }
```

4.5.11 Pending requests for FSA files when carousel is unmounted

A new clause 7.2.7.5 is added as follows:

7.2.7.5   File groups referenced by multiple carousels

A file group might be referenced by several carousels. If a carousel is unmounted and a new carousel mounted which references the same file group then the following apply:

•   The file group cache may continue to be filled

•   Pending requests for files in the group to be cached may continue or be restarted in an implementation specific manner

4.5.12 URL prefixes for DVB URLs in App Boundary Descriptor

The following text (that occurs in both clauses 7.2.3.1 and 7.2.3.2) is modified in both places as shown using underline / strike-through markup.

> When prefixes start with "http://" or "https://", ~~Only~~ only prefixes forming at least a second-level domain shall be supported.

4.5.13 Minimum performance requirements for DSM-CC OC

A new clause 7.2.5.5 is added as follows.

7.2.5.5   Performance (informative)

Terminals should take account of the following recommendations;
- Terminals should speculatively cache object carousel modules when a terminal mounts the carousel without waiting for an object in a module to be needed by an HbbTV application

NOTE 1:   Clause 10.2.1 of the present document requires the cache size to be at least 3 Mbytes.

- Terminals should cache object carousel modules in the compressed form.

NOTE 2:   Experience suggests compression ratios are approximately 1:3 and the minimum cache size has been based on this ratio and the assumption modules are cached in the compressed form. An HbbTV terminal with a minimum size cache that stored modules or objects uncompressed will likely give an unacceptably poor user experience on carousel delivered applications except for the simplest.

- Terminals should be able to load an entire carousel in one cycle assuming the carousel size is less than the size of the DSM-CC OC cache.

NOTE 3:   For example, if a carousel less than 3MBytes has a cycle time of 6s then all the data from that carousel should have been loaded and immediately accessible to applications after 6s from when the terminal mounts the carousel.

NOTE 4:   Loading an entire carousel in one cycle cannot be guaranteed under all circumstances as terminals do more than thing at one time and it is possible that an unrelated operation might interfere with loading a carousel and result in part of a carousel being missed. Alternatively an unrecoverable reception error may result in part of a carousel being missed.

- If a terminal misses part of a carousel in the first cycle then it should retain the data acquired in that cycle and fill in the gap in the next cycle.

NOTE 5:   Given a carousel that is less than 3MBytes which has a cycle time of 6s, if there is an error loading some of the data then the full data of the carousel should be loaded within 12s of the terminal from when the terminal mounts the carousel.

- Within a carousel, some of the data may be repeated more frequently than others and may be put in the same module (see TS 102 809 B.2.6). Terminals should make this data available to applications as soon as it (and any dependencies) have been loaded without waiting for a complete cycle time.

NOTE 6:   For example, if the initial HTML page of an HbbTV application, the red button image and any dependencies (e.g. directory, service gateway objects) are in the same module and are repeated every 2s in a carousel whose cycle time is 30s then terminals should load that module and make the initial HTML page and red button image available within 2s and without waiting for the complete 30s cycle of the carousel.

- Terminals should allow for carousels with relatively long cycle time such as 30s or longer without timing out.

- When the module containing the service gateway object has been loaded, all files that have been loaded and whose directory objects have been loaded should be immediately accessible without waiting for any more modules to be loaded.

- Terminals should not;

  - load only one module in each carousel cycle

- wait until the data from a module is needed before loading a module or processing the data of a module
- descend the directory tree in a carousel only loading the module containing the definition of each new directory after it has loaded the module containing the parent directory
- fall back to not caching at all when receiving a carousel that is too large to fit in their DSM-CC OC cache
- wait until the module containing the service gateway object has been successfully loaded and parsed before loading other modules

4.5.14 Table reference in clause 7.3.1.1

In clause 7.3.1,1, in the following paragraph, the reference to table 9 is changed to table 9a.

- For each of the technologies listed in Table 9, terminals supporting the broadcast IRD from ETSI TS 101 154 [14] shall also support the related DASH requirement as shown.

4.5.15 Processing time for stream events varies a lot (even on the same device)

The text underlined is added to clause 7.2.4 as shown.

7.2.4 Synchronization

The terminal shall support "do-it-now" events as defined in clause 8 of ETSI TS 102 809 [3]. Support of events synchronized to a DVB timeline as referred to in that document is not included.

Broadcasters shall place all "do-it-now" stream descriptors to be monitored simultaneously by an application on a single PID. This may be the same PID as is used for other DSM-CC sections.

NOTE: DSM-CC stream events are usually reported to an HbbTV application very soon after being received by the terminal. Since it may take between 250ms and 750ms (or longer) for video and audio to pass through the media pipeline in a terminal, typically stream events will be passed to the application before the video and audio received at the same time is visible and audible.

The XML event description file defined in clause 8.2 of TS 102 809 [3] mustshall not contain an XML Document Type Definition ("<!DOCTYPE ...>").

7.2.5 DSM-CC carousel

4.5.16 DSM-CC cache validity

In clause 7.2.2, the words "is mounted and" are removed as shown with strikethrough markup.

The terminal shall consider cached information to remain valid only whilst the relevant object carousel is mounted and is being monitored. This prevents the possibility of retrieving stale data from a carousel which has been unmounted and remounted if the version number of an object has been incremented such that it has the same value as when it was cached. For the avoidance of doubt, changes to DSI messages shall not be considered to be an

unmounting of the carousel.

4.5.17 Reconciling user understanding of what terminal model they have with entries in a broadcaster log file

In clause 7.3.2.4, text is added as shown underlined.

- The <vendorName> field shall reflect the consumer-facing make / brand of the terminal. In the exceptional case that a terminal does not have any consumer-facing make / brand visible anywhere in its user interface, vendorName may instead identify the manufacturer / supplier of the platform hardware with the prefix "Unknown-".

- The <vendorName>, <softwareVersion>, <familyName> combination, along with <hardwareVersion> where included, shall differ between terminals with significantly different implementations or performance. Two products that have the same software but where HbbTV applications behave differently due to hardware differences (e.g. processor speed, RAM speed, RAM size) shall differ by either <familyName> or <hardwareVersion> or both.

- The <modelName> field should be representative of the consumer-facing model name to allow log messages to be matched up with user reported problems. In the exceptional case that a terminal does not have text representative of the consumer-facing model name visible anywhere in its user interface, modelName may instead be set to the string "Unknown".

- The <familyName> field shall have the semantics defined in clause 7.3.3.2 of the OIPF DAE specification [1] but shall additionally be chosen so as to be globally unique. This shall be achieved either by prefixing with a reverse domain name of the organisation allocating the remaining portion of the familyName, or by using a version 4 UUID as the familyName, formatted as a simple string (i.e. without any urn:uuid prefix) [64]. Devices in a family differ only by details that do not impact the behaviour of the HbbTV aspect of the device, e.g. screen size, remote control, number of HDMI ports, size of hard disc. For white label terminal implementations where no development is done by the consumer facing brand, the organization that integrates the implementation of the present document into the terminal is responsible for populating this field.

EXAMPLE: If company A integrates an implementation into a TV set, manufactures a number of these TV sets, which are then sold to companies B, C and D who each sell them in turn to consumers under their own brand without changing the software, company A is responsible for populating this field.

4.5.18 TTML, mismatch between HbbTV and ETSI EN 303 560

In clause 7.3.1.5.1, the handling of "default" is modified as shown using underline / strikethrough markup.

When resolving tts:fontFamily references from EBU-TT-D subtitles, terminals shall search for a match in fonts that have been successfully downloaded before considering the embedded fonts listed in clause 10.2.1. When matching embedded fonts, the following mappings for TTML genericFamilyName shall apply:

- ~~"default",~~ "sansSerif " and "proportionalSansSerif" shall match the Tiresias™ embedded font;

- "monospace" and "monospaceSansSerif" shall match the Letter Gothic embedded font.

- <u>"default" shall match either of the embedded fonts listed above</u>

Other genericFamilyNames may match an appropriate embedded font if one is available; otherwise they shall be treated as "default".

4.5.19 Backporting XML AIT schema changes from OpApp into core spec

In clause 7.2.3,2, in table 7, in the row for applicationDescriptor/version, the contents both of the cells "Requirement on XML AIT file" and "Requirement on terminal" are replaced with the following;

> The syntax of <xsd:element name="version" type="ipi:Version"/> is replaced with <xsd:element name="version" type="mhp:unsignedInt31Bit"/>. No semantics are defined in the present document.

4.6 Clause 8 – Browser application environment

4.6.1 Avoid MediaSynchroniser API leaking "secret" stream URLs

The following property is added to those defined for the MediaSynchroniser embedded object in clause 8.2.3.2.1.

`String contentIdOverride`	
Description	This value overrides the content ID that would normally be reported to Companion Screen Applications and slave terminals during inter-device synchronisation. When the terminal is a master terminal and inter-device synchronisation functionality is enabled and the value of this property is a string then the content ID that the terminal uses for the CSS-CII service endpoint and the CSS-TS service endpoint is overridden and the value of this property is used instead. If the value of this property is `null` or `undefined` then there is no override. This behaviour is defined in clauses 13.6.2 and 13.8.2. The value of this property shall initially be `null`.

4.6.2 Clarifying media synchroniser error codes

In clause 8.2.3.2.2, the description of the initMediaSynchroniser method is modified with the addition of clarifications as shown underlined.

> If the MediaSynchroniser has already been initialized (including if it is in a permanent error state) then this call shall fail and an error event shall be triggered <u>with error code 13 or 17 (according to the definition of the error codes)</u>.

> If the media stream for the media object is determined to be not available or if the selected timeline is determined to be not available then this shall result in a permanent error of the MediaSynchroniser and an error event shall be triggered <u>with error code 15 or 16 (according</u>

to the definition of the error codes).

If this method completes without error then the MediaSynchroniser shall be considered initialized.

When this MediaSynchroniser is initialized, if there is an existing MediaSynchroniser that has already been initialized then this shall result in a permanent error of the existing MediaSynchroniser and it shall trigger an error event with error code 18.

In clause 8.2.3.2.2, the description of the initSlaveMediaSynchroniser method is modified with the additions of clarifications as shown underlined.

If the service endpoint at the specified URL is not available then this shall result in a permanent error of the MediaSynchroniser and an error event shall be triggered (see clause 13.3.8) with error code 6.

If the MediaSynchroniser has already been initialized (including if it is in a permanent error state) then this call shall fail and an error event shall be triggered with error code 13 or 17 (according to the definition of the error codes).

If the terminal does not support the capability to act as a slave terminal, then this method shall be undefined.

If this method completes without error then the MediaSynchroniser shall be considered initialized.

When this MediaSynchroniser is initialized, if there is an existing MediaSynchroniser that has already been initialized then this shall result in a permanent error of the existing MediaSynchroniser and it shall trigger an error event with error code 18.

In clause 8.2.3.2.2, the description of the addMediaObject method is modified with the additions of clarifications as shown underlined.

If the MediaSynchroniser is not initialized, or is in a permanent error state, then this call shall be ignored and an error event dispatched with error code 7 or 13 (according to the definition of the error codes).

If the media object has already been added to the MediaSynchroniser (either by passing it to addMediaObject() or initMediaSynchroniser() methods), then this call shall be ignored and an error event dispatched with error code 4.

If adding the media object would result in multi-stream synchronisation using a combination of streams that is unsupported by the terminal, then this call shall be ignored and a transient error of the MediaSynchroniser shall be generated with error code 20.

The actual presentation of the content might be delayed while the terminal aligns the master media object and the other media object(s) to achieve synchronized presentation in accordance with the correlation timestamps.

The terminal may be required to buffer one or more of the media objects. If the terminal has insufficient buffer space or cannot present the media sufficiently early then the media object shall be added to the MediaSynchroniser but a transient error of the MediaSynchroniser shall be generated with error code 1 or 11.

The terminal shall select the components from the media object to be presented in accordance with the value of the multiDecoderMode parameter and the definitions in clause

10.2.7.

If the terminal fails to access a media item or its timeline, e.g. the resource is not available, then adding the media object shall fail and the MediaSynchroniser shall dispatch an error event with error code 2 or 3 (according to the definition of the error codes).

If the correlation timestamp correlationTimestamp is undefined a correlation timestamp where the value of both properties is 0 shall be assumed. If the correlation timestamp is null or has an invalid format, adding the media object shall fail and the terminal dispatch an error event with error code 5.

In clause 8.2.3.2.2, the description of the removeMediaObject method is modified with the additions of clarifications as shown underlined.

If the media object has not already been added to the MediaSynchroniser or is the master media object then this call shall be ignored and an error event dispatched with error code 8.

If the MediaSynchroniser is not initialized, or is in a permanent error state, then this call shall be ignored and an error event dispatched with error code 7 or 13 (according to the definition of the error codes).

In clause 8.2.3.2.2, the description of the updateCorrelationTimestamp method is modified with the additions of clarifications as shown underlined.

If the media object either is not already added to the MediaSynchroniser or is the master media object, then this call shall be ignored and an error event dispatched with error code 8.

If the MediaSynchroniser is not initialized, or is in a permanent error state, then this call shall be ignored and an error event dispatched with error code 7 or 13 (according to the definition of the error codes).

In clause 8.2.3.2.2, the description of the enableInterDeviceSync method is modified with the additions of the clarification as shown underlined.

If the MediaSynchroniser is not initialized, or is in a permanent error state, then this call shall be ignored and an error event dispatched with error code 7 or 13 (according to the definition of the error codes).

In clause 8.2.3.2.2, the description of the disableInterDeviceSync method is modified with the additions of the clarification as shown underlined.

If the MediaSynchroniser is not initialized, or is in a permanent error state, then this call shall be ignored and an error event dispatched (see clause 13.3.8) with error code 7 or 13 (according to the definition of the error codes).

In clause 8.2.3.2.4, some of the error values are clarified as shown using underline / strike-through markup.

Value	Description	Permanent or Transient
1	Synchronization is unachievable because the terminal could not delay presentation of content (represented by a media object added using the `addMediaObject()` method) sufficiently to synchronize it with the master media. For example: the buffer size for media synchronization is not sufficient.	Transient
2	The presentation of media object(that was added using the `addMediaObject()` method) failed. The specific reason is given by the error handler of that media object.	Transient
3	The media or the selected timeline for the media could not be found or the media timeline is no longer present (for media represented by a media object that was added using the `addMediaObject()` method).	Transient
4	Media object is already associated with the `MediaSynchroniser`.	Transient
5	The correlation timestamp set for a media object is `null` or has an invalid format.	Transient
6	While acting as a slave terminal, i~~I~~nter-device synchronization with a master terminal failed because of unavailability, e.g. an endpoint is not available or disappeared. Applications should rediscover available terminals as defined in clause 14.7.2 before continuing with inter-device synchronization.	Permanent
7	The call failed because the `MediaSynchroniser` is not yet initialized.	Transient
8	The media object ~~referenced as an argument in the call~~has not been ~~needed to have already been~~ added to the `MediaSynchroniser` using the `addMediaObject()` method ~~but it has not been~~.	Transient
9	The media object (that was passed using the `addMediaObject()` method) is not in a suitable state to participate in synchronization. See clause 9.7.1.	Transient
10	While acting as a slave terminal, i~~I~~nter-device synchronization with a master terminal failed because of a fault in protocol interaction, e.g. the master terminal did not provide required messages or data. Applications can consider trying again.	Permanent
11	Synchronization is unachievable because the terminal could not present the content (represented by a media object added using the `addMediaObject()` method) sufficiently early to synchronize it with the master media.	Transient
13	The method call failed because the `MediaSynchroniser` is in a permanent error state or because it has been replaced by a newer initialized `MediaSynchroniser`.	Transient (see note ~~1~~)
14	The presentation of the master media (that was specified as an argument when the `initMediaSynchroniser()` method was called) failed. The specific reason is given by the error handler of that media object.	Permanent
15	Either t~~T~~he master media object or the selected timeline for a~~the master~~ media object (that were specified as arguments when the `initMediaSynchroniser()` method was called) could not be found or the media timeline is no longer present.	Permanent
16	The master media object is not in a suitable state to participate in synchronization. See clause 9.7.1.	Permanent
17	The method call failed because the `MediaSynchroniser` is already initialized.	Transient
18	Thi~~s~~e `MediaSynchroniser` has been replaced by a new `MediaSynchroniser` being initialized.	Permanent
19	The master terminal has reported that the `presentationStatus` of the master media has changed to "transitioning" (see clause 13.6.3).	Transient
20	The combination of streams requested for multi-stream synchronisation (by a call to the `addMediaObject()` method) is unsupported.	Transient
NOTE:	The `MediaSynchroniser` will already be in a permanent error state. If this error occurs, the `MediaSynchroniser` remains in the permanent error state.	

4.6.3 Issue with companion launcher API

The description of the `discoverCSLaunchers()` method is modified as shown by underline / strike-through markup.

When true is returned, the terminal shall determine a set of CS Launcher Applications that are Connected (as defined in clause 14.3.2) and report these by scheduling the onCSDiscovery() callback ~~shall be scheduled to fire~~ within 1 second. There shall be no callback scheduled if false is returned.

In determining the set of Connected CS Launcher Applications, the terminal performs any discovery and/or association steps that are needed. The details of what is done during this function call or after this function call dependes depends on the protocol between the HbbTV® terminal and the CS launcher application and is implementation specific.

In the description of the `onCSDiscovery()` method, the description of the `csLaunchers` property is modified as shown using underline / strike-through markup:

A JavaScript Array object containing zero or more `DiscoveredCSLauncher` objects (see clause 8.2.6.3) where each object in the array represents a CS Launcher application that is Connected (as defined in clause 14.3.2). ~~was either:~~

- ~~currently connected at the time of the call to `discoverCSLaunchers()` that triggered this callback;~~
- ~~or subsequently connected after the time of the call to discoverCSLaunchers() that triggered this callback.~~

The protocol for determining the CS Launchers to be included in this array is out of scope, and not defined by the present document.

4.6.4 XML parsing

In clause 8.2.1.1, the description of the targetURL argument of the addStreamEventListener method shall be amended with additional text as shown underlined below.

The URL of the DSM-CC StreamEvent object or an HTTP or HTTPS URL referring to an XML event description file (as defined in clause 8.2 of [3] and profiled in section 7.2.4) describing the event.

4.6.5 Update inter-device sync spec from DVB blue book back to ETSI

In clause 8.2.3, in the description of the updateCorrelationTimestamp method, "DVB Bluebook A167-2" is replaced by "ETSI TS 103 286-2".

4.6.6 Clarification on value for property "lastError" if no error occurred

The following changes are made in clause 8.2.3.2.1;

The description of the `lastError` property is changed from

Shall be the code of the last error that occurred for this MediaSynchroniser object as defined in clause 8.2.3.2.4.

to

If no error has yet occurred for this MediaSynchroniser object then the value of this property shall be null, otherwise the value shall be the code of the last error that occurred for this MediaSynchroniser object as defined in clause 8.2.3.2.4.

In the description of the `lastErrorSource` property, the second sentence is changed from

> If the error was not caused by a media object or the master terminal or interaction with the master terminal, then this shall be null.

to

> If no error has yet occurred for this MediaSynchroniser object, or if the error was not caused by a media object and was not caused by the master terminal or interaction with the master terminal, then this shall be null.

4.6.7   errorCode(s) for onCSLaunch

In clause 8.2.6.1, the table of error code descriptions is modified as follows:

1) The text introducing the table is changed from:

> The following error codes may be carried in the onCSLaunch callback:

to

> The error code in the onCSLaunch callback shall be one of the following as defined by the error description:

2) The following text is added to the row for "op_not_guaranteed" at the end of the "Error Description" column:

> Since the Launcher application on the CS may not have knowledge that the CS app has actually launched, this error code is termed 'op_not_guaranteed'. If the launcher application is aware that the launch or install operation has completed successfully then this code shall be used.

4.6.8   Media synchronization informative examples don't construct CorrelationTimestamp objects

In clause 8.2.3.3.1, the text shown underlined is added;

> Applications shall construct objects that conform to this class definition to deliver correlation timestamps to the terminal. Any object conforms to this class definition if it includes the named properties described in this class definition.

4.6.9   Warning to app developers that some values/properties are insecure

In clause 8.2.1.2, the following text is added to the definition of the text property of the StreamEvent interface.

> Application developers should be aware that in some circumstances an attacker may be able to modify the broadcast signalling from which this data is derived. Applications shall not use this data in a way that would result in it being executed by the browser. Applications should be written to be tolerant of incorrectly formatted data or values for this data which are outside the expected range without hanging up or crashing.

In clause 8.2 the following text is added right at the end.

> Application developers should be aware that in some circumstances an attacker may be able

to modify the broadcast signal carrying a carousel file. Applications should be written to be tolerant of incorrectly formatted data or values which are outside the expected range without hanging up, locking up or crashing.

Application developers should be aware that if a broadband-delivered application uses data from a carousel in a way that would result in it being executed by the browser, then the level of security of that application is then only as secure as the carousel delivery mechanism. Broadband-delivered applications should avoid such use unless the protection mechanism described in clause 9 of TS 102 809 is in use.

4.6.10 initMediaSynchroniser

In clause 8.2.3.2.2, in the description of the initMediaSynchroniser method, the words "and is not in permanent error state already" are inserted as shown underlined.

When this MediaSynchroniser is initialized, if there is an existing MediaSynchroniser that has already been initialized and is not in permanent error state already then this shall result in a permanent error of the existing MediaSynchroniser and it shall trigger an error event with error code 18.

In clause 8.2.3.2.2, in the description of the initMediaSynchroniser method, the words "Initialisation is immediate, however if at that time, or subsequently," are inserted as shown underlined.

Initialisation is immediate, however if at that time, or subsequently, if the media stream for the media object is determined to be not available or if the selected timeline is determined to be not available then this shall result in a permanent error of the MediaSynchroniser and an error event shall be triggered with error code 15 or 16 (according to the definition of the error codes).

4.6.11 Media Synchronisation

4.6.11.1 Clause 8.2.3.1

The text shown underlined is added in clause 8.2.3.1.

```
// this method call is asynchronous. If onSynchroniserInitialised is supported then wait for that
// Otherwise allow some time for it to finish.
ms.initMediaSynchroniser(vba1, timeline_spec_vba1);

// ... some xmlHTTPRequest to get correlation timestamps (e.g. from MRS) for the DASH stream related
//   to the timeline for the MediaSynchroniser API

timestamp_vba1_dasha1 = {'tlvMaster' : 12345, 'tlvOther' : 12445};

// this method call is asynchronous. If onMediaObjectAdded is supported then wait for that.
// Otherwise do not assume synchronisation happens immediately.
ms.addMediaObject(dasha1, timeline_spec_dasha1, timestamp_vba1_dasha1);
```

4.6.11.2 Clause 8.2.3.2.1

The following are added to clause 8.2.3.2.1.

function onSynchroniserInitialised ()	
Description	The function that gets called when the initialization of this MediaSynchroniser object is successfully completed – specifically after the timeline is being monitored and the currentTime property has started to be updated.

function onMediaObjectAdded (Object mediaObject)	

Description	The function that gets called when a call to the addMediaObject method is successfully completed. Specifically, component selection has completed and either;
	• The set of selected components wil not change or
	• The changed set of components have started to be presented synchronised (timeupdate events are being generated due to "the usual monotonic increase of the current playback position during normal playback" as defined in HTML5 []).
	The terminal shall pass one argument in the call, the media object that was passed as an argument to `addMediaObject` and that has been successfully added.

4.6.11.3Clause 8.2.3.2.2

The following changes are made in clause 8.2.3.2.

The text shown underlined is added to the description of the method initMediaSynchroniser;

> If this method completes successfully without an error event being triggered then the `MediaSynchroniser` shall be considered initialized and a `SynchroniserInitialised` event shall be triggered after the `currentTime` property starts being updated.

> In some early implementations where `SynchroniserInitialised` event is not supported,there is no explicit event indicating a successful call to this method. Applications may detect success by polling the `currentTime` property and detecting when it changes from NaN to a numeric value.

The following text is removed from the descirption of the method addMediaObject.

> The actual presentation of the content might be delayed while the terminal aligns the master media object and the other media object(s) to achieve synchronized presentation in accordance with the correlation timestamps.

The following text is added to the descirption of the method addMediaObject (Object mediaObject, String timelineSelector, CorrelationTimestamp correlationTimestamp, Number tolerance, Boolean multiDecoderMode);

> The method is asynchronous. The process of determining any changes in the set of media components to be presented might not be completed before the method call returns. The actual changes in the content being presented will probably be delayed while the terminal de-initialises and re-initialises media decoders and aligns the master media object and the other media object(s) to achieve synchronized presentation in accordance with the correlation timestamps.

> If the method completes successfully without an error event bejng dispatched then a `MediaObjectAdded` event shall be dispatched.

> In some early implementations where `MediaObjectAdded` is not supported,there is no explicit event indicating a successful call to this method. In these implementations, if the set of selected components changes to include ones in the added media object, applications may detect success implicitly by registering for events on the media object being added. For an HTML5 media element that is paused;
> - Firing of a playing event without any subsequent error will indicate that a component will be presented soon (e.g. within 250 ms – 2s)
> - Firing of timeupdate events due to usual monotonic increase of the current playback position during normal playback (see the "time marches on steps" in HTML5 []) will indicate that a component is being presented. Applications need to be careful to distinguish these timeupdate events from timeupdate events fired for other reasons.
> In these early implementations, there is no implicit event enabling applications to detect success if the set of selected components does not change to include ones in the added media object.

The following text is aded to the description of the method removeMediaObject. Additionally "or is" is changed to "and is not".

Using this method to remove all media objects added to a `MediaSynchroniser` using `addMediaObject` shall terminate multi-stream synchronization but shall leave the timeline specified when the `MediaSynchroniser` was initialized being monitored. Applications shall be able to continue to read the value of the timeline from the `currentTime` property.

Using this method to remove the master media object (specified in the call to `initMediaSynchroniser`) shall stop timeline monitoring and release all scarce resources associated with this `MediaSynchroniser`. The `MediaSynchroniser` object shall enter the permanent error state and en error event shall be triggered with error code 18.

4.6.11.4Clause 8.2.3.2.3

The following are added to clause 8.2.3.2.3

For the intrinsic event `onSynchroniserInitialised`, a corresponding DOM level 2 event shall be generated, in the following manner:

Intrinsic event	Corresponding DOM 2 event	DOM 2 Event properties
onSynchroniserInitialised	SynchroniserInitialised	Bubbles: No Cancelable: No Context Info: None

For the intrinsic event `on onMediaObjectAdded`, a corresponding DOM level 2 event shall be generated, in the following manner:

Intrinsic event	Corresponding DOM 2 event	DOM 2 Event properties
onMediaObjectAdded	MediaObjectAdded	Bubbles: No Cancelable: No Context Info: `mediaObject`

4.7 Clause 9 – System integration

4.7.1   Clarify URL scheme and MPD anchor rules

The following paragraph is added at the start of clause 9.2.

This clause describes how URL schemas can be used within HbbTV applications (HTML, JavaScript, images and references to A/V content).

4.7.2   Clarify resource usage by HTML5 media elements

In clause 9.6.2, a sentence is added in the following paragraph as shown underlined.

The terminal may use hardware audio and video decoders to decode and render <video> and <audio> HTML5 media elements. These hardware resources shall not be allocated to an HTML5 media element before it changes from being paused to 'potentially playing' (as defined in the HTML5 specification). When subsequently paused, an HTML5 media element shall retain its hardware resources, but shall be able to release these resources if required to start playing another HTML5 media element. Hardware resources shall also be released if the HTML5 media element is removed from the DOM and no other references to it exist (see Annex J for a code example of how to achieve this). When resources are released, the terminal may discard any decoded frames that have not been displayed.

4.7.3 Additional cases when 250ms ad insertion transition may not be possible to meet

In clause 9.6.3, the following two items are added to the list "The delay between the end of presentation of an HTML5 media element and starting presentation of another HTML5 media element shall be less than 250 ms if all of the following conditions are met:"

- the video in the two video elements either has the same frame rate, or one frame rate is an integer multiple of the other (see frame rate families in clause 10.4 of DVB-DASH)

- the video in the two video elements has the same colour primaries and transfer characteristics, e.g. BT.709 or BT.2020

4.7.4 Timing requirements for end of mid roll adverts

A new clause 9.6.12 is added as follows;

9.6.12 End of stream indication

The end of presentation of an HTML5 media element is notified to the application by means of an 'ended' event. This event shall not arrive before the last frame of video or the last audio sample is guaranteed to be presented (e.g. because it has entered a display processing pipeline). It should arrive within 80 ms of the last frame of video or the last audio sample being presented (whichever is the later) and shall be received within 250 ms of that time.

NOTE: When considered with the requirements in clause 9.6.3, this means that the transition at the end of an advert, either to another advert or back to the content, should be possible within 330ms but may be up to 500 ms.

4.7.5 Require the same DASH player for HTML5 video element and A/V control object?

A new clause 9.1.1.3 is added as follows:

9.1.1.3 Media player implementations and API behaviour

For DASH, terminals shall use the same DASH player implementation for any given MPD regardless of whether the A/V control object or the HTML5 video element are used.

4.7.6 Content supporting both Clear Key and conventional DRM

Clause 9.6.7 is amended as shown using underline / strikethrough markup.

If an application attempts to present DRM protected MPEG DASH content using the HTML5 <video> element and this is ~~denied~~ not decrypted by any of the DRM systems that are both listed in the DASH MPD using a ContentProtection element and active according to the state of the oipfDrmAgent object (see the `setActiveDRM` method defined in clause A.2.27), then this failure shall be reported to the application by a `MediaError` whose `code` property is set to `MEDIA_ERR_DECODE`. The application is then responsible for checking if the reason for this error was related to DRM and if so, obtaining more details about the error from the DRM system. For DRM systems that an application can access through the `oipfDrmAgent` object, these two steps would be done using the

`sendDrmMessage` method.

Errors relating to the presentation of protected content when the EME API is being used (see clause B.3) shall be reported as specified in the EME specification [66].

4.7.7 Use of current.ait for starting a b-r app in the time between successful service selection and AIT acquisition

In clause 9.2, the text shown underlined is inserted as shown.

It shall be possible to use dvb: URLs referring to applications signalled in the current service as defined in Table 4 of ETSI TS 102 851 [Error: Reference source not found] and optionally appended fragment component with the Application.createApplication() method. Use of dvb: URLs referring to applications from another service will cause createApplication() to fail as if the initial page could not be loaded. Attempts to use such a dvb: URL from the start of a channel change until a ChannelChangeSucceeded or ChannelChangeError event is generated shall fail. Attempts to use such a dvb: URL between when a ChannelChangeSucceeded event is generated and when the AIT of the new channel is acquired shall not be completed until the AIT of the new channel is received (if no AIT is received, the application will be terminated by the usual lifecycle rules). Any query component and fragment component assigned to this dvb:URL shall be attached to the application location URL signalled inside the corresponding AIT as follows:

4.7.8 HTML5 media element audio resume position

In clause 9.6.3, the last paragraph is modified as shown using underline / strikethrough markup.

When resuming the playback of an HTML5 media element that has previously been paused and references video content, the terminal shall start playback at or before the IDR following the pause position. When resuming the playback of an HTML5 media element that has previously been paused and references only audio content, the terminal shall start playback at or before the audio RAP following the pause position. When resuming ~~the~~ playback, the terminal should start playback as close as possible to the pause position, preferably from the next frame following the pause position. When resuming the playback of an HTML5 media element that has previously been paused while presenting and references content containing both video and audio tracks, the terminal should start playback with video and audio as closely synchronised as possible.

4.7.9 DASH - MPD events

Text is added and removed in clause 9.3.2.2 as shown by underline and strike-through markup below.

DASH events shall be reported to applications as `DataCues` according to the following mapping:

DataCue(TextTrackCue) property	MPD Events	Inband Events
Id	@id. If the @id attribute is not specified, id shall be set to an empty string. With reference to clause 9.1.5 of DVB-DASH, events without a value of @id specified shall not be considered as repetitions of the same event.	Id
startTime	@presentationTime (scaled according to the EventStream @timescale attribute) + the time offset of the start of the period from the start of the presentation. (MPEG DASH defines that the value of @presentationTime defaults to zero if the attribute is not present).	presentation_time_delta (scaled according to the timescale value) + the time offset of the start of the segment from the start of the presentation.
endTime	The startTime + @duration, subject to the minimum duration requirements below. If the @duration attribute is not specified, endTime shall be set to `Number.MAX_VALUE`.	The startTime + the event_duration, subject to the minimum duration requirements below. If event_duration is 0xFFFF, endTime shall be set to `Number.MAX_VALUE`.
pauseOnExit	False	False
Onenter	As defined in the HTML5 Recommendation [Error: Reference source not found].	As defined in the HTML5 Recommendation [Error: Reference source not found].
Onexit	As defined in the HTML5 Recommendation [Error: Reference source not found].	As defined in the HTML5 Recommendation [Error: Reference source not found].
data	The (character data) string value of the `Event` element. If the optional `contentEncoding` attribute is set to "base64" then the (character data) value of the `Event` element shall be decoded as described in IETF RFC 4648 [1] before use. If the `Event` element does not solely contain character data (e.g. if it contains child elements or if it is empty) then this property shall be an XML document subset such that the canonical form of the XML returned by reading this property shall be indentical to the canonical form of the `Event` element in the MPD starting with the opening `<Event> tag` and the closing `</Event>` tag. See clause 2.4, "Document subsets" of Canonical XML Version 1.1 [2].	`message_data`

Terminals shall ignore unsupported XML elements and attributes, regardless of namespace, except as required above. Events with unsupported XML elements and attributes shall be processed as required by the present document based on the values of the supported elements and attributes.

NOTE 1: A consequence of the above requirements is that SCTE 35 carried in MPD events as defined in clause 6.7.4 of ANSI/SCTE 214-1 [i.1] will be passed to applications in their entirety (including the `Signal.Binary` element) as that specification does not use a (character data) value for the `Event` element.

4.7.10 Potential conflict between HbbTV 9.6.2 and HTML5 re HTML5 load() method taking video & audio decoders

The following text in clause 9.6.2 is changed as shown using underline / strike-through markup.

The terminal shall support the existence within the same DOM of at least one HTML5 media element that is playing together with at least two HTML5 media elements in a paused state, where each HTML5 media element may be in a readyState of HAVE_CURRENT_DATA or higher. The terminal shall support each of the following scenarios:

• all three HTML5 media elements refer to DASH content;

• all three HTML5 media elements refer to ISOBMFF content;

• two of the three HTML5 media elements refer to DASH content and one refers to ISOBMFF content;

• two of the three HTML5 media elements refer to ISOBMFF content and one refers to DASH content.

The terminal may use hardware audio and video decoders to decode and render <video> and <audio> HTML5 media elements. There is no requirement for hardware resources to ~~These hardware resources shall not~~ be allocated to an HTML5 media element before it changes from being paused to 'potentially playing' (as defined in the HTML5 specification). An HTML media element that is not 'potentially playing' may make use of hardware resources that are not in active use elsewhere, but shall be able to release those resources to an HTML5 media element, A/V Control object or video/broadcast object where needed by any such object. When subsequently paused, an HTML5 media element shall retain its hardware resources, but shall be able to release these resources if required to start playing another HTML5 media element.

When resources are released, the terminal may discard any decoded frames that have not been displayed. Releasing hardware resources shall be as recommended in clause 4.7.14.17 of HTML5 [54], when either:

• The source is removed from an HTML5 media element and the media element is re-loaded. S~~Hardware resources shall also be released f~~ the HTML5 media element is removed from the DOM and no other references to it exist (~~see annex J for a code example of how to achieve this~~); or

• An HTML5 media element is garbage collected – the timing of this is non-deterministic and applications should not rely on this.

~~When resources are released, the terminal may discard any decoded frames that have not been displayed.~~

In the algorithm for seeking in HTML5 [54], the requirement to "Wait until the user agent has established whether or not the media data for the new playback position is available, and, if it is, until it has decoded enough data to play back that position." is modified in the present document. There is no requirement to wait until any data has been decoded. Applications cannot assume that the seeked event being fired means that playback is ready to start almost immediately.

The visual appearance of a <video> element that has no decoder resource currently allocated

is undefined. It is recommended that the terminal render such an element using the same behaviour as if the "visibility" CSS property was equal to "hidden". <u>Media elements may enter HAVE_CURRENT_DATA or higher readyState without the decoder being allocated. Applications cannot assume, based just on readyState, that something has already been decoded or presented.</u> , e.g. that data up to current position has been already decoded or that the current frame will be visible.

<u>NOTE 1:</u>   This is a relaxation of the normal HTML5 behaviour. In HTML5, a paused video element with a readyState of HAVE_CURRENT_DATA or greater is represented by a still video frame, but this is not possible on a device using hardware video decoders if no decoder resource is available to decode that frame.

If a terminal supports only one HTML5 media element that is 'potentially playing', and multiple media elements exist within the DOM, the request to transition to 'potentially playing' of one HTML5 media element (e.g. calling the play() method) shall cause all other media elements to pause and release their allocated hardware resources. The transition to 'potentially playing' shall be deferred until all other HTML5 media elements have paused and released their hardware resources. HTML5 media elements that are forced to pause shall emit a "pause" event and set the "paused" attribute to true.

NOTE <u>2</u>:   The policy for managing hardware resources between instances of the HTML5 media element defined here (automatically releasing allocated hardware resources when a new request occurs) is intentionally the exact opposite of the policy defined for the A/V control and video/broadcast objects by the OIPF DAE specification [1] and refined by clause A.2.1 of the present document.

See clause A.2.1 of the present document for the policy for managing hardware resources between instances of the HTML5 media element and instances of the A/V control or video/broadcast objects.

Buffering of video in an HTML5 media element shall be possible while broadcast video is presented in a video broadcast object or by the terminal.

4.7.11 Unreasonably demanding a/v sync timing requirement

Clause 9.7.4 is modified as shown below using underline / strike-through markup.

9.7.4 Minimum synchronization accuracy

The minimum accuracy for multi-stream ~~and inter-device~~ synchronization for a given terminal <u>shall be as shown in Table 11b:</u>

	Ahead of video by no more than	**Behind video by no more than**
Audio	Largest of A or 35 ms	Largest of A or 50 ms
Subtitles	Largest of A or 20 ms	Largest of A or 20 ms
NOTE: A is defined to be the duration of 1/2 tick of any timeline selected (during the corresponding `initMedia-Synchroniser()` or `addMediaObject()` method call) by the HbbTV® application for any of the streams under the control of the `MediaSynchroniser` object.		

<u>Table 11b: Accuracy requirements for multi-stream synchronisation</u> ~~is the largest of:~~

- ~~10 ms (being the duration of 1/2 a frame of video at 50 fps);~~

- ~~the duration of 1/2 tick of any timeline selected (during the corresponding initMediaSynchroniser() or addMediaObject() method call) by the HbbTV®~~

~~application for any of the streams under the control of the MediaSynchroniser object;~~

- ~~the duration of 1/2 tick of the Synchronization timeline (see clause 13.4.3.2) if inter-device synchronization is being performed.~~

How this minimum accuracy applies to terminals performing multi-stream synchronization is defined in clause 10.2.8.1. <u>Minimum accuracy for multi-stream synchronisation between audio and video shall be met when audio and video are output via</u>

<u>•	an integrated display with integrated speakers or headphone output; or</u>

<u>•	a combined audio and video output (such as an HDMI) to downstream devices that render the audio and video without introducing any additional differential delay between audio and video.</u>

<u>In all other situations, minimum accuracy for multi-stream synchronisation between audio and video should be met with best-effort, such as when audio is output via SPDIF, Bluetooth or HDMI (e)ARC.</u>

<u>Minimum accuracy for multi-stream synchronisation between video and subtitles shall be met irrespective of how video is output by the terminal.</u>

<u>The minimum accuracy for inter-device synchronization for a given terminal is the largest of:</u>

<u>•	10 ms (being the duration of 1/2 a frame of video at 50 fps);</u>

<u>•	the duration of 1/2 tick of the Synchronization timeline (see clause 13.4.3.2).</u>

How this minimum accuracy applies to terminals while performing inter-device synchronization in the role of a master terminal is defined in clause 13.8.2.4.

How this minimum accuracy applies to terminals while performing inter-device synchronization in the role of a slave terminal is defined in clauses 13.8.3.4 and 13.8.3.5.

<u>Minimum accuracy for inter-device synchronisation shall be met for video when it is output via an integrated display and, separately, shall be met for audio when it is output via integrated speakers or headphone output. Minimum accuracy for inter-device synchronisation should be met with best-effort for audio and for video in all other situations.</u>

<u>NOTE:	HDMI 2.0 [i.7] provides functionality for dynamic synchronization of video and audio streams. Information from the HDMI can be used to make a best-effort estimate of the extra travel time between a set-top box and the light and sound output of the TV screen.</u>

4.7.12 Crashes after a number of transitions from broadcast to broadband and back again

The text shown underlined is added to clause 9.8 as shown.

Terminals shall be able to present broadcast audio and video reliably when HbbTV applications are launching and stopping. Specifically:

- ▄ When the terminal launches a broadcast-related application and broadcast audio or video is being presented and that application does not try to control video playback, there shall not be any artifacts or glitches in presented broadcast audio or video. This includes applications delivered by broadband and DSM-CC object carousel, as well as both autostart and non-autostart applications.

- ▄ When the terminal terminates a broadcast-related application and broadcast audio or video is being presented and that application did not try to control video playback, there shall not be any artifacts or glitches in presented broadcast audio or video. This includes where the application calls `destroyApplication()` to exit itself, when application signalling causes the application to be stopped and when it is terminated manually by the user.

Terminals shall be able to reliably switch from presenting broadcast video, audio and subtitles to presenting broadband video, audio and then return to presenting broadcast video, audio and subtitles. Including specifically the following;

- ▄ Both where the broadband content is played to the end and where playback is stopped before the end is reached

- ▄ Including where the broadband content has subtitles (and the user has enabled subtitles), where the broadband content has subtitles (but the user has disabled subtitles) and where it does not have subtitles.

- ▄ Both where the broadband content is preloaded using the load method and where it is played without that method being used.

Terminals shall be resilient to transient error conditions that are likely to occur, as well as to conditions of low resource availability. Specifically, the terminal shall remain responsive to channel change and application termination requests in the following circumstances:

4.7.13 Equivalence between an InbandEventStream and an EventStream

The underlined text is added to clause 9.3.2.2 as shown.

It is recommended not to use the same @schemeIdUri and @value for MPD events and for inband events. Some implementations may create two TextTrack objects that cannot be distinguished by an application. Other implementations may create a single TextTrack object with the union of the MPD events and the inband events. Other possibilities may exist.

DASH events shall be reported to applications as DataCues according to the following mapping:

4.7.14 multiple video decoders with differing capabilities

The following text is added to clause 9.6.2.

Where a terminal has more than one video decoder that can decode a piece of content and is available, it is implementation dependent how the terminal decides which video decoder to

use. If the terminal needs to select between video decoders that have different capabilities then terminals should attempt to use the more restrictive or less flexible of the available decoders for adverts. How the terminal determines what is an advert is implementation specific however one possibility is to assume that content where the duration is known and less than an arbitrary value (e.g. 60s) is an advert and content with a longer duration or no known duration is not an advert.

> EXAMPLE: A terminal has two video decoders, one without restrictions and one with restrictions including lack of support for UHD, HDR, DRM and without implementer-specific picture improvement algorithms. When an application plays a piece of content with a duration of 1 hour, the unrestricted decoder is selected. When an application plays a pre-roll advert with a duration of 30s followed by a piece of content with a duration of one hour, if terminal decides to use the restricted decoder for the pre-roll advert and the unrestricted decoder for the content then a faster and cleaner switch from the advert to the content may result.

4.7.15 Consistency between HbbTV-TA and HbbTV 2.0.1/2/3 re multiple decoders and ad insertion/replacement

The bulleted list in clause 9.6.3 introduced by "The delay between the end of presentation of an HTML5 media element and starting presentation of another HTML5 media element shall be less than 250 ms if all of the following conditions are met:" is extended with the following two points;

- at most one video element refers to content where the video is UHD

- any UHD video does not exceed 30fps.

4.7.16 Undefined MediaSynchroniser behaviour for HTML5 media elements if source is changed

In clause 9.7.1.2, "e.g." and " or changing the src property" are inserted as shown underlined.

> If the HTML5 media element source is reloaded (e.g. by the application calling the load() method, or changing the src property), or the application sets an HTML5 MediaController for this media element, then:

4.7.17 Media Synchronisation

The text shown underlined is added in clause 9.7.1.2;

> If either an error occurs while fetching the media data (and hence an error event is triggered) or the current playback position of the media element reaches the end of the media resource (see "ended playback" as defined in HTML5 []), then:
> - o If it represents master media (it was passed to a MediaSynchroniser via the init-MediaSynchroniser() method) then this shall result in a permanent error of the MediaSynchroniser (see clause 13.3.8) with an error being dispatched with error number 14.

o If it represents other media (it was added to a `MediaSynchroniser` via the `addMediaObject()` method) then this shall result in a transient error of the `MediaSynchroniser` (see clause 13.3.7) and the object shall be removed as if an application had called the `removeMediaObject()` method and an error event dispatched with error 2.

If the media element was added to a `MediaSynchroniser` via the `addMediaObject` method, the time in the media timeline of the media element corresponding to the current time of the master media object shall be determined using the correlationTimestamp (if any).

o If the time in the media element timeline begins, or enters into, the period that is before the "earliest possible position" in the media resource (as defined in HTML5 []) then this shall result in a transient error of the `MediaSynchroniser` (see clause 13.3.7) and an error event dispatched with error 11. Synchronization of the HTML5 media element to the media object representing the master media shall resume automatically when (if) the time in the master media advances such that the corresponding time in the timeline of the media element reaches the earliest possible position in the media resource.

o If the time in the media element timeline begins, or enters into, the period that is after the end of the media resource (as defined in HTML5 []) then this shall result in a transient error of the `MediaSynchroniser` (see clause 13.3.7) and the object shall be removed as if an application had called the `removeMediaObject()` method and an error event dispatched with error 2.

4.8 Clause 10 – Capabilities

4.8.1 Removing support for CSS3 navigation

Text is added in clause 10.2.2.1 as shown underlined.

On up, down, left, right keydown events, terminals shall choose one of the following navigation mechanisms in the priority order listed below:

- Allow applications to capture the events and prevent the default action (known as "JavaScript navigation").
- Handle CSS3 directional focus navigation when the nav-up, nav-right, nav-down and nav-left CSS properties are used by the application.
  NOTE: Terminal support for these will be removed in a future version of the present document. Their use by application authors is discouraged. See Annex L.
- A default navigation mechanism provided by the terminal which shall allow focus to be moved between navigable elements and allow all navigable elements to gain focus.

4.8.2 XML Parsing

In clause 10.2.4, the following new paragraphs are added at the end of the clause.

The "doctype" property of the "xmlCapabilities" property of the "application/oipfCapabilities" embedded object shall be null.

NOTE: This indicates that there is no Document Type Definition for the xmlCapabilities.

4.8.3 Subtitle track selection - ability for an app to suppress subtitles reliably

The following is inserted in clause 10.2.7.1 after the paragraph "The terminal shall always perform…" and before the paragraph "The set of components that are available…":

The following figure shows a logical model for the component selection controls for subtitles, illustrating how user settings and application APIs interact with the state maintained by the terminal:

User language
preferences
(see 10.2.7.2)

User subtitle
enable/disable
(see 10.2.7.2)

Subtitle
decode
r

Control
"S1"

Control "E1"
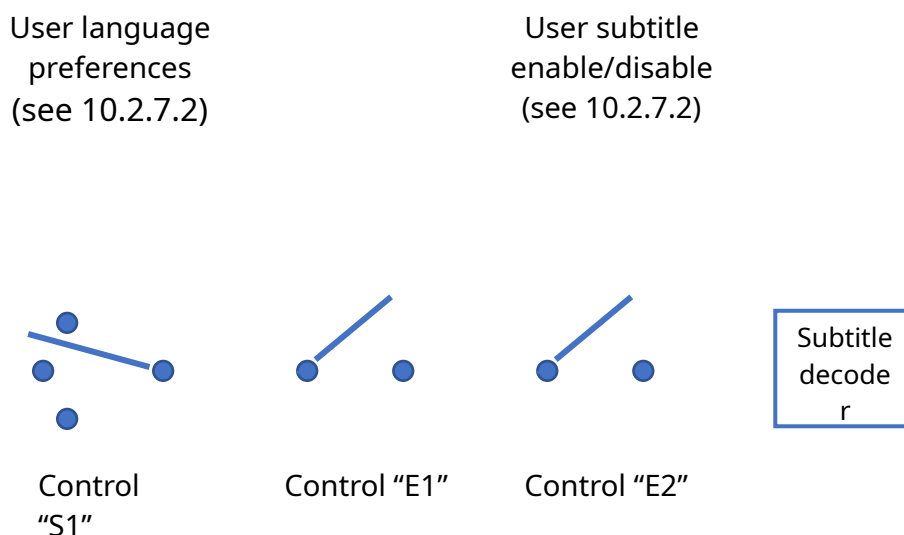
Control "E2"

Application control as described
in 10.2.7.3 including via HTML5,
video/broadcast and A/V Control APIs

Selection control **S1** determines which of the available subtitle streams is selected for possible presentation. At any one time, this control can be is either under the control of the terminal, according to user language preferences (see clause 10.2.7.2) or it can be under application control and influenced by the relevant component selection APIs (see clause 10.2.7.3). The circumstances under which control passes from one to the other are described in clause 10.2.7.3.

Enabling control **E1** is the application's control over whether subtitles are presented. It is 'closed', enabling subtitles, unless an application is running and all available subtitle components are deselected (via the `unselectComponent` method of a video/broadcast or A/V control object, or by setting the mode attribute of all TextTracks linked to an HTML5 media element to 'disabled' or 'hidden'). The user does not have any ability to control this setting directly. Moreover, this control is influenced solely by the application state and the action of application APIs and can may be 'open', disabling subtitles, even when component selection is under the control of the terminal (as in clause 10.2.7.2).

Enabling control **E2** is the user's control over whether subtitles are presented. It can may be changed when the user changes terminal subtitle preferences. An HbbTV Application does not have any ability to control this setting but its current setting can may be read using the

`subtitlesEnabled` property of the Configuration class (see A.2.20.1).

Controls **S1** and **E2** are separate logical controls but may be operated through a combined terminal user interface that offers both language choices and an option to select no subtitles.

Applications may influence both **S1** and **E1** and a single API call may cause either or both of these controls to change. The current state of components as seen by the application shall correctly reflect the state of both **S1** and **E1**. That is, if **E1** is 'open', the relevant APIs and events shall indicate that no component is selected for presentation, regardless of the current state of **S1**.

Clause 10.2.7.3 is changed as shown using underline / strikethrough markup.

The terminal shall maintain such changes made by an application until one of the following occurs:

…

🎬 a component, selected by the application, is being presented and is part of a video/ broadcast object or an A/V Control object or an HTML5 media element or a Media-Synchroniser object (as appropriate) which is destroyed:

in which case component selection for that component type shall revert to the control of the terminal;

🎬 a component of a particular type is being presented and the user makes a change using the terminal's subtitle/audio description (or other) selection mechanism relating to that component type:

- in which case component selection for that component type shall revert to the control of the terminal;

NOTE: Where an application has explicitly disabled presentation of a particular component type, changes to terminal preferences do not override this. Applications may disable presentation by means of the unselectComponent(Integer componentType) method of the video/broadcast or A/V control object, or by deselecting tracks in an HTML5 media element using the `enabled` attribute of an `AudioTrack` or `VideoTrack` or by setting the `mode` attribute of a `TextTrack` to "disabled" or "hidden".

4.8.4 Update inter-device sync spec from DVB blue book back to ETSI

In clause 10.2.8.1, the reference to "DVB Bluebook A167-2 [47]" is replaced by "ETSI TS 103 286-2 [47]".

4.8.5 Cookie writing to persistent storage time requirement

In clause 10.2.1, in table 11, "Minimum terminal capabilities", the following text is added to the "value" column of the rows for "Cookie support" and "Web Storage";

Terminals shall write data to persistent storage within 5 minutes of the the terminal being put into standby. Terminals should write data to persistent storage soon after that data has been set or modified, e.g. within 30 seconds.

4.8.6 Web Storage minimum storage requirement

In clause 10.2.1, in table 11, "Minimum terminal capabilities", the following text is added to the "value" column of the row for "Web Storage";

> Terminals shall support at least 8 Mbytes of storage overall, with at least 1 Mbyte being available to any individual domain, subject to sufficient overall space remaining.

4.8.7 Keeping "supportsPointer" read only and false for not-activated applications

In clause 10.2.2.2 is changed as shown using underline / strike-through markup.

> Applications shall not rely on receiving any mouse or wheel events <u>unless</u> they have ~~not requested~~ <u>indicated that they support a pointer based interaction model</u> by using the Keyset.supportsPointer property in clause 7.2.5.2 of the OIPF DAE specification [1].

4.8.8 Issues with HbbTV apps selecting inband TTML subtitles

In clause 10.2.7.3, the following text is changed as shown using underline / strike-through markup.

~~Applications may~~<u>Terminals shall allow applications to </u>change the terminal-derived component selection and discover the presentation status using the methods defined in clause 7.16.5 of OIPF DAE [1] and in clauses 4.7.10.10.1 and 4.7.10.12.5 of HTML5 [54].

4.8.9 Alpha blending in HbbTV

Clause 10.1 (which contains only one sentence, "This clause is replaced by annex H, "Display Model" of the OIPF DAE specification [1].") is replaced with the following.

> This clause is replaced by annex H, "Display Model" of the OIPF DAE specification [1] except as follows:

> As defined in CSS3 Color [i.22], compositing of graphical elements shall, by default, be performed according to the "Simple alpha compositing" rules. This shall include compositing graphics over video, whether from a video/broadcast object, an A/V control object or an HTML5 video element.

> NOTE 1:   Compositing rules other than "Simple alpha compositing" can be requested by the application using CSS Compositing and Blending [i.23]. Support for this is not required in HbbTV terminals.

> NOTE 2:   The compositing equations defined in CSS produce pixels represented in "alpha pre-multiplied form", whereby the R, G and B colour values of the pixel are each "pre-multiplied" by the pixel's resulting alpha value. For example, a white pixel with 50% opacity on a 32-bit graphics plane would be represented by pixel values of (127,127,127,127) in pre-multiplied form, and not (255,255,255,127). This representation of pixels is commonly used for alpha blending to avoid the need for division. However, care is needed to account for this when compositing a graphics plane over a video plane. If the graphics plane pixels are left in alpha pre-multiplied form, then this must be taken into account when the graphics plane is combined with the video plane. Typically, this will involve setting an appropriate configuration on a hardware compositor.

EXAMPLE 1:   A video is playing full screen and an opaque image covers the right-hand half of the picture such that the displayed colours match. When a semi-transparent graphics-plane element is placed over both the video and the image, the colour also matches.

The following figure illustrates this with black video and an opaque black image:
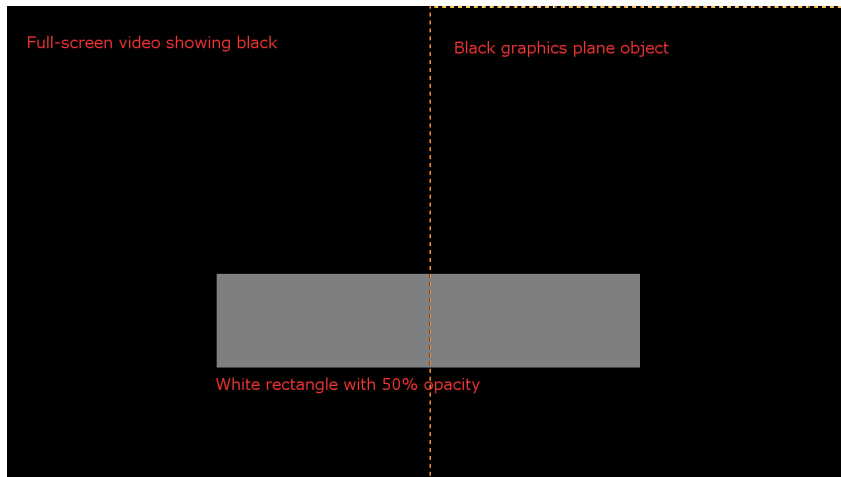
The following figure shows a possible incorrect rendering in which the graphics and video have been composited without taking into account that the graphics plane is in alpha pre-multiplied form:
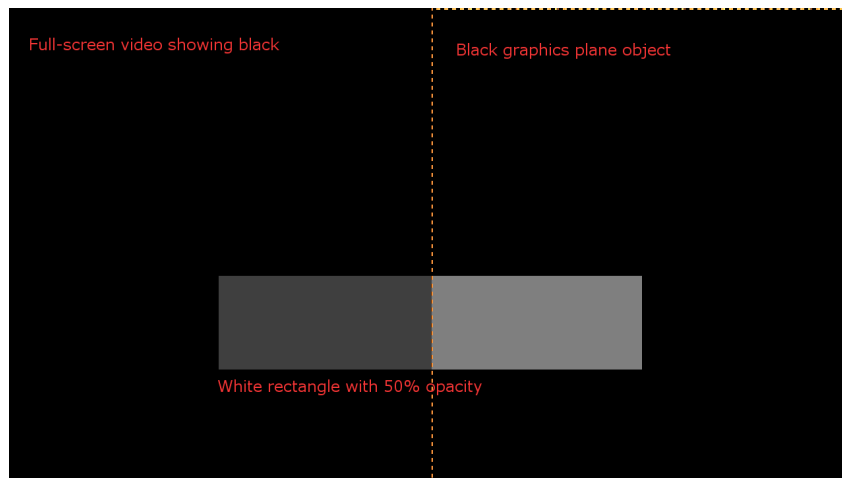
4.8.10 Incompatibility with modern soft input (virtual keyboards) that operate on words or phrases

In clause 10.2.1, in the table "Minimum terminal capabilities", text is added as shown underlined.

| Text entry method | Either multi-tap (e.g. as defined in ETSI ES 202 130 [i.2]) or an equivalent (e.g. software keyboard) where characters are input character by character in the text field. | The input-format CSS property may be used by terminals to determine which text entry method to use. Multi-tap also known as SMS- |

	For multi-tap or other methods which use multiple supported key events to generate a single character, these intermediate key events shall not be reported to applications. Only the final character result shall be reported to applications. <u>For speech-to-text or autocomplete / predictive text in virtual keyboards, terminals are not required to generate any key events. Instead an HTMLElement change event shall be generated when the contents of the text field is updated.</u>	tap is not to be confused with T9 text entry which is not required.

4.8.11 Media Synchronisation

The following text is added to the end of clause 10.2.7.1.

Components in media objects attached to to the `MediaSynchroniser` using `addMediaObject` shall not be available for component selection if any of the following apply;

- If the `MediaSynchroniser` is in a transient error state

- If no media can be presented at the currentTime of the a `MediaSynchroniser`

- If the component cannot be presented by the terminal for any reason

4.8.12 XML Namespaces and error in XML capabilities example and XSD

Clause 10.2.4 is modified as shown below using underline / strike-through markup. The modifications shown are for HbbTV 2.0.1 and for the text that HbbTV 2.0.2 inherits from HbbTV 2.0.1. Text specific to HbbTV 2.0.2 is omitted.

10.2.4 HbbTV® reported capabilities and option strings

10.2.4.1 General structure

The `xmlCapabilities` property of the `application/oipfCapabilities` embedded object shall describe an XML document that conforms to the schema defined in clause A.2.15. <u>The root element shall be `<profilelist>`.</u>

10.2.4.2 Forward compatibility – Terminals

<u>That XML document shall also conform to that schema with all the following modifications:</u>

1. <u>Optionally update the schema to any newer version of that schema in any standard released by ETSI. The version of schema to be used can be chosen by the terminal manufacturer, although testing may also use any newer version of the schema that is strictly backward compatible.</u>
2. <u>In the schema, remove all instances of `<xs:any>` or `<xs:anyAttribute>` that have namespace="##targetNamespace"</u>

3. In the schema, change all instances of `<xs:any>` or `<xs:anyAttribute>` that have namespace="##any" to have namespace="##other".
   NOTE 1: The above schema modifications are only for testing HbbTV terminals when it is certain that the latest version of the schema is being used. They breaks forward-compatibility with newer terminals that implement newer versions of the schema. However, when conformance testing a terminal, the tester can ensure they are using the latest version of the schema, forward-compatibility is not needed, and any use of not-yet-defined XML elements is an error. The normal, unmodified schema will cause XML elements defined in a newer version of the schema to be ignored, which is what everyone else needs.

   NOTE 2: Assuming HbbTV ensures that each new schema version is strictly backward compatible with all previous versions, this allows testing to always use the latest version of the schema. The wording is carefully chosen so that if HbbTV accidentally or deliberately releases a non-backward-compatible schema, existing TVs do not suddenly become non-compliant.

10.2.4.3   Forward compatibility – Applications

HbbTV applications may assume that newer versions of the schema will use the same XML namespace name, and will be defined such that any XML document that validates against any released version of the schema will also validate against the latest version of the XML schema.
   NOTE: However, if a bug is found before a new schema version has been widely implemented, HbbTV may publish an errata that breaks compatibility only with that specific version of the schema.

Newer versions of the schema may:
- add new elements to be used in the `<ext>` element, and/or

- add new elements to be used at the end of the `<profilelist>` element, after all the currently-allowed elements, and/or

- add new attributes where the schema types include an `<anyAttribute>` element.

HbbTV applications need to ignore any such elements and attributes that were not specified in the schema version they support.

The order of elements in the `<ext>` element may vary; applications should look for the specific element they want and shall not assume it is at any particular index in the `<ext>` element.

In the `<profilelist>` element:
- the order of the `<audio_profile>` elements in the group of `<audio_profile>` elements may vary

- the order of the `<video_profile>` elements in the group of `<video_profile>` elements may vary

- the order of the `<drm>` elements in the group of `<drm>` elements (if any) may vary

Applications should look for the specific element they want, and shall not assume it is at any particular index in the group.

The `<html5_media>` and `<drm>` elements may appear either in the `<ext>` element or in the `<profilelist>` element or both. Applications should check both locations.

10.2.4.4   Third Party Extensions

Anyone may create separate XML schemas that specify additional elements or attributes. Those schemas must use a different namespace name from the HbbTV-defined namespace. They may be used as follows:

- Third-party elements may be used directly inside the `<ext>` element

- Third-party attributes may be added to the root `<profilelist>` element.

In both cases, use of a namespace prefix is required. For example:

```
<profilelist xmlns="urn:hbbtv:config:oitf:oitfCapabilities:2017-1"
             xmlns:examplebroadcaster="http://example.com"
             xmlns:broadcastergroup="http://example.org"
             examplebroadcaster:specialFeature="true">
    <ui_profile name="OITF_HD_UIPROF+DVB_S+TRICKMODE">
        <ext>
            <broadcastergroup:numberOfWidgets>99</broadcastergroup:numberOfWidgets>
            <parentalcontrol schemes="dvb-si">true</parentalcontrol>
            <temporalClipping />
            <examplebroadcaster:greeting prime="3" language="English">
                <examplebroadcaster:short_message>
                    Hi
                </examplebroadcaster:short_message>
                <examplebroadcaster:long_message>
                    Hello, World.
                </examplebroadcaster:long_message>
            </examplebroadcaster:greeting>
            <examplebroadcaster:spec1supported />
            <clientMetadata type="dvb-si">true</clientMetadata>
            <examplebroadcaster:numberOfWidgets>42</examplebroadcaster:numberOfWidgets>
        </ext>
    </ui_profile>
    <audio_profile name="MPEG1_L3" type="audio/mpeg" />
    <audio_profile name="HEAAC" type="audio/mp4" />
    <audio_profile name="MP4_HEAAC" type="audio/mp4" transport="dash" sync_tl="dash_pr" />
    <video_profile name="MP4_AVC_SD_25_HEAAC" type="video/mp4" transport="dash"
sync_tl="dash_pr" />
    <video_profile name="MP4_AVC_HD_25_HEAAC" type="video/mp4" transport="dash"
sync_tl="dash_pr" />
    <video_profile name="MP4_AVC_SD_25_HEAAC_EBUTTD" type="video/mp4" transport="dash"
sync_tl="dash_pr" />
    <video_profile name="MP4_AVC_HD_25_HEAAC_EBUTTD" type="video/mp4" transport="dash"
sync_tl="dash_pr" />
    <video_profile name="TS_AVC_SD_25_HEAAC" type="video/mpeg" sync_tl="temi" />
    <video_profile name="TS_AVC_HD_25_HEAAC" type="video/mpeg" sync_tl="temi" />
    <video_profile name="MP4_AVC_SD_25_HEAAC" type="video/mp4" />
    <video_profile name="MP4_AVC_HD_25_HEAAC" type="video/mp4" />
    <html5_media>true</html5_media>
</profilelist>
```

HbbTV applications shall ignore such elements and attributes unless they are designed to understand the specific namespace that is used.

> NOTE: As a consequence of clause 10.2.4.2, third party XML elements may not be used anywhere except in the `<ext>` element, and third party XML attributes may not be used on any HbbTV-defined XML element.

10.2.4.5   Namespaces

The root element of the XML document shall include the default namespace binding xmlns="urn:hbbtv:config:oitf:oitfCapabilities:2017-1". The XML document shall not bind any prefixes to that namespace. Any namespace bindings used for third-party extensions must be on the root element. Namespaces must not be bound, unbound or rebound on other elements.

> NOTE: Because of the rules above, the XML elements defined by HbbTV must not

have a namespace prefix. Also, all namespace prefixes are defined on the root element, and the namespace prefixes are constant throughout the whole document. Applications may take advantage of these facts to use simpler code.

10.2.4.6   Document Type Definition

The "doctype" property of the "xmlCapabilities" property of the "application/oipfCapabilities" embedded object shall be null.

> NOTE: This indicates that there is no Document Type Definition for the xmlCapabilities.

10.2.4.7   XML Contents

For a terminal supporting only the base level of features, the XML Document object provided by the xmlCapabilities property of the application/oipfCapabilities embedded object shall describe an XML document that when canonicalized according to the W3C XML Canonicalization specification [Error: Reference source not found] shall be equal to the canonicalized form of the following XML:

```
<profilelist xmlns="urn:hbbtv:config:oitf:oitfCapabilities:2014-1"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="urn:hbbtv:config:oitf:oitfCapabilities:2017 4-1 config-hbbtv-
oitfCapabilities.xsd">
  <ui_profile name="OITF_HD_UIPROF+DVB_S+TRICKMODE">
  <ext>
    <parentalcontrol schemes="dvb-si">true</parentalcontrol>
    <clientMetadata type="dvb-si">true</clientMetadata>
    <temporalClipping />
  </ext>
  </ui_profile>
  <audio_profile name="MPEG1_L3" type="audio/mpeg"/>
  <audio_profile name="HEAAC" type="audio/mp4"/>
  <audio_profile name="MP4_HEAAC" type="audio/mp4" transport="dash" sync_tl="dash_pr"/>
  <video_profile name="MP4_AVC_SD_25_HEAAC" type="video/mp4" transport="dash"
             sync_tl="dash_pr" />
  <video_profile name="MP4_AVC_HD_25_HEAAC" type="video/mp4" transport="dash"
             sync_tl="dash_pr" />
  <video_profile name="MP4_AVC_SD_25_HEAAC_EBUTTD" type="video/mp4" transport="dash"
             sync_tl="dash_pr" />
  <video_profile name="MP4_AVC_HD_25_HEAAC_EBUTTD" type="video/mp4" transport="dash"
             sync_tl="dash_pr" />
 <video_profile name="TS_AVC_SD_25_HEAAC" type="video/mpeg" sync_tl="temi" />
  <video_profile name="TS_AVC_HD_25_HEAAC" type="video/mpeg" sync_tl="temi" />
  <video_profile name="MP4_AVC_SD_25_HEAAC" type="video/mp4" />
  <video_profile name="MP4_AVC_HD_25_HEAAC" type="video/mp4" />
  <html5_media>true</html5_media>
</profilelist>
```

"DVB_S" shall be replaced by the appropriate string(s) for the supported broadcast delivery system(s).

Other parental control schemes in addition to "dvb-si" may be listed in the <parentalcontrol> element.

Only the video format profiles supported for broadband shall be listed.

As mentioned in Error: Reference source not found, the terminal may also support E-AC3 audio, in which case the following elements shall be added after the elements listed in the <profilelist> element in the above XML:

```
<audio_profile name="MP4_E-AC3" type="audio/mp4" />
<audio_profile name="MP4_E-AC3" type="audio/mp4" transport="dash" sync_tl="dash_pr" />
<video_profile name="TS_AVC_SD_25_E-AC3" type="video/mpeg" sync_tl="temi" />
<video_profile name="TS_AVC_HD_25_E-AC3" type="video/mpeg" sync_tl="temi" />
<video_profile name="MP4_AVC_SD_25_E-AC3" type="video/mp4" />
<video_profile name="MP4_AVC_HD_25_E-AC3" type="video/mp4" />
```

```
<video_profile name="MP4_AVC_SD_25_E-AC3" type="video/mp4" transport="dash" sync_tl="dash_pr" />
<video_profile name="MP4_AVC_HD_25_E-AC3" type="video/mp4" transport="dash" sync_tl="dash_pr" />
<video_profile name="MP4_AVC_SD_25_E-AC3_EBUTTD" type="video/mp4" transport="dash"
          sync_tl="dash_pr" />
<video_profile name="MP4_AVC_HD_25_E-AC3_EBUTTD" type="video/mp4" transport="dash"
          sync_tl="dash_pr" />
```

Terminals that support HEVC UHD video as defined in clause 7.3.1.3 shall include the following video profiles:

```
<video_profile name="MP4_HEVC_UHD_25_HEAAC_EBUTTD" type="video/mp4" transport="dash"
          sync_tl="dash_pr"/>
<video_profile name="MP4_HEVC_UHD_25_HEAAC_EBUTTD" type="video/mp4" />
```

and, if E-AC3 audio is supported in the broadcast channel, shall additionally include the following video profiles:

```
<video_profile name="MP4_HEVC_UHD_25_E-AC3_EBUTTD" type="video/mp4" transport="dash"
          sync_tl="dash_pr"/>
<video_profile name="MP4_HEVC_UHD_25_E-AC3_EBUTTD" type="video/mp4" />
```

Terminals that support 8-bit HEVC HD video and not 10-bit HEVC HD video as defined in clause 7.3.1.3 shall include the following video profiles:

```
<video_profile name="MP4_HEVC_HD_25_8_HEAAC_EBUTTD" type="video/mp4" transport="dash"
          sync_tl="dash_pr"/>
<video_profile name="MP4_HEVC_HD_25_8_HEAAC_EBUTTD" type="video/mp4" />
```

and, if E-AC3 audio is supported in the broadcast channel, shall additionally include the following video profiles:

```
<video_profile name="MP4_HEVC_HD_25_8_E-AC3_EBUTTD" type="video/mp4" transport="dash"
          sync_tl="dash_pr"/>
<video_profile name="MP4_HEVC_HD_25_8_E-AC3_EBUTTD" type="video/mp4" />
```

Terminals that support 10-bit HEVC HD video as defined in clause 7.3.1.3 shall include the following video profiles:

```
<video_profile name="MP4_HEVC_HD_25_10_HEAAC_EBUTTD" type="video/mp4" transport="dash"
          sync_tl="dash_pr"/>
<video_profile name="MP4_HEVC_HD_25_10_HEAAC_EBUTTD" type="video/mp4" />
```

and, if E-AC3 audio is supported in the broadcast channel, shall additionally include the following video profiles:

```
<video_profile name="MP4_HEVC_HD_25_10_E-AC3_EBUTTD" type="video/mp4"
          transport="dash" sync_tl="dash_pr"/>
<video_profile name="MP4_HEVC_HD_25_10_E-AC3_EBUTTD" type="video/mp4" />
```

The strings defined in Table 1 shall be used to indicate which options are supported by a terminal. They shall be used:

- In the HTTP User-Agent header for applications data retrieval through HTTP.

- In the ui_profile element's name property of the xmlCapabilities property of the application/oipfCapabilities embedded object.

- As parameters of the hasCapability() method of the application/oipfCapabilities embedded object to dynamically query the options supported by the terminal.

  NOTE: Some of the strings defined in the clause intentionally match with the "UI Profile Name Fragment" strings defined in the OIPF DAE specification [Error: Reference source not found].

Table 1: HbbTV® Option Strings

Option string	Meaning
"+DL"	Support for file download feature.
"+PVR"	Support for PVR feature.
"+DRM"	Support for the DRM feature - specifically that the XML capabilities include a <drm> element as defined below (see note).
"+SYNC_SLAVE"	Support for the terminal behaving as a slave terminal for inter-device synchronization (see clause 10.2.9).
"+IPH"	Support for the "IP delivery Host player mode" as defined in the DVB Extensions to CI Plus ETSI TS 103 205 [Error: Reference source not found].
"+IPC"	Support for the "IP delivery CICAM player mode" as defined in the DVB Extensions to CI Plus ETSI TS 103 205 [Error: Reference source not found].
"+AFS"	Support for the CICAM Auxiliary File System as defined in the DVB Extensions to CI Plus ETSI TS 103 205 [Error: Reference source not found].
NOTE:	"+DRM" has a specific meaning in OIPF which it does not have in the present document.

The support of the DRM feature shall be indicated by the addition of one or more <drm> elements ~~in the OIPF extension to the <profilelist> element as defined in annex F of the OIPF DAE specification [Error: Reference source not found]~~ to the end of the <profilelist> element in the above XML. For example:

    <drm DRMSystemID="urn:dvb:casystemid:12345">TS_PF</drm>

The support of one or more CA systems on a CICAM shall be indicated using the <drm> element defined in annex F of the OIPF DAE specification [Error: Reference source not found] and providing the protectionGateways attribute with "ci+" string. All of the CA systems exposed by the CICAM using the ca_info APDU shall be listed in this way. For example:

    <drm DRMSystemID="urn:dvb:casystemid:12345" protectionGateways="ci+">TS_PF</drm>

Terminals that support the "IP delivery Host player mode", as defined in the DVB Extensions to CI Plus ETSI TS 103 205 [Error: Reference source not found], shall expose the DRMs supported by any compliant CICAMs inserted in the terminal by using the <drm> element as described above. In this case, the protection format shall be "MP4_CENC" although others may also be listed where appropriate. All of the DRM systems exposed by the CICAM using the sd_info_reply APDU shall be listed in this way. This implies that the CICAM shall identify a supported DRM by filling in the drm_system_id field in the sd_info_reply APDU. The URN string describing the DRMs shall be suffixed by ":cicam". For example:

    <drm DRMSystemID="urn:dvb:casystemid:12345:cicam">MP4_CENC</drm>

~~The "doctype" property of the "xmlCapabilities" property of the "application/oipfCapabilities" embedded object shall be null.~~

~~NOTE: This indicates that there is no Document Type Definition for the xmlCapabilities.~~

10.2.4.8   Option strings

The strings defined in Table 1 shall be used to indicate which options are supported by a terminal. They shall be used:

- 🎬 In the HTTP User-Agent header for applications data retrieval through HTTP.

- In the `ui_profile` element's `name` property of the `xmlCapabilities` property of the `application/oipfCapabilities` embedded object.

- As parameters of the `hasCapability()` method of the `application/oipfCapabilities` embedded object to dynamically query the options supported by the terminal.

  NOTE: Some of the strings defined in the clause intentionally match with the "UI Profile Name Fragment" strings defined in the OIPF DAE specification [Error: Reference source not found].

Table 2: HbbTV® Option Strings

Option string	Meaning
"+DL"	Support for file download feature.
"+PVR"	Support for PVR feature.
"+DRM"	Support for the DRM feature - specifically that the XML capabilities include a `<drm>` element as defined below (see note).
"+SYNC_SLAVE"	Support for the terminal behaving as a slave terminal for inter-device synchronization (see clause 10.2.9).
"+IPH"	Support for the "IP delivery Host player mode" as defined in the DVB Extensions to CI Plus ETSI TS 103 205 [Error: Reference source not found].
"+IPC"	Support for the "IP delivery CICAM player mode" as defined in the DVB Extensions to CI Plus ETSI TS 103 205 [Error: Reference source not found].
"+AFS"	Support for the CICAM Auxiliary File System as defined in the DVB Extensions to CI Plus ETSI TS 103 205 [Error: Reference source not found].
NOTE: "+DRM" has a specific meaning in OIPF which it does not have in the present document.	

4.9 Clause 11 – Security

4.9.1 Update reference to TS 102 809

The following paragraph is added at the end of clause 11.1.

Security for broadband-delivered applications is provided through TLS as described below. Some security for broadcast-delivered applications and broadcast application signalling is provided by the inherent difficulty in modifying broadcast signals in a way that impacts a significant number of people. More security may be provided using the protection mechanism defined in clause 9 of TS 102 809 [3], see clauses 7.2.2 and 7.2.3.1.

4.9.2 SHA-1 sunset date has passed

The following changes are made in clause 11.2:

- The sentence "Terminals shall cease to trust any signature that uses SHA-1 as the digest algorithm after 31st December 2016." is removed.

- The two sentences "Mandatory until forbidden by SHA-1 sunset requirement specified above." are both replaced by "Forbidden".

4.9.3 Inconsistency on support for encrypted content over broadband

The following paragraph of clause 11.5 is amended as shown with underline and strike-through markup.

> For terminals that do not support the "IP delivery Host player mode" as defined in the DVB Extensions to CI Plus ETSI TS 103 205 [37], the only requirement for supporting for decryptiong of content delivered via the broadband channel is that in clause B.3 of optional in the present document. When decryption is supported via the integration of HbbTV® with one or more embedded content protection technologies, the terminal shall support at least the ISO base media file format using MPEG common encryption as defined by CENC ISO/IEC 23001-7 [30] and constrained by annex B of the present document as a format for encrypted content.

4.10 Clause 12 – Privacy

4.10.1 deviceId

Clause 12.1.5 is modified as shown using underline / strike-through markup.

12.1.5   ~~Unique device IDs~~Distinctive identifiers

> Terminals shall implement the extensions to the Configuration class for distinctive identifiers as defined in A.2.20.5 and as required in this clause. These extensions support a per-origin, non-associable, user-clearable identifier (these terms are used in W3C EME [66]).

> Terminals shall support the `deviceId` property in A.2.20.5 but may restrict the availability of the distinctive identifier. If the availability is restricted, the terminal shall implement one or more of the following:

> 1) O~~o~~ffer the user the option to enable or disable the availability of a ~~unique~~ distinctive identifier~~device ID (via the deviceId property defined in clause A.2.20.5 of the present document)~~ on a per-application or per organisation basis (e.g. as part of the device settings or installation menu). The availability of~~Access to~~ the distinctive identifier ~~device ID~~ should be enabled by default unless blocked due to local regulatory requirements.

> EXAMPLE: The EU General Data Protection Regulation (GDPR) could be considered as a "local regulatory requirement" which may result in some terminal manufacturers setting this option to disabled by default.

> 2) Display some native UI requesting the user to allow the terminal to make the distinctive identifier available to the application in response to a call to the `requestAccessToDeviceID` method.

> NOTE 1:   Some terminals may restrict the number of times that an application may call this method.

> 3) A manufacturer specific method for determining access to the distinctive identifier, for example by maintaining a list of those application providers where the application provider

and the terminal manufacturer have entered into a suitable agreement covering such availability.

4) Access to the distinctive identifier is blocked to applications until they have been activated as defined in clause 10.2.2.1.

NOTE 2: Hence access to the distinctive identifier by autostart broadcast-related applications will be denied until the user has, for example, pressed the red button.

Terminals that support the second option above shall support the `requestAccessToDistinctiveIdentifier` method in A.2.20.5.

NOTE 3: The `deviceId` property defined in clause A.2.20.5 is in the Configuration class. This should not be confused with the `deviceID` property in the LocalSystem class defined by the OIPF DAE specification [1], which is marked as "NI" in Table A.1 of the present document.

It shall not be possible to determine the identifier that would be presented to one origin or by a specific device, knowing the identifier that was generated for a different origin or ~~on~~ by a different device. The distinctive identifier ~~device ID~~ shall be generated by deriving an ID of at least 128 bits using a secure hash function from a combination of a device unique value that is not required to be secret (e.g. serial number), plus a common secret value (e.g. common to a manufacturer or model or product family), plus the origin of the HTML document (see clause 6.3.2), plus a value that changes each time the user requests that a new value of the identifier is generated (e.g. the time the user request was made).

It shall be possible for the user to generate a new but distinct value for the distinctive identifier ~~device ID~~.

NOTE 4: This mechanism is modelled on the IOS 7 mechanism [i.14] and on the Android advertising identifier [i.15].

4.10.2 WebStorage behaviour if user has disabled persistent storage

In 12.1.4, the following text is added at the end of the clause.

If the user has disabled persistent storage in this way then either (a) access to the localStorage attribute shall fail with a SecurityError exception or (b) a call to setItem on the localStorage object shall fail with a QuotaExceededError exception as defined in the Web Storage specification as referenced through the OIPF DAE specification [1]. Storage attempts shall not fail silently as a result of user preferences.

4.10.3 Distinctive identifiers: requestAccessToDeviceID method

In clause 12.1.5, the reference to requestAccessToDeviceID is replaced with a reference to requestAccessToDistinctiveIdentifier.

4.11 Clause 13 - Media synchronization

4.11.1 Avoid MediaSynchroniser API leaking "secret" stream URLs

Clause 13.6.2, "CSS-CII service endpoint (master terminal)", is changed as shown using underline / strike-through markup below.

CII messages sent by the master terminal via a connection to the CSS-CII service endpoint shall convey the following:

- 🎬 When the `contentIdOverride` property of the `MediaSynchroniser` object is (or is set to) a non-null value then the `contentId` and `contentIdStatus` properties of the CII message shall be overridden as follows:

  o the value of the `contentId` property shall be the value of `contentIdOverride`, and
  o the `contentIdStatus` shall be "`final`".

- 🎬 When `contentIdOverride` is (or is set to) undefined or `null` then no override takes place and t~~T~~he `contentId` and `contentIdStatus` properties shall correspond to the Content Identifier of the master media. For DVB broadcast services (and PVR re-cordings made from them) and MPEG DASH streams this shall be as defined in clause 5.2 of DVB Bluebook A167-2 [47]. For ISOBMFF and MPEG2 TS delivered via broadband:

  - the value of the `contentId` property shall be the absolute version of the URL provided by the HbbTV® application to specify the location of the media stream, before any redirect that may occur, and
  - the `contentIdStatus` shall be "`final`".

  NOTE 1: When playing back a PVR recording of a DVB broadcast service, the contentId represents the original broadcast. Although the contentId incorporates elements that come from components that are not necessarily recorded (e.g. NIT, BAT and SDT) these elements are considered pseudo static and therefore can be captured once during the recording process for inclusion in the contentId during playback.

  NOTE 2: The effect of an application setting the `contentIdOverride` property of the `MediaSynchroniser` is to prevent exposing the original content ID for the master media. If `contentIdOverride` is set before inter-device synchronisation is activated and remains set, then clients using this protocol will only ever see the value of `contentIdOverride` as the value of the `contentId` property in messages.

The other numbered notes in 13.6.2 are renumbered accordingly.

Clause 13.8.2.2, "Synchronization timeline availability", is changed as shown using underline / strike-through markup below.

4.11.1.113.8.2.2 Synchronization timeline availability

As the first stage of the protocol session, the MSAS function of the master terminal awaits a setup-data message from the slave terminal or CSA. This message requests the Synchronization Timeline to be used for the remainder of the protocol session. The Synchronization Timeline defines the reference frame for contentTime property values in Control Timestamps and Actual, Earliest and Latest Presentation Timestamps exchanged during the protocol session.

The requested Synchronization Timeline shall be available if the requirements for determining the availability defined in clause 9.7.3 of the present document and clause 9.2 of DVB Bluebook A167-2 [47] are met and the requested Timeline is supported by the

master terminal (see clause 13.4.2) and the master terminal has sufficient resources to decode the requested Timeline (see clause 13.4.2).

When the contentIdOverride property of the MediaSynchroniser object is (or is set to) a non-null value the value of this property overrides the content ID of the master media and shall be used in its place when determining availability according to the process defined in clause 9.2 of DVB Bluebook A167-2 [47]. When contentIdOverride is (or is set to) undefined or null, then no override takes place.

NOTE 1: The availability of the Synchronization Timeline is dependent on whether the contentIdStem matches the contentId for the master content (which might be overridden as described above) and whether the requested timeline is currently derivable for the master media.

4.11.2 Clarification of TEMI timeline requirements

In clause 13.4.2, the note shown underlined is added and the following notes renumbered accordingly.

DVB Bluebook A167-2 [47] defines support in the terminal for the decoding of MPEG-TS Timed External Media Information (TEMI) timeline descriptors in the adaptation field of Transport Stream packets carrying Packetized Elementary Streams (PES). Terminals shall support at least the following components of a DVB service to carry MPEG TEMI timeline descriptors:

- Any component that is supported by the terminal for use with media synchronization and MPEG TEMI, i.e. audio, video and subtitles.

- Any component with stream_type 6 (private PES) and stream_id 1011 1101 ("private_stream_1") in the PES packet header, including, but not limited to, components where the PES packet payloads are empty.

NOTE 2: The MPEG specification for TEMI (referenced via ETSI TS 103 286-2) defines carriage in adaptation fields of "media components". This is extended by the requirements above to include components with PES packets with empty payloads.

NOTE 3 : Selection of the correct timeline descriptors by component tag and timeline id is done via the timeline selector by using the media sync API as defined in clause 8.2.3. This also means that there can be different timelines present if applications use either multiple components or timeline ids or a combination of both.

4.11.3 Update inter-device sync spec from DVB blue book back to ETSI

In clause 13, all of the references to "DVB Bluebook A167-2 [47]" are replaced by "ETSI TS 103 286-2 [47]".

4.11.4 Clarify what timelineSelector to convey in CII message

In clause 13.6.2, the paragraph before table 21, "Primary aspect of presentationStatus when master media is a video/broadcast object", is extended as shown underlined.

While the `MediaSynchroniser` API timeline is available (see clause 9.7.3) the timelines

property shall convey a list where the first item in the list is a timeline options JSON object (as defined in clause 5.6 of TS 103 286-2 [47]) that describes the `MediaSynchroniser` API Timeline (as defined in clause 13.4.3). <u>To do this, the `timelineSelector` in the first item in the list shall by an exact string match for the `timelineSelector` passed as an argument to the `initMediaSynchroniser()` method.</u>

4.11.5 InitMediaSynchroniser

In clause 13.11.3, "If the timeline is not yet available (see clause 9.7.3) then the value returned shall be NaN. In all other situations," is added as shown underlined.

• For a MediaSynchroniser object that has been initialized by calling the initMediaSynchroniser() method, the returned value shall correspond to the current playback position of the media object that was passed as an argument to the initMediaSynchroniser() method (the master media). <u>If the timeline is not yet available (see clause 9.7.3) then the value returned shall be NaN. In all other situations,</u> the value returned shall be the time of the last video frame that was composed with graphics before the property was queried and shall be accurate to within 100 ms. The precision of the playback position shall be at least correlate with either:

4.11.6 Media Synchronisation

In clause 13.3.1, the figure is replaced with the following;
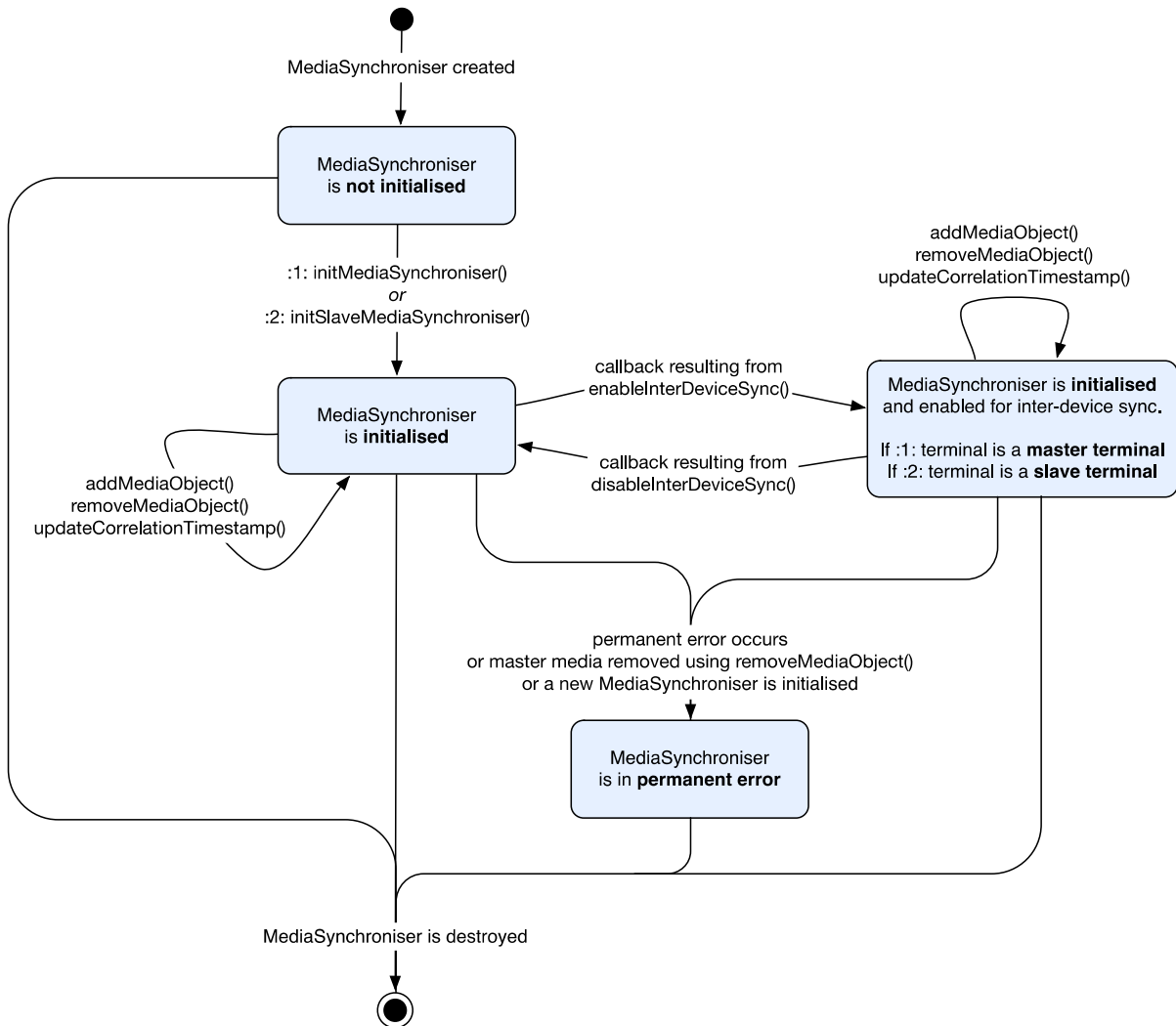
and the following text is inserted below.

> A `MediaSynchroniser` object that has been previously initialized can also be pushed into the permanent error state by calling `removeMediaObject` to remove the master media object from the `MediaSynchroniser`.

In clause 13.3.8, the following item is added to the list of "A permanent error of the MediaSynchroniser can occur if any of the following occurs:".

- <u>a call to</u> `removeMediaObject` <u>is used to remove the master media object from the</u> `Media-Synchroniser`;

4.12 Clause 14 – Companion screens

4.12.1 UPnP version to be used with DIAL

The start of clause 14.7.2 is extended with the text shown underlined.

> HbbTV® is a DIAL [50] application registered at the DIAL registry [i.8]. The registered name for HbbTV® applications is 'HbbTV'. For terminal and service endpoint discovery, the terminal shall support DIAL [50] <u>except that the response to an M-SEARCH request, as specified by section 5.2 of DIAL [50], may be compliant with section 1.2.2 of UPnP Device</u>

Architecture 1.0 [67] instead of section 1.3.3 of UPnP Device Architecture 1.1 [68].

NOTE: Section 1.3.2 of UPnP Device Architecture 1.1 [68] requires devices issuing an M-SEARCH request to be fully backwards compatible with previous versions.

The discovery response example in clause 14.7.3.1 is changed as shown using underline / strike-through markup below.

Discovery Response

A UPnP/1.0 compliant terminal responds with HTTP/1.1 OK, LOCATION header and DIAL ST:

```
HTTP/1.1 200 OK
CACHE-CONTROL: max-age = 1800
EXT:
LOCATION: http://192.168.1.11:50201/dial.xml
SERVER: Linux/2.6 UPnP/1.0 Sony-BDP/2.0
ST: urn:dial-multiscreen-org:service:dial:1
USN: uuid:00000004-0000-1010-8000-d8d43c1923dc::urn:dial-multiscreen-
org:service:dial:1
```

The A UPnP/1.1 compliant terminal respondses with HTTP/1.1 OK, and LOCATION header, and DIAL ST:

```
HTTP/1.1 200 OK
CACHE-CONTROL: max-age = 1800
EXT:
LOCATION: http://192.168.1.11:50201/dial.xml
SERVER: Linux/2.6 UPnP/1.1 Sony-BDP/2.0
BOOTID.UPNP.ORG: 1
ST: urn:dial-multiscreen-org:service:dial:1
USN: uuid:00000004-0000-1010-8000-d8d43c1923dc::urn:dial-multiscreen-
org:service:dial:1
```

4.12.2 Clarification on XML AIT example

In the example XML AIT in clause 14.6.2, "whizzo-app.html" is moved from the `URLBase` element to the `applicationLocation` element as shown with underline and strike-through markup.

```
<mhp:applicationTransport xsi:type="mhp:HTTPTransportType">
   <mhp:URLBase>http://www.example.com/whizzo-app.html</mhp:URLBase>
</mhp:applicationTransport>
<mhp:applicationLocation>whizzo-app.html?launch=from-cs</mhp:applicationLocation>
```

4.12.3 URL used in in DIAL example should be changed

In clauses 14.7.3.1 and 14.7.3.2, the references to

Origin: http://cs.services.broadcaster.com

are replaced with

Origin: http://cs.services.example.com/

4.12.4 Approval and pre-approval to launch an HbbTV app on request by a CS app

Clause 14.6.2 is modified in two places as shown below using underline / strike-through markup.

The terminal might have states where the feature is temporarily unavailable, e.g. during a channel scan. The states when the feature is not available are not defined by the present document. If the terminal rejects the application launch for this reason it shall respond with

the response code 503.

Terminals <u>shall support at least one of the following mechanisms for approvals or pre-approvals and</u> shall not, by default, launch applications without ~~at least one of the following~~ <u>such</u> approval~~s~~ or pre-approval~~s~~:

and

The terminal UI should provide means for the user to either approve or pre-approve application launching. This may include means for the user to accept or block requests from particular companion devices. If the terminal rejects the application launch because approval or pre-approval by the user was requested and denied, then it shall respond with the response code 403, where the body of the response is the 4 character string "USER" and has content type "text/plain".

<u>The terminal shall allow future applications to be launched by supporting either, or both, of:</u>

- <u>explicit user approval,</u>

- <u>and/or mechanism by which explicit pre-approvals can be updated.</u>

If the terminal rejects the request for reasons other than any of the above, then it shall respond with the response code 403, with an empty response body.

4.12.5 Broken cross-reference to 14.5.6

In clause 14.5.1, in the paragraph beginning "HbbTV® applications determine the location of the service endpoints using JavaScript APIs defined in clause 8.2.6.", the references to 14.5.4, 14.5.5 and 14.5.6 are replaced by references to 14.5.3, 14.5.4 and 14.5.5 respectively.

4.13 Annex A – OIPF DAE Specification Profile

4.13.1 Text referring to MetadataSearch in wrong location in annex A

The following text is in the wrong row in table A.1.

The count parameter of the findProgrammesFromStream method of the MetadataSearch class is not included.

It is moved from the row for the row for the "Basics" of the Programme classs to the row for "The MetadataSearch class".

4.13.2 "channels" should be "audioChannels"

In the clause A.2.4.6, in the definition of the createAVAudioComponent method, the channels argument is renamed to "audioChannels" and the three references to the "channels" property are replaced with a reference to the "audioChannels" property.

4.13.3 Clarify resource usage by HTML5 media elements

In clause A.2.1, the following paragraph is amended as indicated using underline / strike-through markup.

If the resources that would be needed by an A/V Control object or a video/broadcast object are <u>allocated to</u> ~~in use by~~ an HTML5 media element <u>(see clause 9.6.2)</u>, and the media element requiring the resource and the current media element owning the resource have not

been added to the same media synchronizer object, then the request to present media through the object shall fail. For an A/V control object, the object shall go to playState 6 with the error property being 3, "insufficient resources". For a video/broadcast object, this shall be reported by an onChannelChangeError with errorState 11, "insufficient resources are available to present the given channel (e.g. a lack of available codec resources)".

4.13.4 VK_RECORD key is not in any set of keys

In table A.1, in the row for "The Keyset class", the notes column is amended as indicated using underline / strike-through markup.

The Keyset class	7.2.5	M(*)	For terminals not making the VK_RECORD key event available to HbbTV applications, t~~T~~he otherKeys and maximumOtherKeys properties are not included. For terminals making the VK_RECORD key event available to HbbTV applications, the otherKeys and maximumOtherKeys properties shall be supported and applications shall be able to request the VK_RECORD key event using them.~~and~~ T~~t~~he getKeyLabel method ~~are~~ is not included. The icons returned by the getKeyIcon method shall be 32 x 32 pixels.	None

4.13.5 XML Parsing

1) In clause A.1 "Detailed section by section definition":

In the row "Content Access Download Descriptor Format", add to the "Notes" column:

```
The Content Access Download Descriptor shall not contain an XML Document
Type Definition ("<!DOCTYPE ...>").
```

In the row "Content Access Streaming Descriptor Format", add to the "Notes" column after the existing text:

```
The Content Access Streaming Descriptor shall not contain an XML Document
Type Definition ("<!DOCTYPE ...>").
```

2) In clause A.2.6.2 "MIME type and DOCTYPE", after the bulleted list, add a new paragraph:

```
The "doctype" declaration shall not contain an "intSubset" as that is defined in
the XML specification [69].
```

4.13.6 bindToCurrentChannel() while an A/V Control object is presenting

Clause A.2.1 is modified as shown using underline / strike-through markup.

A.2.1   Resource management

In clause 4.4.5 of the OIPF DAE specification [1], the STATIC_ALLOCATION model is not included in the present document. All resource allocation is under the DYNAMIC_ALLOCATION model.

Resource allocation between any number of A/V control objects and/or video/broadcast objects shall be based on a "first-come, first-served" policy. Resources shall not be taken away from one object of either of these types in order to meet a request on a second object of either of these types.

•   If the resources needed for the request on the second object (suitable video decoder, suitable audio decoder and, suitable tuner if the second object is a video/broadcast object, suitable tuner) are not available then the request on the second object shall fail as defined by the API for the type of object concerned.

•   If the resources needed for the request on the second object are available (e.g. the terminal has multiple audio and video decoders available to the HbbTV implementation) then the resources shall be allocated to the second object and the request shall not fail due to lack of resources (although it may fail for an another unrelated reason).

•   If the request on the second object succeeds then the terminal shall present both objects at the same time without synchronisation. If applications wish to have multiple objects present media with synchronisation then the objects need to be added to a MediaSynchroniser object.

NOTE 1:   Broadcast-related applications that wish to use a video/broadcast object and media decoders foralso wish to use broadband-delivered content need to put the video/broadcast object into the stopped state to release the media decoders. Calling the unselectComponent method on a video/broadcast object does not release the media decoder for that component type. Changing a video/broadcast object from a presenting a TV service to presenting a radio service should not release the video decoder. . Changing a video/broadcast object from a presenting a TV or radio service to presenting a data service (see clause 7.2.6 of the present document) should not release the video or audio decoder.

NOTE 12:   The policy for managing hardware resources defined here that applies to the A/V Control object and video/broadcast objects (first-come, first-served) is intentionally the exact opposite of the policy defined for the HTML 5 media element in clause 9.6.2 of the present document.

If a broadcast-related application that either

•   does not include a video/broadcast object at all or

•   includes a video/broadcast object that is in the unrealized state

attempts to start playing broadband-delivered video/audio then the presentation of the broadcast channel shall be suspended and allocation of the required media resources by the A/V control object shall succeed. After the A/V control object release the allocated resources, e.g. by stopping the media, presentation of the broadcast service shall resume.

NOTE 3: In spite of the above requirement, applications wishing to present only broadband-delivered video/audio should explicitly stop broadcast video/audio presentation in order to avoid implementation-dependent behaviour during the transition.

NOTE 4: The above requirement is unrelated to availability of video and audio decoder resources. Hence such applications will give the same user experience on terminals supporting multiple video and audio decoders as they do on terminals supporting only one decoder of each type. Applications wishing to simultaneously present broadcast-delivered video/audio and broadband-delivered video/audio need to create both a video/broadcast object and an A/V control object or HTML5 media element.

Clause A.2.4.1 is modified as shown. In this text, additions to TS 102 796 are shown in italics. Underlining is used to indicate text that is to be underlined in TS 102 796. Strike-through text in italics is to be included in TS 102 796 also with strike-through. Strike-through text not in italics indicates text to be deleted from TS 102 796.

•€€€€€€€€€€€ In clause 7.13.1.3 of the OIPF DAE specification [1], the definition of the bindToCurrentChannel() method is modified as shown:

If the video/broadcast object is in the unrealized state and ~~video from~~ exactly one channel is currently being presented by the OITF then this binds the video/broadcast object to that ~~video~~channel (even if the current channel does not contain video and/or audio). If more than one channel is currently being presented by the OITF then this binds the video/broadcast object to the channel whose audio is being presented. *A successful call shall result in control of the resources used to present the channel (tuner, video decoder if the channel includes video and audio decoder if the channel includes audio) being seamlessly transferred to the calling HbbTV application. This is intentionally the opposite of the "first-come, first-served" policy used between a video/broadcast object and other video/broadcast or A/V control objects.*

If the video/broadcast object is in the stopped state then this restarts presentation of ~~video and audio from~~ the current channel under the control of the video/broadcast object. ~~If video from more than one channel is currently being presented by the OITF then this binds the video/broadcast object to the channel whose audio is being presented.~~

If the video/broadcast object is in the unrealized state and there is no channel currently being presented, or binding to the necessary resources to play the channel (suitable tuner, suitable video decoder if the channel includes video and suitable audio decoder if the channel includes audio) through the video/broadcast object fails for whichever reason, the OITF SHALL dispatch an event to the onPlayStateChange listener(s) whereby the state parameter is given value 0 ("unrealized") and the error parameter is given the appropriate error code.

Clause A.2.4.5 is extended with the following text;

In clause 7.13.3, the definition of the property onProgrammesChanged is modified with the addition of the text shown underlined:

> The function that is called <u>for a video/broadcast object in the presenting or stopped states</u> when the programmes property has been updated with new programme information, e.g. when the current broadcast programme is finished and a new one has started. The specified function is called with no arguments.

4.13.7 Key "Label" should be "label"

In clause A.2.5.3, in the table of keys and values for the "value" attribute, the key "Label" shall be "label".

4.13.8 video/broadcast object issues when playing A/V from broadband

An extra row is added to the table in clause A.2.4.1 as shown:

Connecting	The terminal successfully connected to the broadcast or IP multicast stream but presentation of content is blocked, e.g. by a parental rating mechanism ~~or~~, content protection mechanism <u>or resources can't be claimed that are currently in use for presenting broadband content</u>	Connecting	ChannelChangeSucceeded PlayStateChange	This is conceptually equivalent to a successful channel change where a transient error immediately pre-empts media presentation without the video/broadcast object entering the presenting state.

The following text is added at the end of the bulleted list in clause A.2.4.1.

🎬 The following paragraph is amended as shown using underline/strike-through markup.

> If the <u>current</u> channel ~~currently being presented~~ is requested to be changed due to an action outside the application (for example, the user pressing the CH+ key on the remote) then any video/broadcast object ~~presenting~~ <u>bound to that</u> channel (i.e. <u>in the connecting, presenting or stopped states</u> as the result of a call to bindToCurrentChannel()) SHALL perform the same state transitions and dispatch the same events as if the channel change operation was initiated by the application using the setChannel() method.

The following text is added at the end of clause A.2.4.5.

🎬 In clause 7.13.7.1, the definition of the property currentChannel is changed as shown;

> The channel currently ~~being presented by~~<u>bound to</u> this embedded object <u>(i.e. the object is in the connecting, presenting or stopped states as the result of a call to bindToCurrentChannel())</u> if the user has given permission to share this information, possibly through a mechanism outside the scope of this specification. If no channel is ~~being presented~~ <u>bound to this embedded object</u>, or if this information is not visible to the caller, the value of

this property SHALL be null.

A new clause A.2.26 is added a shown.

A 2.26 Correction to the ApplicationPrivateData class

In clause 7.2.4.1, the definition of the property `currentChannel` is changed as shown;

For a broadcast-related application, the value of the property contains the channel whose AIT is currently controlling the lifecycle of this application. If ~~no channel is being presented, or if~~ the application is not broadcast-related, the value of this property shall be null. <u>During a channel change, the value of the property shall reflect the new channel once a</u> `ChannelChangeSucceeded` <u>event has been sent to any registered listeners on the corresponding video/broadcast object.</u>

<u>NOTE: If the terminal does not acquire the AIT signalling for the new channel until after the</u> `ChannelChangeSucceeded` <u>event has been generated then an application that is not allowed to survive the channel change will see the new value for a short time before it is stopped.</u>

4.13.9 Lack of linkage between spec version in AIT signalling and version in DOCTYPE in individual pages of app

In clause A.2.6.2, the following note is added immediately after the paragraph "Terminals are not required to load or run documents which do not include one of the DOCTYPE doctype declarations defined or referenced above."

> NOTE: There is no linkage between the DOCTYPE used in the pages that form part of an HbbTV application and the contents of the version fields in the AIT or XML AIT from which the application was launched. For example, an application signalled as requiring version 1.4.1 can include pages with any of the DOCTYPEs listed above.

Also all instances of "doctype" in lower case (either with or without quotation marks) are replaced by DOCTYPE in upper case.

4.13.10 Two setChannel methods of video/broadcast defined with same signature

In clause A.2.4.7.4, the method "void setChannel(Channel channel, Boolean trickplay, String contentAccessDescriptorURL, Integer offset)" and the preceeding paragraph are removed.

4.13.11 PVR API issues

In table A.1 in clause A.1, the notes column of the "Extensions to video/broadcast for recording and timeshift" row has "- onPlaySpeedsArrayChanged " added to the list under "Terminals that support time-shift of broadcast video shall support the following events and properties even if they do not support the full PVR option:".

The following changes are made in clause A.2.4.7.3:

1. The first instance of the property "ScheduledRecordingCollection getInProgressRecordings()" has the signature replaced by "function onPlayPositionChanged(Integer position)". The definition of the property remains unchanged.

2. The second instance of the property "ScheduledRecordingCollection getInProgressRecordings()" is deleted.

3. The type of the playSpeed property is changed from Integer to Number.

4. The property onplaySpeedsArrayChangedonPlaySpeedsArrayChanged() is renamed onPlaySpeedsArrayChangedonPlaySpeedsArrayChanged() (i.e. "play"→"Play").

5. The following text is added at the end of the clause - "In addition, the properties recordingState and onRecordingEvent defined in A.2.4.8.2 shall be supported."

In clause A.2.4.8.1, the paragraph immediately before the figure is extended with the underlined text.

> The recording functionality is subject to the state transitions represented in the state diagram in Figure A.1. The timeshift functionality is not represented explicitly in these state diagrams but is defined in the following clauses.

The following changes are made in clause A.2.4.8.2.

1. The definition of the "error" argument of the "onRecordingEvent" property is changed from "The current state of the recording. One of:" to "If the state of the recording has changed due to an error, this field contains an error code detailing the type of error. One of:".

2. In the table listing error codes for the error argument of the onRecordingEvent property, the rows for values 5 and 6 ("Value not used" and "Recording has successfully completed") are deleted.

4.13.12 Scope of AVAudio and AVVideoComponents

The following changes are made in clause A.2.4.6.

1) The following text is added at the end of the description of the createAVAudioComponent method.

> The scope of successfully created objects shall be limited to the application that created them. When that application exits, they shall cease to exist and shall no longer be presented. This may result in no audio being presented.

2) The following text is added at the end of the description of the createAVVideoComponent method.

> The scope of successfully created objects shall be limited to the application that created them. When that application exits, they shall cease to exist and shall no longer be presented. This may result in no video being presented.

4.13.13 Allow applications to prioritise DRM to use

NOTE: The text added below is further modified by clause 4.13.27.

A new clause A.2.27 is added as follows:

A.2.27 Extensions to the application/oipfDrmAgent embedded object

> This object shall be extended with the following additional method.

Boolean setActiveDRM(String DRMSystemID)

Description	Sets the DRM system, specified by `DRMSystemID`, that the terminal shall use with any new requests for playing protected broadband content. Any other DRM systems present in the terminal shall not be used with new requests until this method is called again even if this means playback of content fails.	
	If the method is called with the `DRMSystemID` set to null, the algorithm used by the terminal to determine which DRM to use is outside the scope of the present document. The value `true` shall always be returned in this case. This shall be the default state if no calls to this method have been made.	
	A call to this method with `DRMSystemID` set to `"urn:hbbtv:oipfdrm:inactive"` shall disable the use of all DRM systems in response to requests to play protected broadband content with the exception that the operation of the EME API is not affected (see clause B.3). Methods on the `oipfDrmAgent` object may still be called in this state, though depending on the DRM system, some uses of `sendDRMMessage` may fail. Protected broadband content may still play if suitable keys or licences are provided using the EME API.	
	If for any reason the terminal is unable to set the specified DRM system as requested, the method shall return `false`, otherwise it shall return `true`.	
Arguments	*DRMSystemID*	The DRM system as defined in clause 9.3.10 of the OIPF DAE specification [1] and in Table 9 ("DRMControlInformation Type Semantics") of the OIPF Metadata specification [18].

In table A.1 in clause A.1, the notes column of the "Content Service Protection API " row has the following text added at the end.

> If the DRM feature is supported (even if with only one DRM system) or if the terminal supports CI Plus then the extensions defined in clause A.2.27 shall be supported.

4.13.14 Unclear language in A 2.20.1 - Extensions to Represent Subtitle Presentation

In clause A.2.20.1, the description of the subtitles enabled property is changed from:

> Shall be set to false if subtitles are disabled by the terminal and applications cannot enable subtitles using the component selection API of the supported media objects i.e. A/V Control object, video/broadcast object and HTML5 media elements. Otherwise shall be set to true.

to:

> Shall be set to false if subtitles are disabled by the terminal. When set to false, subtitle components that are selected using a video/broadcast object, A/V control object or HTML5 media element will not be presented. See also clause 10.2.7.

4.13.15 Clarification to parental rating

In clause A.1, in the row for the ParentalRating class, in the Notes column;

> For example, "13" means a programme that is rated suitable for persons of 13 years of age or older.

is replaced with

> For further information, see clause A.2.28.

A new clause A.2.28 is added as follows:

A.2.28   Clarification of encoding of DVB-SI parental ratings

The DVB parental rating scheme is represented in a ParentalRating object by setting the scheme property to "dvb-si". The contents of the ParentalRating object are determined by the DVB parental_rating_descriptor, as defined in clause 8.4.4 of DAE[1]. The relationship between the rating field in the DVB parental_rating_descriptor and the ParentalRating object name and value properties is shown in Table A.6.

Table A.6: Encoding of parental rating

Value in DVB-SI rating field	ParentalRating name property	ParentalRating value property	Description
0x01	"4"	4	Recommended minimum age is 4 years old
0x02	"5"	5	Recommended minimum age is 5 years old
0x03	"6"	6	Recommended minimum age is 6 years old
0x04	"7"	7	Recommended minimum age is 7 years old
0x05	"8"	8	Recommended minimum age is 8 years old
0x06	"9"	9	Recommended minimum age is 9 years old
0x07	"10"	10	Recommended minimum age is 10 years old
0x08	"11"	11	Recommended minimum age is 11 years old
0x09	"12"	12	Recommended minimum age is 12 years old
0x0A	"13"	13	Recommended minimum age is 13 years old
0x0B	"14"	14	Recommended minimum age is 14 years old
0x0C	"15"	15	Recommended minimum age is 15 years old
0x0D	"16"	16	Recommended minimum age is 16 years old
0x0E	"17"	17	Recommended minimum age is 17 years old
0x0F	"18"	18	Recommended minimum age is 18 years old

4.13.16 Reference to NI clause 10.1.1 of the DAE specification

In table A.2, in the row for "trusted" security, in the "Description" column, the text in brackets is removed as shown in strike-through.

Only trusted applications as defined in clause 11.1 shall have access to the referenced API.

If other applications or web pages try to use this API, the terminal shall throw an error with the name property set to SecurityError ~~(see clause 10.1.1 of the OIPF DAE specification [1])~~.

Note that for embedded objects, untrusted applications may acquire instances of them without restrictions, either through the object factory or by using HTMLObjectElements. Security restrictions are enforced only when the application attempts to access properties or execute functions on the objects.

4.13.17 Incoherent AVSubtitleComponent language with TTML tracks

In clause A.1, in table A.1, the following changes are made in the "Notes" column:

1) In the row for "Extensions for playback of selected media components", clause 7.16.5 of the OIPF DAE specification, the following text is added at the end:

The value of the language property shall be either an ISO 639-1 [60] 2-character language code or an ISO 639-2 [61] 3-character language code as defined by clause 8.4.2 of the OIPF DAE specification [1] as modified in the present document.

2) In the row for "AVComponent", clause 8.4.2 of the OIPF DAE specification, the following text is added at the end:

> See clause A.2.5.3 of the present document for the mapping for EBU-TT-D subtitles.

4.13.18 Possible ambiguity in table A.1 re subtitlesEnabled

In clause A.1, in table A.1, in the row for "The Configuration class", the notes column is changed as shown using underline / strike-through notation.

> Support for read-only access to the following properties is mandatory:
>
> - preferredAudioLanguage
>
> - preferredSubtitleLanguage
>
> - preferredUILanguage
>
> - countryId
>
> The extensions to the Configuration class defined in clause A.2.20 shall be supported.
>
> All other properties and methods are ~~optional~~not included.
>
> ~~The extensions to the Configuration class defined in clause A.2.20 shall be supported.~~

4.13.19 Errors in mapping from DASH roles to HTML5 audio/video track kind values

In clause A.2.12.3, the following text is changed as shown by underline and strike-through markup.

For a `VideoTrack`, g~~G~~iven a role scheme of "`urn:mpeg:dash:role:2011`", determine the `kind` attribute from the value of the role descriptors in the `<AdaptationSet>` element as follows.

> - "alternative": if the role is "alternate" but not also "main" or "commentary", or "dub";
> - "captions": if the role is "caption" and also "main";
> - ~~"descriptions": if the role is "description" and also "supplementary";~~
> - "main": if the role is "main" but not also "caption", "subtitle", or "dub";
> - ~~"main-desc": if the role is "main" and also "description";~~
> - "sign": permitted for `VideoTrack`s by HTML5 [54] but not used in the present document;
> - "subtitles": if the role is "subtitle" and also "main";
> - ~~"translation": if the role is "dub" and also "main";~~
> - "commentary": if the role is "commentary" but not also "main";
> - "": otherwise.

For an `AudioTrack`, given a role scheme of "`urn:mpeg:dash:role:2011`", determine the `kind` attribute from the value of the role descriptors in the `<AdaptationSet>` element as follows.

> - "alternative": if the role is "alternate" but not also "main" or "commentary", or "dub";
> - "descriptions": if the role is "description" and also "supplementary";

- o "main": if the role is "main" but not also "caption", "subtitle", "dub" or "description";

- o "main-desc": if the role is "main" and also "description";

- o "translation": if the role is "dub" and also "main";

- o "commentary": if the role is "commentary" but not also "main";

- o "": otherwise.

4.13.20 deviceID

Clause A.2.20.5 is modified as shown using underline / strike-through markup.

A.2.20.5 Extensions for distinctive identifiersdevice IDs

The following property is added to the `Configuration` class.

readonly String deviceId
NOTE 1: This property is named deviceId for historical reasons but it does not return a permanent identifier for the device. Returns either a string representing an distinctive identifier that is unique for the combination of the receiver terminal and the HTML document origin or a status code. The distinctive identifier shall use a character set that is restricted to alphanumeric characters and the hyphen. The status code shall be a number preceded by the '#' character. Valid status codes are:<table><tr><td>**Status code**</td><td>**Description**</td></tr><tr><td>#1</td><td>The terminal is configured to request explicit user approval for this application. The application may call `requestAccessToDistinctiveIdentifier` to obtain such approval even if this method has previously been called and the user did not grant access.</td></tr><tr><td>#2</td><td>Access to the distinctive identifier is denied explicitly by the user following a previous call by the application to `requestAccessToDistinctiveIdentifier`. Further calls to `requestAccessToDistinctiveIdentifier` willdo not result in the user being asked again for approval. This is for use by terminals that restrict the number of times that an application may call this method.</td></tr><tr><td>#3</td><td>Access to the distinctive identifier is denied in accordance with the user option setting – see clause 12.1.5 2).</td></tr><tr><td>#4</td><td>Access to the distinctive identifier is denied by the terminal manufacturer, e.g. because the application provider and the terminal manufacturer have not entered into a suitable agreement covering such access.</td></tr><tr><td>#5</td><td>Access to the distinctive identifier is denied as the application has not yet been activated.</td></tr></table>NOTE 2:   Other status codes may be defined in future versions of the present document. The value of this property may change after a call to `requestAccessToDistinctiveIdentifier`, a change to the user option, a request by the user to generate a new distinctive identifier or some other event. If this identifier is not being made available to the application (see clause 12.1.5 of the present document), then the value of this property shall be the empty string ("").

The following method is added to the `Configuration` class.

requestAccessToDistinctiveIdentifier(function callback)

Description	Calls the callback with `true` as the first argument if the `deviceId` property contains a distinctive identifier, otherwise calls the callback with `false` as the first argument. This callback takes place either immediately or after a user interaction according to the following rules.
	Calls to this method while a callback is outstanding shall be ignored.
	If this method is supported, the terminal shall provide some native UI that requests the user to grant access to the distinctive identifier for the calling application. The terminal may persistently store the user response between invocations of the application.
	If the `deviceId` property contains the value "#1", the terminal shall display this native UI when this method is called. The callback shall be called only after the UI is removed and the argument shall reflect the updated state of the `deviceId` property following the interaction with the user. This method call shall not block while the UI is displayed.
	If the `deviceId` property contains a different status code, the terminal shall not display the native UI and shall immediately call the callback with `false` as the first argument.
	If the `deviceId` property already contains a distinctive identifier, the terminal shall not display the native UI and shall immediately call the callback with `true` as the first argument.

4.13.21 Warning to app developers that some values/properties are insecure

A new clause A.2.29 is added as follows;

A.2.29   Security

A.2.29.1   Risk of tampering with data returned by APIs

Application developers should be aware that some APIs return data that may not be authenticated. In some circumstances an attacker may be able to modify the broadcast signalling from which this data is derived. This particularly applies to the properties and methods of the Channel and Programme classes.

○ Applications should be written to be tolerant of values which are outside the expected range without hanging up, locking up or crashing.

○ Applications should treat the values returned by Channel.name, Programme.name, Programme.description and Programme.longDescription with caution as an attacker may modify the broadcast signalling to include HTML or JavaScript as well as values that are outside the expected set. Applications shall not use the data returned by these properties in a way that would result in them being executed by the browser.

○ Applications should treat data returned by the Programme.getSIDescriptors method with caution. Applications shall not use this data in a way that would result in that such data being executed by the browser. Applications should be written to be tolerant of values which are outside the expected range without hanging up, locking up or crashing.

The following text is added to table A.1 in the "Notes" column of the rows "The Programme Class"/"Basics", "The Programme Class"/"DVB-SI extensions to Programme" and "Channel class".

Application developers need to be aware of clause A.2.29.1 concerning security.

4.13.22 Security level for videobroadcast currentChannel property : remove Broadcast-related

In clause A.1, table A.1, in the row for "Extensions to video/broadcast for current channel information", the contents of the "Security" column are removed.

4.13.23 Invalid reference

In clause A.2.20.5, in the description of the deviceId property, in the description of status code "#3", the reference to "12.1.5 2" is replaced with a reference to "12.1.5".

4.13.24 HBBTV and W3C/DASH-IF text track kind calculation inconsistency

In clause A.2.12.3, 'OR (role is "caption");' and ' OR (role is "subtitle");' are added as shown underlined.

The definition of the value of the kind property of a TextTrack in the MPEG DASH system layer shall be replaced with the following:

• "captions": if (role is "main" AND the MPD contains an audio Adaptation Set or Preselection with role "main" and the same language as the subtitle track AND an accessibility descriptor is present with the schemeIdUri set to "urn:tva:metadata:cs:AudioPurposeCS:2007" and a value 2 [for the hard of hearing]) OR (role is "commentary") OR (role is "caption");

• "subtitles": if (role is "alternate") OR (role is "main" AND no accessibility scheme is specified) OR (role is "subtitle");

• "metadata": otherwise.

4.13.25 Additional web security APIs

A new clause A.3.14 is added as follows.

A.3.14 Web security improvements

Terminals may implement web security improvements, for example Cross-Origin Read Blocking (CORB) as defined in clause 3.5 of Fetch [i.24].

Applications need to be aware of these improvements and should be prepared for behaviour that fails on desktop browsers to also fail on implementations of the present document even if not explicitly required.

4.13.26 Clarification required on expected state when video/broadcast object stops

The underlined text is added in clause A.2.4.1 as shown.

• A video/broadcast object with a CSS rule of display:none shall not be loaded and hence shall not be decoding audio or video.

• Setting CSS display:none on a video/broadcast object should be equivalent to calling the release method or the setChannel method with a parameter of null. Applications should not depend on the state of the video/broadcast object if the application sets CSS

<u>display:none. Setting CSS visibility to hidden does not cause a state change.</u>

•   In Table 8, "State transitions for the video/broadcast embedded object", the following rows are modified as shown underlined.

4.13.27 setActiveDRM : need to specify how/when the default behavior is restored

For TS 102 796 V1.4.1, the text in clause 4.13.13 above is modified as described. For TS 102 796 V1.5.1, clause A.2.27 is modified as described.

The underlined text is added as shown.

Sets the DRM system, specified by DRMSystemID, that the terminal shall use with any new requests for playing protected broadband content. Any other DRM systems present in the terminal shall not be used with new requests until this method is called again <u>or the application stops for any reason,</u> even if this means playback of content fails.

4.13.28 Behaviour of html5 media when HbbTV app is hidden

The following text is added as the end of clause A.2.14.

The requirements in the following paragraph shall also apply to audio being presented by an HTML5 media element.

Calling the Application.hide() method SHALL cause video (and any subtitles) being presented under the control of that application to be hidden, and any audio being presented by the video/broadcast or A/V Control object under the control of that application to be muted. Calling Application.show() SHALL cause video and audio presentation to be restored.

4.13.29 OSD display when returning to broadcast from broadband

The following text is added at the end of clause A.2.4.5.

Terminals shall not show any UI (e.g. an indication of video or audio properties or attributes such as "16:9") when a video broadcast object successfully changes either from the Stopped state to the Connecting state or from the Connecting state to the Presenting state when the previous state change was from Stopped to Connecting.

4.13.30 Firing playStateChange when the state does not change

The following text is added at the end of clause A.2.4.1.

•   In clause 7.3.1.2 of the OIPF DAE specification [1], the definition of the onPlayStateChange property is modified as shown using underline/strike-through markup:

The function that is called when the play state of the video/broadcast object changes <u>as defined in Table 8, including changes from a state back to the same state</u>. This function may be called either in response to an action initiated by the application, an action initiated by the OITF or an error (see section 7.13.1.1). <u>In some cases, Table 8 defines that ChannelChangeError will be called instead of this function.</u>

4.13.31 selectComponent/unselectComponent when v/b playstate is other than PRESENTING

The following text is added at the end of clause A.2.4.5.

> In clause 7.16.5.1.3, if either signature of the selectComponent method is called when the set of components are not known (where "known" has the meaning defined for getComponent and getActiveComponents) then the method call shall fail with an InvalidStateError DOMException.

4.13.32 bindToCurrentChannel from STOPPED state when resources are not available

In clause A.2.4.1, an additional row is added to the modifications to table 8 of the OIPF DAE specification and an additional paragraph is added below as shown underlined.

- In table 8, "State transitions for the video/broadcast embedded object", the following rows are modified as shown underlined;

Old State	Trigger	New State	State Transition Events	Description
Stopped	bindToCurrentChannel()	Connecting	PlayStateChange	Video and audio presentation is enabled The terminal starts to present the current channel.
Stopped	bindToCurrentChannel when suitable video and audio decoders are not available	Stopped	PlayStateChange	
Connecting	The terminal successfully connected to the broadcast or IP multicast stream but presentation of content is blocked, e.g. by a parental rating mechanism ~~or~~, content protection mechanism or resources can't be claimed that are currently in use for presenting broadband content	Connecting	ChannelChangeSucceeded PlayStateChange	This is conceptually equivalent to a successful channel change where a transient error immediately pre-empts media presentation without the video/broadcast object entering the presenting state.

- In clause 7.13.1.3 of the OIPF DAE specification [Error: Reference source not found], the definition of the `bindToCurrentChannel()` method is modified as shown:

> If the video/broadcast object is in the unrealized state and video from exactly one channel is currently being presented by the OITF then this binds the video/broadcast object to that videochannel (even if the current channel does not contain video and/or audio). If more than one channel is currently being presented by the OITF then this binds the video/broadcast object to the channel whose audio is being presented. A successful call shall result in control of the resources used to present the channel (tuner, video decoder if the channel includes video and audio decoder if the channel includes audio) being seamlessly transferred to the calling HbbTV application. This is intentionally the opposite of the "first-come, first-served" policy used between a video/broadcast object and other video/broadcast or A/V control objects.

If the video/broadcast object is in the stopped state and suitable video/audio decoders are available then this restarts presentation of video and audio from the current channel under the control of the video/broadcast object. If video from more than one channel is currently being presented by the OITF then this binds the video/broadcast object to the channel whose audio is being presented.

When bindToCurrentChannel is called on a video/broadcast object in the stopped state and no suitable media decoders are available then the method call shall fail. For example when the media decoders are used by an A/V control object or by an HTML5 video element that is playing. A playStateChange event shall be generated with error '11'- "insufficient resources are available to present the given channel (e.g. a lack of available codec resources)". The video/broadcast object shall stay in the stopped state. The current channel of the video/broadcast object, the current channel of the terminal and the current channel of the application shall all remain unchanged (see OIPF [] clause H.4). If the media decoders would become available and bindToCurrentChannel is called again then the video/broadcast object shall behave as specified.

If the video/broadcast object is in the unrealized state and there is no channel currently being presented, or binding to the necessary resources to play the channel (suitable tuner, suitable video decoder if the channel includes video and suitable audio decoder if the channel includes audio) through the video/broadcast object fails for whichever reason, the OITF SHALL dispatch an event to the onPlayStateChange listener(s) whereby the state parameter is given value 0 ("unrealized") and the error parameter is given the appropriate error code.

4.13.33 AudioTrack and VideoTrack creation with EssentialProperty descriptors

In clause A.2.12.1, the text shown underlined is added.

The following shall apply when an HTML5 media element is presenting content whose system format is MPEG DASH:

• A VideoTrack object shall be created for each video Adaptation Set in the MPD for which @profiles includes or is inferred to include a profile supported by the terminal. VideoTrack objects shall not be created for any Adaptation Set that players are required to ignore according to clause 10.17 of TS 103 285 []. The order of VideoTrack objects in the VideoTrackList shall be the same as the order that the corresponding Adaptation Sets are in the MPD.

• An AudioTrack object shall be created for each Preselection element in the MPD for which @profiles includes or is inferred to include a profile supported by the terminal. An AudioTrack object shall be created for each audio Adaptation Set in the MPD for which @profiles includes or is inferred to include a profile supported by the terminal and whose id is not used in any Preselection/@preselectionComponents attribute within the same Period. AudioTrack objects shall not be created for any Adaptation Set that players are required to ignore due to an EssentialProperty descriptor according to clause 10.17 of TS 103 285 []. AudioTrack objects may be created for Adaptation Sets with @codecs (including information such as profile and level, object type etc.) that the player does not understand or cannot process. The order of AudioTrack objects in the AudioTrackList shall be the same as the order that the corresponding Adaptation Sets and Preselection elements have in the MPD.

4.13.34 XML Schema

4.13.34.1 HbbTV 2.0.1

Clause A.2.15 is replaced with the following;

A.2.15 Extensions to the OIPF-defined capability negotiation mechanism

The following schema is used instead of the schema defined by annex F of the OIPF DAE specification [1]:

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:termcap="urn:hbbtv:config:oitf:oitfCapabilities:2017-1"

        xmlns:xs="http://www.w3.org/2001/XMLSchema"

        xmlns:vc="http://www.w3.org/2007/XMLSchema-versioning"

        targetNamespace="urn:hbbtv:config:oitf:oitfCapabilities:2017-1"

        elementFormDefault="qualified"

        attributeFormDefault="unqualified"

        vc:minVersion="1.1"

        version="2016.1">


    <xs:annotation>

        <xs:documentation>This is the schema for the HbbTV XML Capabilities.</xs:documentation>

    </xs:annotation>


    <!-- ***************************************************************** -->

    <!-- Start CEA-with-OIPF-and-HbbTV-mods -->


    <xs:redefine schemaLocation="ce-html-profiles-1-0.xsd">


        <xs:complexType name="profileListType">

            <xs:complexContent>

                <xs:extension base="termcap:profileListType">

                    <xs:sequence>

                        <!-- Introduced in HbbTV 2.0.1 -->

                        <!-- NOTE: This element was defined in the original HbbTV 2.0.1

                                schema as going in the <ext> tag, not here.  However, the

                                original HbbTV 2.0.1 specification text included an
```

```
                        example that put it here.  Hence this fixed schema allows

                        it in either place.  For best compatibility, applications

                        should look for it in both places.

                        -->

                    <xs:element name="html5_media" type="xs:boolean" minOccurs="0"/>


                    <!-- NOTE: This element was defined in the original OIPF/HbbTV

                        schema as going in the <ext> tag, not here.  However, the

                        original HbbTV 2.0.1 specification text said to put here.

                        Hence this fixed schema allows it in either place.

                        For best compatibility, applications should look for it in

                        both places.

                        -->

                    <xs:element name="drm" type="termcap:drmType" minOccurs="0"
maxOccurs="unbounded"/>


                    <!-- For future compatibility -->

                    <xs:any namespace="##targetNamespace" processContents="lax"/>

                </xs:sequence>


                <!-- For future compatibility -->

                <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

            </xs:extension>

        </xs:complexContent>

    </xs:complexType>



    <xs:complexType name="uiProfileType">

        <xs:complexContent>

            <xs:extension base="termcap:uiProfileType">

                <!-- For future compatibility -->

                <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

            </xs:extension>

        </xs:complexContent>

    </xs:complexType>
```

```xml
<xs:complexType name="uiExtensionType">

    <xs:complexContent>

        <xs:extension base="termcap:uiExtensionType">

            <xs:choice minOccurs="0" maxOccurs="unbounded">

                <!-- If adding new elements to this list, please note:


                        * Do NOT add minOccurs/maxOccurs to any entry

                          here.  The parent <choice> element has

                          minOccurs="0" maxOccurs="unbounded" and they

                          will apply to all elements listed here.

                          Specifying a minOccurs/maxOccurs with those

                          values here will just confuse XML parsers.

                          Specifying different values will have no effect

                          except to confuse humans.

                -->


                <!-- Valid values for HbbTV 1.5 / OIPF 1.3 -->

                <xs:element name="clientMetadata" type="termcap:metadataType"/>

                <xs:element name="communicationServices" type="xs:boolean"/>

                <xs:element name="configurationChanges" type="xs:boolean"/>

                <xs:element name="drm" type="termcap:drmType"/>

                <xs:element name="extendedAVControl" type="xs:boolean"/>

                <xs:element name="mdtf" type="xs:boolean"/>

                <xs:element name="overlayIPbroadcast" type="termcap:overlayType"/>

                <xs:element name="overlaylocaltuner" type="termcap:overlayType"/>

                <xs:element name="parentalcontrol" type="termcap:parentalControlType"/>

                <xs:element name="pollingNotifications" type="xs:boolean"/>

                <xs:element name="presenceMessaging" type="xs:boolean"/>

                <xs:element name="recording" type="termcap:pvrType"/>

                <xs:element name="remote_diagnostics" type="xs:boolean"/>

                <xs:element name="video_broadcast" type="termcap:videoBroadcastType"/>


                <!-- HbbTV 2.0.0 was withdrawn, so elements are listed

                        under HbbTV 2.0.1 instead -->


                <!-- Introduced in OIPF 2.3 (part of HbbTV 2.0.1) -->

                <xs:element name="hibernateMode" type="xs:boolean"/>
```

```xml
                        <xs:element name="html5_media" type="xs:boolean"/>

                        <xs:element name="playbackControl" type="termcap:playbackType"/>

                        <xs:element name="remoteControlFunction" type="xs:boolean"/>

                        <xs:element name="telephony_services" type="termcap:telephonyServicesType"/>

                        <xs:element name="temporalClipping" type="termcap:hasCapability"/>

                        <xs:element name="wakeupApplication" type="xs:boolean"/>

                        <xs:element name="wakeupOITF" type="xs:boolean"/>

                        <xs:element name="widgets" type="xs:boolean"/>


                        <!-- Introduced in HbbTV 2.0.1 -->
                        <xs:element name="graphicsPerformance"
type="termcap:graphicsPerformanceType"/>


                        <!-- For future compatibility and 3rd party extensions -->
                        <xs:any namespace="##any" processContents="lax"/>
                    </xs:choice>

                </xs:extension>

            </xs:complexContent>

        </xs:complexType>



        <xs:complexType name="fontFormatType">

            <xs:simpleContent>

                <xs:extension base="termcap:fontFormatType">

                    <!-- For future compatibility -->

                    <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

                </xs:extension>

            </xs:simpleContent>

        </xs:complexType>



        <xs:complexType name="securityType">

            <xs:simpleContent>

                <xs:extension base="termcap:securityType">

                    <!-- For future compatibility -->

                    <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

                </xs:extension>
```

```
                </xs:simpleContent>

            </xs:complexType>




        <xs:complexType name="downloadType">

            <xs:simpleContent>

                <xs:extension base="termcap:downloadType">

                    <xs:attribute name="manageDownloads" type="termcap:manageDownloadsType"
default="none"/>


                    <!-- For future compatibility -->

                    <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

                </xs:extension>

            </xs:simpleContent>

        </xs:complexType>




        <xs:complexType name="mimeExtensionType">

            <xs:simpleContent>

                <xs:extension base="termcap:mimeExtensionType">

                    <!-- For future compatibility -->

                    <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

                </xs:extension>

            </xs:simpleContent>

        </xs:complexType>




        <xs:complexType name="audioProfileType">

            <xs:complexContent>

                <xs:extension base="termcap:audioProfileType">

                    <xs:attribute name="DRMSystemID" type="xs:string"/>


                    <!-- Introduced in HbbTV 2.0.1 -->

                    <xs:attribute name="sync_tl" type="termcap:sync_tl_type" use="optional"/>


                    <!-- For future compatibility -->

                    <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
```

```xml
            </xs:extension>

        </xs:complexContent>

    </xs:complexType>




    <xs:complexType name="videoProfileType">

        <xs:complexContent>

            <xs:extension base="termcap:videoProfileType">

                <xs:attribute name="DRMSystemID" type="xs:string"/>


                <!-- Introduced in HbbTV 2.0.1 -->

                <xs:attribute name="sync_tl" type="termcap:sync_tl_type" use="optional"/>

                <xs:attribute name="hdr" type="xs:anyURI" use="optional"/>


                <!-- For future compatibility -->

                <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

            </xs:extension>

        </xs:complexContent>

    </xs:complexType>



</xs:redefine>



<!-- End CEA-with-OIPF-and-HbbTV-mods -->
<!-- *************************************************************** -->



<!-- *************************************************************** -->
<!-- Start OIPF-with-HbbTV-mods -->


<!-- Note: We don't reference the OIPF schema, we just copy-paste it

     in here.  This avoids having multiple namespaces.

     -->


<!--ADDED: type definitions for the new elements defined in Section 9.3 of the
```

```
    Open IPTV forum Volume 5 Declarative Application Environment Release 2 specification
    -->
<xs:simpleType name="manageDownloadsType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="none"/>
        <xs:enumeration value="initiator"/>
        <xs:enumeration value="samedomain"/>
        <xs:enumeration value="all"/>
    </xs:restriction>
</xs:simpleType>



<xs:simpleType name="manageRecordingsType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="none"/>
        <xs:enumeration value="initiator"/>
        <xs:enumeration value="samedomain"/>
        <xs:enumeration value="all"/>
    </xs:restriction>
</xs:simpleType>



<xs:complexType name="videoBroadcastType">
    <xs:attribute name="type" type="xs:string" use="required"/>
    <xs:attribute name="transport" type="xs:string"/>
    <xs:attribute name="nrstreams" type="xs:unsignedInt" default="1"/>
    <xs:attribute name="scaling" type="termcap:scalingType" default="arbitrary"/>
    <xs:attribute name="minSize" type="xs:unsignedInt" default="0"/>
    <xs:attribute name="postList" type="xs:boolean" default="false"/>


    <!-- Introduced in OIPF 2.3 (part of HbbTV 2.0.1) -->
    <xs:attribute name="networkTimeshift" type="xs:boolean" default="false"/>
    <xs:attribute name="localTimeshift" type="xs:boolean" default="false"/>


    <!-- For future compatibility -->
    <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
</xs:complexType>
```

```
<xs:complexType name="pvrType">
    <xs:simpleContent>
        <xs:extension base="xs:boolean">
            <xs:attribute name="ipBroadcast" type="xs:boolean" default="false"/>
            <xs:attribute name="manageRecordings" type="termcap:manageRecordingsType"
default="none"/>
            <xs:attribute name="postList" type="xs:boolean" default="false"/>


            <!-- Introduced in OIPF 2.3 (part of HbbTV 2.0.1) -->
            <xs:attribute name="HAS" type="xs:boolean" default="false"/>
            <xs:attribute name="DASH" type="xs:boolean" default="false"/>


            <!-- For future compatibility -->
            <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>



<xs:complexType name="parentalControlType">
    <xs:simpleContent>
        <xs:extension base="xs:boolean">
            <xs:attribute name="schemes" type="xs:string"/>


            <!-- For future compatibility -->
            <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>



<xs:complexType name="metadataType">
    <xs:simpleContent>
        <xs:extension base="xs:boolean">
            <xs:attribute name="type" type="xs:string"/>
```

```xml
            <!-- For future compatibility -->

            <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

        </xs:extension>

    </xs:simpleContent>

</xs:complexType>



<xs:complexType name="drmType">

    <xs:simpleContent>

        <xs:extension base="xs:string">

            <xs:attribute name="DRMSystemID" type="xs:string" use="required"/>

            <xs:attribute name="protectionGateways" type="xs:string" default=""/>


            <!-- For future compatibility -->

            <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

        </xs:extension>

    </xs:simpleContent>

</xs:complexType>



<!-- Introduced in OIPF 2.3 (part of HbbTV 2.0.1) -->

<xs:complexType name="telephonyServicesType">

    <xs:simpleContent>

        <xs:extension base="xs:boolean">

            <xs:attribute name="video" type="xs:boolean" default="false"/>


            <!-- For future compatibility -->

            <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

        </xs:extension>

    </xs:simpleContent>

</xs:complexType>



<!-- Introduced in OIPF 2.3 (part of HbbTV 2.0.1) -->

<xs:complexType name="playbackType">
```

```xml
        <xs:simpleContent>

            <xs:extension base="xs:boolean">

                <xs:attribute name="type" type="xs:string"/>


                <!-- For future compatibility -->

                <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

            </xs:extension>

        </xs:simpleContent>

</xs:complexType>



<!-- Introduced in OIPF 2.3 (part of HbbTV 2.0.1) -->

<xs:complexType name="hasCapability">

    <!-- For future compatibility -->

    <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

</xs:complexType>



<!-- End OIPF-with-HbbTV-mods -->

<!-- ***************************************************************** -->



<!-- ***************************************************************** -->

<!-- Start HbbTV-defined -->



<!-- Introduced in HbbTV 2.0.1 -->

<xs:simpleType name="sync_tl_type">

    <xs:list itemType="termcap:sync_tl_values_type"/>

</xs:simpleType>



<!-- Introduced in HbbTV 2.0.1 -->

<xs:simpleType name="sync_tl_values_type">

    <xs:restriction base="xs:string">

        <xs:enumeration value="pts"/>

        <xs:enumeration value="ct"/>
```

```
            <xs:enumeration value="temi"/>

            <xs:enumeration value="dash_pr"/>

        </xs:restriction>

    </xs:simpleType>



    <!-- Introduced in HbbTV 2.0.1 -->

    <xs:complexType name="graphicsPerformanceType">

        <xs:attribute name="level" type="xs:string"/>


        <!-- For future compatibility -->

        <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

    </xs:complexType>



    <!-- End HbbTV-defined -->

    <!-- ***************************************************************** -->



</xs:schema>
```

4.13.34.2HbbTV 2.0.2

Clause A.2.15 is replaced with the following;

A.2.15 Extensions to the OIPF-defined capability negotiation mechanism

The following schema is used instead of the schema defined by annex F of the OIPF DAE specification [1]:

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:termcap="urn:hbbtv:config:oitf:oitfCapabilities:2017-1"

        xmlns:xs="http://www.w3.org/2001/XMLSchema"

        xmlns:vc="http://www.w3.org/2007/XMLSchema-versioning"

        targetNamespace="urn:hbbtv:config:oitf:oitfCapabilities:2017-1"

        elementFormDefault="qualified"

        attributeFormDefault="unqualified"

        vc:minVersion="1.1"

        version="2018.1">
```

```xml
<xs:annotation>

    <xs:documentation>This is the schema for the HbbTV XML Capabilities.</xs:documentation>

</xs:annotation>


<!-- ******************************************************************** -->
<!-- Start CEA-with-OIPF-and-HbbTV-mods -->



<xs:redefine schemaLocation="ce-html-profiles-1-0.xsd">



    <xs:complexType name="profileListType">

        <xs:complexContent>

            <xs:extension base="termcap:profileListType">

                <xs:sequence>

                    <!-- Introduced in HbbTV 2.0.1 -->

                    <!-- NOTE: This element was defined in the original HbbTV 2.0.1

                         schema as going in the <ext> tag, not here.  However, the

                         original HbbTV 2.0.1 specification text included an

                         example that put it here.  Hence this fixed schema allows

                         it in either place.  For best compatibility, applications

                         should look for it in both places.

                         -->

                    <xs:element name="html5_media" type="xs:boolean" minOccurs="0"/>


                    <!-- NOTE: This element was defined in the original OIPF/HbbTV

                         schema as going in the <ext> tag, not here.  However, the

                         original HbbTV 2.0.1 specification text said to put here.

                         Hence this fixed schema allows it in either place.

                         For best compatibility, applications should look for it in

                         both places.

                         -->

                    <xs:element name="drm" type="termcap:drmType" minOccurs="0"

maxOccurs="unbounded"/>


                    <!-- For future compatibility -->
```

```xml
                    <xs:any namespace="##targetNamespace" processContents="lax"/>
                </xs:sequence>


                <!-- For future compatibility -->
                <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>



    <xs:complexType name="uiProfileType">
        <xs:complexContent>
            <xs:extension base="termcap:uiProfileType">
                <!-- For future compatibility -->
                <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>



    <xs:complexType name="uiExtensionType">
        <xs:complexContent>
            <xs:extension base="termcap:uiExtensionType">
                <xs:choice minOccurs="0" maxOccurs="unbounded">
                    <!-- If adding new elements to this list, please note:


                            * Do NOT add minOccurs/maxOccurs to any entry
                              here.  The parent <choice> element has
                              minOccurs="0" maxOccurs="unbounded" and they
                              will apply to all elements listed here.
                              Specifying a minOccurs/maxOccurs with those
                              values here will just confuse XML parsers.
                              Specifying different values will have no effect
                              except to confuse humans.
                    -->


                    <!-- Valid values for HbbTV 1.5 / OIPF 1.3 -->
```

```xml
<xs:element name="clientMetadata" type="termcap:metadataType"/>

<xs:element name="communicationServices" type="xs:boolean"/>

<xs:element name="configurationChanges" type="xs:boolean"/>

<xs:element name="drm" type="termcap:drmType"/>

<xs:element name="extendedAVControl" type="xs:boolean"/>

<xs:element name="mdtf" type="xs:boolean"/>

<xs:element name="overlayIPbroadcast" type="termcap:overlayType"/>

<xs:element name="overlaylocaltuner" type="termcap:overlayType"/>

<xs:element name="parentalcontrol" type="termcap:parentalControlType"/>

<xs:element name="pollingNotifications" type="xs:boolean"/>

<xs:element name="presenceMessaging" type="xs:boolean"/>

<xs:element name="recording" type="termcap:pvrType"/>

<xs:element name="remote_diagnostics" type="xs:boolean"/>

<xs:element name="video_broadcast" type="termcap:videoBroadcastType"/>


<!-- HbbTV 2.0.0 was withdrawn, so elements are listed
        under HbbTV 2.0.1 instead -->


<!-- Introduced in OIPF 2.3 (part of HbbTV 2.0.1) -->
<xs:element name="hibernateMode" type="xs:boolean"/>

<xs:element name="html5_media" type="xs:boolean"/>

<xs:element name="playbackControl" type="termcap:playbackType"/>

<xs:element name="remoteControlFunction" type="xs:boolean"/>

<xs:element name="telephony_services" type="termcap:telephonyServicesType"/>

<xs:element name="temporalClipping" type="termcap:hasCapability"/>

<xs:element name="wakeupApplication" type="xs:boolean"/>

<xs:element name="wakeupOITF" type="xs:boolean"/>

<xs:element name="widgets" type="xs:boolean"/>


<!-- Introduced in HbbTV 2.0.1 -->
<xs:element name="graphicsPerformance"
type="termcap:graphicsPerformanceType"/>


<!-- Introduced in HbbTV 2.0.2 -->
<xs:element name="broadcast" type="xs:anyURI"/>

<xs:element name="video_display_format"
type="termcap:videoDisplayFormatType"/>
```

```xml
                    <!-- For future compatibility and 3rd party extensions -->
                    <xs:any namespace="##any" processContents="lax"/>
                </xs:choice>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>



    <xs:complexType name="fontFormatType">
        <xs:simpleContent>
            <xs:extension base="termcap:fontFormatType">
                <!-- For future compatibility -->
                <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>



    <xs:complexType name="securityType">
        <xs:simpleContent>
            <xs:extension base="termcap:securityType">
                <!-- For future compatibility -->
                <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>



    <xs:complexType name="downloadType">
        <xs:simpleContent>
            <xs:extension base="termcap:downloadType">
                <xs:attribute name="manageDownloads" type="termcap:manageDownloadsType"
default="none"/>


                <!-- For future compatibility -->
                <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
```

```xml
            </xs:extension>

        </xs:simpleContent>

    </xs:complexType>




    <xs:complexType name="mimeExtensionType">

        <xs:simpleContent>

            <xs:extension base="termcap:mimeExtensionType">

                <!-- For future compatibility -->

                <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

            </xs:extension>

        </xs:simpleContent>

    </xs:complexType>




    <xs:complexType name="audioProfileType">

        <xs:complexContent>

            <xs:extension base="termcap:audioProfileType">

                <xs:attribute name="DRMSystemID" type="xs:string"/>


                <!-- Introduced in HbbTV 2.0.1 -->

                <xs:attribute name="sync_tl" type="termcap:sync_tl_type" use="optional"/>


                <!-- For future compatibility -->

                <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

            </xs:extension>

        </xs:complexContent>

    </xs:complexType>




    <xs:complexType name="videoProfileType">

        <xs:complexContent>

            <xs:extension base="termcap:videoProfileType">

                <xs:attribute name="DRMSystemID" type="xs:string"/>


                <!-- Introduced in HbbTV 2.0.1 -->

                <xs:attribute name="sync_tl" type="termcap:sync_tl_type" use="optional"/>
```

```xml
                <xs:attribute name="hdr" type="xs:anyURI" use="optional"/>


                <!-- For future compatibility -->
                <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>



</xs:redefine>



<!-- End CEA-with-OIPF-and-HbbTV-mods -->
<!-- **************************************************************** -->



<!-- **************************************************************** -->
<!-- Start OIPF-with-HbbTV-mods -->


<!-- Note: We don't reference the OIPF schema, we just copy-paste it
     in here.  This avoids having multiple namespaces.
     -->


<!--ADDED: type definitions for the new elements defined in Section 9.3 of the
    Open IPTV forum Volume 5 Declarative Application Environment Release 2 specification
    -->
<xs:simpleType name="manageDownloadsType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="none"/>
        <xs:enumeration value="initiator"/>
        <xs:enumeration value="samedomain"/>
        <xs:enumeration value="all"/>
    </xs:restriction>
</xs:simpleType>
```

```xml
<xs:simpleType name="manageRecordingsType">

    <xs:restriction base="xs:string">

        <xs:enumeration value="none"/>

        <xs:enumeration value="initiator"/>

        <xs:enumeration value="samedomain"/>

        <xs:enumeration value="all"/>

    </xs:restriction>

</xs:simpleType>



<xs:complexType name="videoBroadcastType">

    <xs:attribute name="type" type="xs:string" use="required"/>

    <xs:attribute name="transport" type="xs:string"/>

    <xs:attribute name="nrstreams" type="xs:unsignedInt" default="1"/>

    <xs:attribute name="scaling" type="termcap:scalingType" default="arbitrary"/>

    <xs:attribute name="minSize" type="xs:unsignedInt" default="0"/>

    <xs:attribute name="postList" type="xs:boolean" default="false"/>


    <!-- Introduced in OIPF 2.3 (part of HbbTV 2.0.1) -->

    <xs:attribute name="networkTimeshift" type="xs:boolean" default="false"/>

    <xs:attribute name="localTimeshift" type="xs:boolean" default="false"/>


    <!-- For future compatibility -->

    <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

</xs:complexType>



<xs:complexType name="pvrType">

    <xs:simpleContent>

        <xs:extension base="xs:boolean">

            <xs:attribute name="ipBroadcast" type="xs:boolean" default="false"/>

            <xs:attribute name="manageRecordings" type="termcap:manageRecordingsType"
default="none"/>

            <xs:attribute name="postList" type="xs:boolean" default="false"/>


            <!-- Introduced in OIPF 2.3 (part of HbbTV 2.0.1) -->

            <xs:attribute name="HAS" type="xs:boolean" default="false"/>
```

```xml
                <xs:attribute name="DASH" type="xs:boolean" default="false"/>


                <!-- For future compatibility -->
                <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
            </xs:extension>
        </xs:simpleContent>
</xs:complexType>



<xs:complexType name="parentalControlType">
    <xs:simpleContent>
        <xs:extension base="xs:boolean">
            <xs:attribute name="schemes" type="xs:string"/>


            <!-- For future compatibility -->
            <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>



<xs:complexType name="metadataType">
    <xs:simpleContent>
        <xs:extension base="xs:boolean">
            <xs:attribute name="type" type="xs:string"/>


            <!-- For future compatibility -->
            <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>



<xs:complexType name="drmType">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="DRMSystemID" type="xs:string" use="required"/>
```

```xml
                <xs:attribute name="protectionGateways" type="xs:string" default=""/>


                <!-- For future compatibility -->
                <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
            </xs:extension>
        </xs:simpleContent>
</xs:complexType>



<!-- Introduced in OIPF 2.3 (part of HbbTV 2.0.1) -->
<xs:complexType name="telephonyServicesType">
    <xs:simpleContent>
        <xs:extension base="xs:boolean">
            <xs:attribute name="video" type="xs:boolean" default="false"/>


            <!-- For future compatibility -->
            <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>



<!-- Introduced in OIPF 2.3 (part of HbbTV 2.0.1) -->
<xs:complexType name="playbackType">
    <xs:simpleContent>
        <xs:extension base="xs:boolean">
            <xs:attribute name="type" type="xs:string"/>


            <!-- For future compatibility -->
            <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>



<!-- Introduced in OIPF 2.3 (part of HbbTV 2.0.1) -->
```

```xml
<xs:complexType name="hasCapability">

    <!-- For future compatibility -->

    <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

</xs:complexType>



<!-- End OIPF-with-HbbTV-mods -->
<!-- ***************************************************************** -->



<!-- ***************************************************************** -->
<!-- Start HbbTV-defined -->



<!-- Introduced in HbbTV 2.0.1 -->
<xs:simpleType name="sync_tl_type">

    <xs:list itemType="termcap:sync_tl_values_type"/>

</xs:simpleType>



<!-- Introduced in HbbTV 2.0.1 -->
<xs:simpleType name="sync_tl_values_type">

    <xs:restriction base="xs:string">

        <xs:enumeration value="pts"/>

        <xs:enumeration value="ct"/>

        <xs:enumeration value="temi"/>

        <xs:enumeration value="dash_pr"/>

    </xs:restriction>

</xs:simpleType>



<!-- Introduced in HbbTV 2.0.1 -->
<xs:complexType name="graphicsPerformanceType">

    <xs:attribute name="level" type="xs:string"/>


    <!-- For future compatibility -->

    <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>
```

```xml
    </xs:complexType>



    <!-- Introduced in HbbTV 2.0.2 -->

    <xs:simpleType name="colorimetryListType">

        <xs:list itemType="termcap:colorimetryType"/>

    </xs:simpleType>



    <!-- Introduced in HbbTV 2.0.2 -->

    <xs:simpleType name="colorimetryType">

        <xs:restriction base="xs:string">

            <xs:enumeration value="bt709"/>

            <xs:enumeration value="bt2020"/>

        </xs:restriction>

    </xs:simpleType>



    <!-- Introduced in HbbTV 2.0.2 -->

    <xs:complexType name="videoDisplayFormatType">

        <xs:attribute name="width" type="xs:integer" use="required"/>

        <xs:attribute name="height" type="xs:integer" use="required"/>

        <xs:attribute name="frame_rate" type="xs:integer" use="required"/>

        <xs:attribute name="bit_depth" type="xs:integer" use="required"/>

        <xs:attribute name="colorimetry" type="termcap:colorimetryListType" use="required"/>


        <!-- For future compatibility -->

        <xs:anyAttribute namespace="##targetNamespace" processContents="lax"/>

    </xs:complexType>



    <!-- End HbbTV-defined -->
    <!-- ************************************************************** -->



</xs:schema>
```

4.14 Annex B - Support for protected content delivered via broadband

4.14.1 Content supporting both Clear Key and conventional DRM

The following text is added to the end of clause B.3.

When using this mechanism for encrypted content that could contain DRM licences or triggers, applications need to call setActiveDRM("urn:hbbtv:oipfdrm:inactive") on terminals that support the "DRM feature" before any such encrypted content is buffered. This prevents any automatic licence acquisition or similar action by DRM systems that an application can access using the oipfDrmAgent object. See also clause A.2.27.

Terminal behaviour is undefined if an application attempts to use EME to present content containing DRM licences or triggers (other than those relating to the ClearKey mechanism) on a terminal that supports the "DRM feature" unless setActiveDRM("urn:hbbtv:oipfdrm:inactive") has been called prior to buffering any encrypted content.

A new clause B.4 is added as follows;

B.4   Encrypted media extensions with DRM (informative)

The use of a Content Decryption Module to provide access to a DRM system using the Encrypted Media Extensions [66] is outside the scope of the present document. However, content providers should be aware that some terminals may support one or more DRM CDMs.

Any application that wishes to attempt to use a DRM CDM to handle protected content needs to be aware that a terminal that also supports the "DRM feature" may act on any DRM signalling within media content independently of EME API calls unless setActiveDRM("urn:hbbtv:oipfdrm:inactive") is called prior to any encrypted content being buffered. See also clause A.2.27.

4.15 Annex E - Profiles of MPEG DASH

4.15.1 Clarify URL scheme and MPD anchor rules

The first paragraph of clause E.4.5 is extended with the text shown underlined.

When the URL of an MPD is referred to by an HbbTV Application, the URL may include MPD Anchors. Terminals shall support MPD Anchors using the 't' key of the URI fragment part as defined in clause C.4 of the MPEG DASH specification ISO/IEC 23009-1 [29] as profiled in clause 10.9.2 of the DVB DASH specification [45]. Support for other MPD Anchor keys is not required

4.15.2 Frame rate support with UHD content via DASH

A new clause E.4.6 is added as shown.

E.4.6 DASH specific aspects for video
Terminals shall support those frame rates in the 25Hz/50Hz family required by DVB-DASH [45]. There is no requirement to support frame rates in the 30Hz/60Hz family.

4.15.3 No requirements on the DASH player to adapt

A new clause E.4.7 is added as shown.

E.4.7 Adaptation

The present document does not define specific algorithms for bitrate adaptation in MPEG DASH content. However, to ensure a minimum level of adaptation capability and to improve testability, the following requirements are defined:

When the terminal is presenting a DASH video AdaptationSet in which there is a lower bitrate video representation than the one currently being presented, the terminal shall adapt to a lower representation if all of the following conditions are met:

- the lower representation has a resolution of 704x396 or greater for a progressive representation or 704x576 or greater for an interlaced representation at a frame rate of 25 Hz or greater

- use of the lower representation is not precluded by user preferences

- given a series of consecutive video segments whose indicated duration totals 10 seconds or more, the time taken for the terminal to actively download those segments is greater than 5/4 times the total duration of the segments

- the terminal has not buffered more than 45 seconds beyond the current playback position

When the terminal is presenting a DASH video AdaptationSet in which there is a higher bitrate video representation than the one currently being presented, the terminal shall adapt to a higher representation if all of the following conditions are met:

- the higher representation has parameters which are supported by the terminal and the display

- use of the higher representation is not precluded by user preferences

- given a series of consecutive video segments whose indicated duration totals 30 seconds or more, the time taken for the terminal to actively download those segments is less than 1/K times the total duration of the segments, where K is given by 2 * higher_bandwidth / lower_bandwidth, where higher_bandwidth and lower_bandwidth are taken from the Representation@bandwidth attribute from the DASH MPD.

  NOTE 1: In the context of this clause, "adapt" means to switch to downloading segments from a new representation. There may be a delay before such a switch is observable by the viewer due to segments previously buffered by the terminal.

  NOTE 2: Nothing in this clause precludes adaptation in circumstances other than these, including adapting before the thresholds stated here are reached.

  NOTE 3: "the time taken for the terminal to actively download" segments does not include any time waiting before or between segment requests, nor any time when downloading of segments is suspended.

Manufacturers are strongly advised against implementing an adaptation algorithm that only satisfies these requirements. Such an algorithm is unlikely to perform well.

4.15.4 Mixing audio codecs within a single AdaptationSet

Clause E.3.2 is modified with the addition of the text shown underlined.

All Representations in an Adaptation Set shall use the same codec. However, codec profile, level and options may differ within an Adaptation Set and hence the string of the Representation@codecs attribute may also differ.

NOTE: The other requirements formerly in this clause can now be found in clauses 4.2.4, 4.3, 4.5 and 5.1.2 of the DVB DASH specification [45].

4.15.5 Errata to DVB-DASH in HbbTV 2.0.1/2

Clause E.4.1 is modified with as shown with underline / strike-through markup.

Terminals shall support the 2014 profile from ~~the DVB DASH specification~~TS 103 285 [45] as modified in the present document. MPDs that identify themselves with the profile "urn:hbbtv:dash:profile:isoff-live:2012" and not "urn:dvb:dash:profile:dvb-dash:2014" shall be supported as if they had indicated "urn:dvb:dash:profile:dvb-dash:2014" and "urn:dvb:dash:profile:dvb-dash:isoff-ext-live:2014". Low latency presentation (as defined in clause 10.20) is optional. Support for HDR dynamic mapping (see clause 5.2.7) is optional even when High Dynamic Range using PQ10 is supported.

Terminals supporting delivery of HDR, HFR or NGA content via DASH shall also support the 2017 profile from the DVB DASH specification [45].

Other profiles (e.g. the DASH-IF DASH-AVC/264 main interoperability point (see [i.19])) may be supported.

Features added to MPEG DASH [29] after the second edition are not mandatory for implementations of the present document unless stated otherwise in either the present document or DVB-DASH. For example;

1) Support for the UTCTiming element is required as defined in TS 103 285 [5] clause 4.7.3

2) Support for Preselections is necessary for terminals that support NGA codecs via DASH as defined in clause 6.7 of ETSI TS 103 285 [45] and either or both of clauses 6.3.2 or 6.8 of ETSI TS 103 285 [45].

3) The following normative clauses of MPEG-DASH [29] were added between the second and fourth editions, are not made mandatory by either the present document or ETSI TS 103 285 V1.3.1 [45] and hence are optional for implementations of the present document.

- Clause 5.3.10 Label and Group Label

- Clause 5.3.12 Initialization Set

- Clause 5.4.2 MPD Reset

- Clause 5.8.5.8 Audio Receiver Mix

- Clause 5.8.5.9 DASH MPD Adaptation Set Linking scheme

- Clause 5.8.5.10 Sub-Asset Identifier scheme

- Clause 5.8.5.11 Client Authentication and Content Access Authorization

- Clause 5.8.5.12 Audio Interactivity Descriptor

4.15.6 AdaptationSet@contentType attribute optionality

A new clause E.2.7 is added as follows.

E.2.7 Adaptation Set

Adaptation Sets without a contentType attribute may be ignored.

NOTE: Although this attribute is optional according to MPEG-DASH, the obvious situation when it is not appropriate is with multiplexed representations which are excluded from TS 103 285 [45].

4.16 Annex G - Implementer guidelines for media synchronization

4.16.1 Update inter-device sync spec from DVB blue book back to ETSI

In clause G.1, the reference to DVB Bluebook A167-2 is replaced by a reference to ETSI TS 103 286-2.

4.17 Annex H - HbbTV File Delivery Protocol (FDP)

4.17.1 Incorrect reference to organisation_id definition

In clause H.4.2, in the definition of the organisation_id field, the reference to TS 102 851 is replaced with a reference to TS 102 809.

4.18 New Annex L

4.18.1 Removing support for CSS3 navigation

A new annex L is added as shown.

Annex L (normative): Deprecated features

L.1 Introduction

Some of the features that have been required by previous versions of the present document have been deprecated. These features are listed in this Annex. Application authors should not use these features and terminals will not be required to support them in a future version of the present document.

L.2 CSS3 directional focus navigation

The requirement for terminals to support CSS3 directional focus navigation (the nav-up, nav-right, nav-down and nav-left CSS properties) in clause 10.2.2.1 has been deprecated.

4.19 New Annex M

A new annex M is added as shown.

Annex M (informative): Multi-stream synchronization examples

M.1 Alternate audio track for a single broadcast TV program

M.1.1   Introduction

Figure M.1 illustrates the example of a single broadcast TV programme that has an alternate audio track that is available via broadband. Examples uses for alternate audio tracks include accessibility (audio description for the partly sighted, clean audio for the hard of hearing) as well as additional languages.
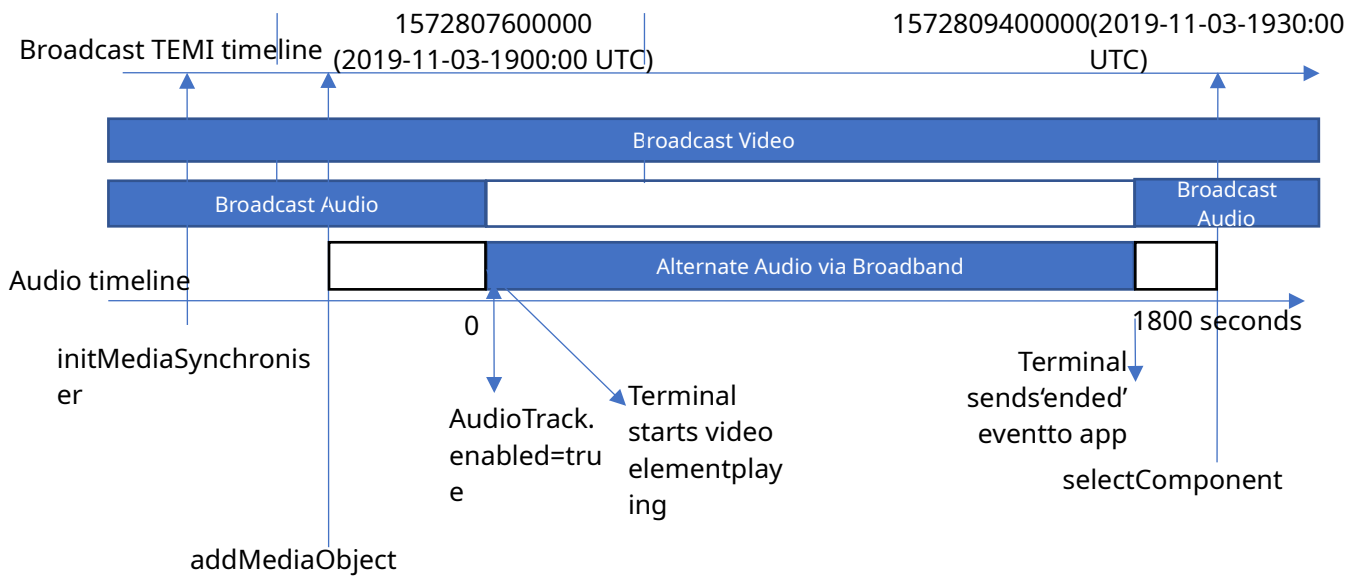
Broadcast TEMI timeline

1572807600000
(2019-11-03-1900:00 UTC)

1572809400000(2019-11-03-1930:00 UTC)

Broadcast Video

Broadcast Audio

Broadcast Audio

Audio timeline

Alternate Audio via Broadband

0

1800 seconds

initMediaSynchroniser

Terminal sends 'ended' event to app

AudioTrack. enabled=true

Terminal starts video element playing

selectComponent

addMediaObject

Figure M.1:   Example of alternate audio track for a single broadast TV programme

This example assumes the following context;
- The broadcast program from 19:00 UTC to 19:30 UTC on November 3rd 2019 has an alternate audio track

- The broadcast has a TEMI timeline that encodes UTC as milliseconds since the Unix epoch. 19:00 UTC corresponds to a value on the TEMI timeline of 1572807600000. .

- The user is already watching broadcast TV some time before 19:00 UTC.

- The alternate audio track is distributed via DASH using a static MPD one Period with its timeline starting from zero.

- The alternate audio track has AdaptationSet@role set to "alternate" and other information set such that the terminal will not automatically select in "Component selection by the terminal". For example;

  - @lang being a language that is not in the user`s audio language preferences
  - The terminal does not have a user setting or preference for clean audio but the Adaptation Set includes an accessibility element with @schemeIdUri = "urn:tva:metadata:cs:AudioPurposeCS:2007" and @value = "2" for the hard of hearing (see TS 103 285 [] clause 6.1.2).

M.1.2   *User viewing service before programme starts*

When the user is viewing the service before 19:00, a broadcast-related application runs and does the following;
- Creates a video/broadcast object & calls the `bindToCurrentChannel` method

- Creates a `mediaSynchroniser` object and initialises it with the video/broadcast object and the timeline selector for the TEMI timeline

  NOTE 1:   There is no need to wait for an `SynchroniserInitialised` event as the call

to the load method may in practice take long enough for the previous method call to complete.

- Creates an HTML5 audio element with the source pointing to the DASH MPD for the alternate audio track

- Calls the `load` method on the HTML5 audio element and waits for the 'loadeddata' event

- Registers a listener for `error` events from the `mediaSynchroniser` object

- Calls `addMediaObject` with the following arguments;

  - mediaObject = the HTML5 audio element
  - timelineSelector = "urn:dvb:css:timeline:mpd:period:rel:50:1"
  - correlationTimestamp = {tlvMaster= 1572807600000, tlvSlave=0}
  - tolerance = 40ms
  - multiDecoderMode = false

- Receives an error event with code 11 indicating a transient error of the `mediaSynchroniser` due to there being no valid content in the alternate audio track at this time. This should be ignored as media synchronisation will automatically resume after this error code.

  NOTE 2: In some early implementations, the call to `addMediaObject` may fail with an error event with a different code – one from which media synchronisation does not automatically resume. On such implementations, applications should wait until the TEMI timeline advances past 19:00 and call the method again.

- Polls the `currentTime` of the `mediaSynchroniser` object waiting for 19:00;

  - The application may unselect the broadcast audio by calling the `unselectComponent( COMPONENT_TYPE_AUDIO)` on the video/broadcast object however this is not required as the broadcast audio will be automatically unselected.

- Sets the `enabled` property of the audio track of the audio element to true.

  NOTE 3: This starts an asynchronous process in the terminal including the following;
  - changing the selected audio component from the broadcast audio to the broadband audio and,
  - resuming media synchronisation after the transient error

  NOTE 4: Media synchronisation will be visible to applications through a number of events. No action is needed.

- Receives a `SelectedComponentChanged` on the video/broadcast object when the broadcast audio is stopped

- Receives a `playing` event on the audio element when the alternate audio is about to be started

- Recives zero or more sets of the following events on the audio element - `seeking` / `timeupdate` / `seeked`

- Receives `timeupdate` events as a result of "the usual monotonic increase of the current playback position during normal playback" (see "time marches on steps" in HTML5 []). Applications need to be careful to distinguish these timeupdate events from timeupdate events fired for other reasons.

- Receives a `SyncNowAchievable` event

The following happens at the end of the alternate content;
- Before 19:30, the application registers a listener on the HTML5 audio element and waits for the ended event.

- When the end of the alternate content is reached;

    - The terminal posts the `ended` event to the application
    NOTE 5:   Clause 9.6.2 of the present document requires the `ended` event to be received within 250ms of the end of the alternate audio.

    - A transient error is generated on the `mediaSynchroniser` object with code 2 and the HTML5 audio element is removed from on the `mediaSynchroniser` by the terminal.
    NOTE 6:   In some early implementations, no transient error will be generated and the audio element may not be removed from the `mediaSynchroniser` object. The application will then need to do this explicity.

    - The application calls the `selectComponent` method on the video/broadcast object to ensure the original audio component is presented.
    NOTE 7:   Implementations may automatically rervert to the broadcast audio when the end of the alternate audio is reached in which case the above method call would be redundant.

- The application calls `removeMediaObject` with the video/broadcast object

    NOTE 8:   In some early implementations, this call may dispatch an error event with code 8. This can be ignored as there is no mechanism to completely dispose of a `mediaSynchroniser` object on such implementations.

M.1.3   User starts viewing service during programme

If the user starts viewing the service during the programme then what happens is the same as if they are viewing the service before the programme except as follows;
- The application registers a handler for `MediaObjectAdded` events before calling `addMediaObject`.

- The call to `addMediaObject` is not followed by a transient error.

- When the application receives a `MediaObjectAdded` event, it sets `AudioTrack.enabled` to 'true'.

    NOTE: This event is not supported in some early implementations. If not supported then applications are recommended to use standard JavaScript features to introduce a delay between the call to `addMediaObject` and setting `AudioTrack.enabled` to 'true'.

M.2 Alternate audio track for a broadcast TV service

M.2.1   Introduction

Figure M.2 illustrates the example of a broadcast TV service that has an alternate audio track that is available via broadband. Examples uses for alternate audio tracks include accessibility (audio description for the partly sighted, clean audio for the hard of hearing) as well as additional
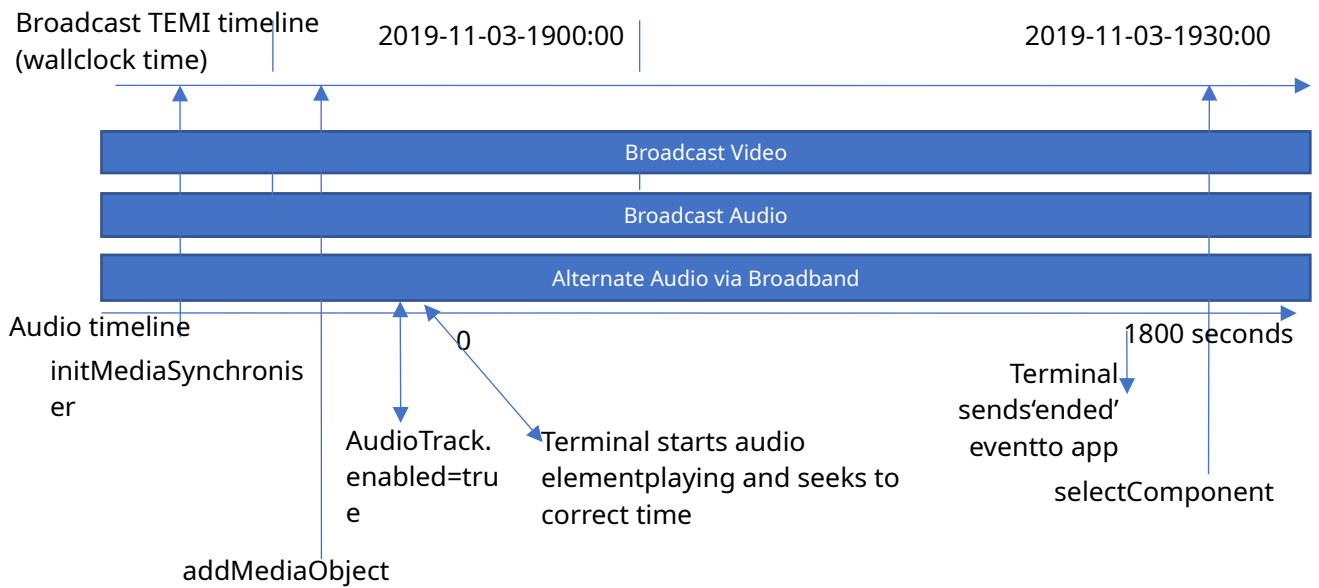
languages.

Figure M.2:   Example of alternate audio track for a broadast TV service

In contrast to clause M.1 and figure M.1, the alternate audio track is distributed as a live DASH stream. The steps involved are as follows;

- Same as M.1 upto and including call to `addMediaObject`.

- No transient error will be generated following the call to `addMediaObject`.

- The application sets `AudioTrack.enabled` to true in order to forces the selection of the alternate audio regardless of any user preferences (e.g. if the audio is clean audio and the terminal does not have a user preference for that).

- The terminal starts the audio element playing and seeks to the correct time. Events are fired on the video element as defined by HTML5 [] – e.g. playing, seeking, seeked, timeupdate.

  NOTE 1:   There may be more than one seek operation. For example, a seek before starting playing based on an estimate of how long starting playing might take and then a seek after starting playing

- The terminal plays the broadcast video and the alternate audio in sync. The terminal may perform seeks on the broadband audio if needed in order to keep the video and audio in sync within the tolerance specified.

- The application polls the `currentTime` on the `mediaSynchroniser` waiting for 19:30.

- The application performs the following steps in any order;

  - Calls the `selectComponent(COMPONENT_TYPE_AUDIO)` on the video/broadcast object to return to the broadcast audio
  - Calls `removeMediaObject` to remove the audio element from the `mediaSynchroniser`, pause it and then remove it as defined in Annex J of the present document.