HbbTV® Specification Version 1.5

16th March 2012

Contents

1 Scope

The present document extends the HbbTV specification to address a number of immediate requirements which will clearly be addressed at a national level if they are not promptly addressed at a European level. The principal requirements of this sort are HTTP adaptive streaming, a common encryption scheme allowing the use of multiple DRM technologies and access to DVB-SI EIT schedule information.

It is anticipated that the present document will be integrated with the text of ETSI TS 102 796 V1.1.1 and the errata to that document and proposed to ETSI as version 1.2.1 of TS 102 796. Hence places where the specification version number is used (e.g. application signalling, HTTP user agents) use 1.2.1.

2 References

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 102 796 (V1.1.1): "Hybrid Broadcast Broadband TV".

[2] ETSI EN 300 468 (V1.10.1): "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".

[3] ISO/IEC 23009-1 : Information technology –Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats

[4] ISO/IEC 23001-7 : Information technology -- MPEG systems technologies -- Part 7: Common encryption in ISO base media file format files

NOTE: This was previously Annex I of [5]

[5] ISO/IEC 14496-12 ISO Base File Format

[6] Open IPTV Forum Release 1 specification, Volume 5 (V1.1): "Declarative Application Environment".

[7] Open IPTV Forum Release 2 specification, Volume 2 (V2.1): "Media Formats".

NOTE: Available at http://www.oipf.tv/downloads.html.

[8] ETSI TS 101 154 (V1.10.1): “Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream”

[9] ISO/IEC 14496-12:2008/FDAM 3:2011(E): “Information technology — Coding of audio-visual objects — Part 12: ISO base media file format, AMENDMENT 3: DASH support and RTP reception hint track processing”

[10] ETSI TS 102 822-3-1 V1.7.1 (2011-11): “Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime");Part 3: Metadata;Sub-part 1: Phase 1 - Metadata schemas”

[11] ETSI TS 102 366 V1.2.1 (2008-08): “Digital Audio Compression (AC-3, Enhanced AC-3) Standard”

...

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 102 796 [1] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TS 102 796 [1] and the following apply:

AES	Advanced Encryption Standard
AVC	Advanced Video Coding
BFF	Base File Format
CTR	Counter
DASH	Dynamic Adaptive Streaming over HTTP
DVB-SI	DVB Service Information
ISO	International Organization for Standardization
ISOBMFF	ISO Base Media File Format
KID	Key Identifier
MPD	Media Presentation Description

4 Overview

4.1 Relationship with TS 102 796

The present document is an extension to the Hybrid Broadcast Broadband TV specification [1] and full compliance with that document is required.

4.2 Specification Overview

The present document largely references parts of already available standards and specifications and adapts those parts where necessary. The most significant documents directly referenced from the present document are the following:

- ETSI TS 102 796 V 1.1.1 “Hybrid Broadcast Broadband TV" [1].
- Open IPTV Forum Release 1 Volume 5 - Declarative Application Environment of the Open IPTV Forum [6].
- MPEG DASH – formally known as ISO/IEC 23009-1 : Information technology –Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats [3].
- MPEG CENC – formally known as ISO/IEC 23001-7 : Information technology -- MPEG systems technologies -- Part 7: Common encryption in ISO base media file format files[4].

Figure 1 shows a graphical overview of the relationship between the contents of the present document and the above mentioned specifications.

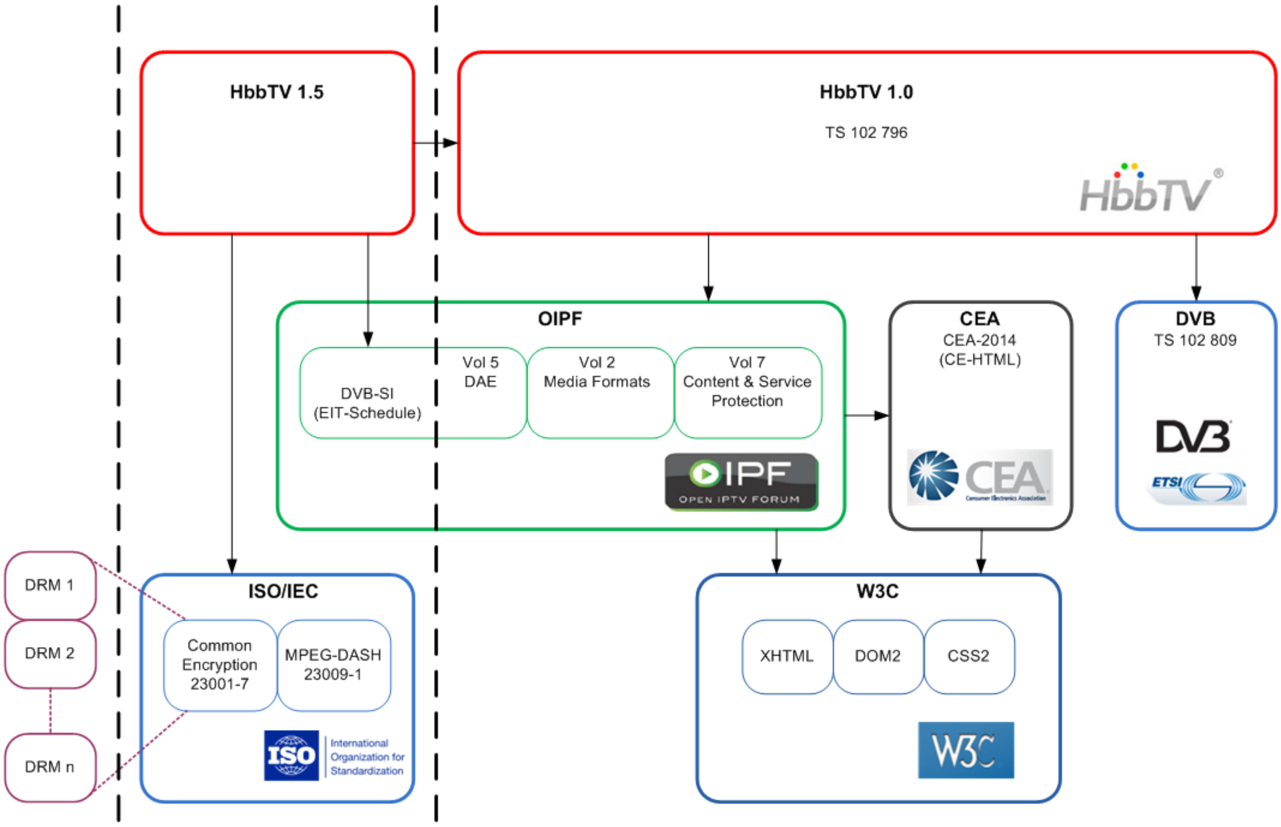

Figure 1: Specification Overview

5 Void

This clause is intentionally left blank.

6 Void

This clause is intentionally left blank .

7 Formats and protocols

7.1 Void

This clause is intentionally left blank .

7.2 Broadcast-specific formats and protocols

7.2.1 Signalling of Applications

In table 5 of TS 102 796 [1], “Supported application signalling features”, the row for section 5.2.5, “platform profiles” shall be revised as follows:

The version.minor field shall be changed to ‘2’.

The following text shall be added to the end of the Notes: “The 3 most significant bits of the `application_profile` are reserved for future use. Terminals shall ignore any profile where these are non-zero.”

7.3 Broadband-specific formats and protocols

7.3.1 HTTP Adaptive Streaming

HTTP adaptive streaming shall be supported using MPEG DASH as defined in annex B.

7.3.2 HTTP User-Agent Header

The HTTP User-Agent header shall be as in TS 102 796 [1] with “HbbTV/1.1.1” replaced with “HbbTV/1.2.1”.

8 Void

This clause intentionally left blank.

9 System Integration

9.1 Presentation of adaptive bitrate content

Terminals shall support applications setting the data attribute of a CEA-2014 A/V control object to a URL referencing an MPD as defined in DASH [3] and identified by the MIME type in annex C of that document. In order to play the content, the terminal shall fetch the MPD from the URL, interpret the MPD and select an initial set of representations. If at any time the MPD is found to be not valid according to the XML schema or semantics defined in DASH[3], the A/V control object shall go to play state 6 (‘error’) with error value 4 (‘content corrupt or invalid’).

If the content access streaming descriptor defined in annex E.2 of the OIPF DAE specification [6] is supported then terminals shall support the <ContentURL> element of this descriptor referencing an MPD as defined in DASH [3]. In these circumstances, the other requirements from the preceeding paragraph shall apply.

10 Capabilities

10.1 Void

This clause intentionally left blank.

10.2 Void

This clause intentionally left blank.

10.3 XML Capabilities

The xmlCapabilities property of the application/oipfCapabilities object (as defined in clause 7.15.3 of the OIPF DAE specification [6]) shall include the following;

- The <clientMetadata> element shall include the <dvb-si> metadata system/protocol name as defined in section 9.3.7 of the OIPF DAE specification [6].

- Support for HTTP adaptive streaming shall be indicated by including <video_profile> elements including “dash” as the transport attribute.

Hence the XML text to be used for the xmlCapabilities property of the application/oipfCapabilities embedded object for the base level of features (see clause 10.2.4 of TS 102 796 [1]) shall be as follows;

```
<profilelist>
<ui_profile name=”OITF_HD_UIPROF+DVB_S” />
<clientMetadata type=”dvb-si”>true</clientMetadata>
<video_profile name=”TS_AVC_SD_25_HEAAC” type=”video/mpeg” />
<video_profile name=”TS_AVC_HD_25_HEAAC” type=”video/mpeg” />
<video_profile name=”MP4_AVC_SD_25_HEAAC” type=”video/mp4” />
<video_profile name=”MP4_AVC_HD_25_HEAAC” type=”video/mp4” />
<video_profile name=”MP4_AVC_SD_25_HEAAC” type=”video/mp4” transport=”dash”/>
```

```
   <video_profile name=”MP4_AVC_HD_25_HEAAC” type=”video/mp4” transport=”dash”/>
   <audio_profile name=”MPEG1_L3” type=”audio/mpeg”/>
   <audio_profile name=”HEAAC” type=”audio/mp4”/>
   </profilelist>
```

As specified in clause 7.3.1.1 of TS 102 796 [1], terminals shall support Enhanced-AC3 for the broadband channel when it is supported by the broadcast channel. The format label “E-AC3” (as defined in clause 8.1.3 of the OIPF Release 2 Media Formats specification [12]) shall be used to indicate the support of Enhanced AC-3 by the terminal.

Where E-AC3 is applicable, the XML text to be used for the xmlCapabilities property of the application/oipfCapabilities embedded object for the base level of features (see clause 10.2.4 of TS 102 796 [1]) shall be as follows;

```
<profilelist>
<ui_profile name=”OITF_HD_UIPROF+DVB_S” />
<clientMetadata type=”dvb-si”>true</clientMetadata>
<video_profile name=”TS_AVC_SD_25_E-AC3” type=”video/mpeg” />
<video_profile name=”TS_AVC_HD_25_E-AC3” type=”video/mpeg” />
<video_profile name=”TS_AVC_SD_25_HEAAC” type=”video/mpeg” />
<video_profile name=”TS_AVC_HD_25_HEAAC” type=”video/mpeg” />
<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_HEAAC” type=”video/mp4” />
<video_profile name=”MP4_AVC_HD_25_HEAAC” type=”video/mp4” />
<video_profile name=”MP4_AVC_SD_25_E-AC3” type=”video/mp4” transport=”dash”/>
<video_profile name=”MP4_AVC_HD_25_E-AC3” type=”video/mp4” transport=”dash”/>
<video_profile name=”MP4_AVC_SD_25_HEAAC” type=”video/mp4” transport=”dash”/>
<video_profile name=”MP4_AVC_HD_25_HEAAC” type=”video/mp4” transport=”dash”/>
<audio_profile name=”MPEG1_L3” type=”audio/mpeg”/>
<audio_profile name=”HEAAC” type=”audio/mp4”/>
</profilelist>
```

The modifications to the corresponding XML text defined in clause 10.2.4 of TS 102 796 [1] also apply to this XML text. For example, “DVB_S” can be replaced by the appropriate string(s) for the supported broadcast delivery system(s).

11 Security

11.1 Protected content via Broadband

Support for delivering protected content via the broadband channel is optional in the present document. If this is supported and the content is provided in an ISO base media file format, then one mechanism by which the content may be encrypted is MPEG common encryption as defined by CENC [4] and constrained by Annex C of the present document.

Annex A – OIPF DAE Specification Profile

A.1 Access to EIT Schedule Information

Access to DVB-SI EIT schedule information (as defined in EN 300 468 [2]) shall be supported with the following sections and sub-sections of the OIPF DAE specification [6]. This replaces the corresponding rows of Table A.1: “Section-by-section profile of the OIPF DAE specification” in the HbbTV specification [1].

Table A.1: Amended section-by-section profile of the OIPF DAE specification

Section, sub-section	Reference in DAE [6]	Status in the present document	Notes	Security
The application/oipfSearchManager embedded object	7.12.1	M(*)	The `guideDaysAvailable` and `onMetadataUpdate` properties are not included. For the `createSearch` method, only the value '1' of the `searchTarget` parameter is included.	Broadcast-related
The MetadataSearch class	7.12.2	M(*)	Only the value '1' of the `searchTarget` property is included. For the `createQuery` method, only the following case-insensitive values for the `field` parameter are included - “startTime”, “title”, “programmeID”. These shall correspond to the properties of the same name. The value “7” for the `comparison` property is not supported. The `addRatingConstraint` and `addCurrentRatingConstraint` methods are not included. The `orderBy` method is not included – all search results shall be returned ordered first by channel, in the same order as presented to applications through a ChannelList object, then by start time in ascending order.	Broadcast-related
The Query class	7.12.3	M(*)	The values of the `field` and `comparison` properties are constrained as defined above for the parameters of the same name on the `createQuery` method of the MetadataSearch class.	Broadcast-related
The SearchResults class	7.12.4	M		Broadcast-related
The MetadataSearchEvent class	7.12.5	M(*)	The value “2” for the `state` property is not supported.	Broadcast-related
The MetadataUpdateEvent class	7.12.6	NI		
The programme class				

Basics	7.16.2.1, 7.16.2.2	M(*)	The following properties are required: - name - programmeID - programmeIDType - description - longDescription - startTime - duration - channelID - parentalRating The constants defined in clause 7.16.2.1 shall be supported however support for CRIDs is outside the scope of the present document. The following method is required for Programme objects returned by the programmes property of the video/broadcast object: - getSIDescriptors All other properties and methods are not included.	Broadcast-related

The meanings of the Status and Security columns shall be as defined in tables A.2 “Table A.2: Key to security column” and A.3 “Table A.3: Key to status column” of the HbbTV specification [1].

A.2 Modifications, Extensions and Clarifications

A.2.1 Metadata Property Mappings

A.2.1.1 The Channel class

Property name	Source	Comment
channelType	Assigned by the terminal.	Assigned by the terminal based on the service type signalled in SDT/service descriptor/service type or TYPE_OTHER if not known.
ccid	Assigned by the terminal.	Unique identifier for the channel
onid	Assigned by the terminal or by the application.	Assigned by the terminal from SDT. onid or by the application using the value passed in to the createChannelObject() method.
tsid	Assigned by the terminal or by the application.	Assigned by the terminal from SDT.tsid or by the application using the value passed in to the createChannelObject() method.
sid	Assigned by the terminal or by the application.	Assigned by the terminal from SDT.sid or by the application using the value passed in to the createChannelObject() method.
name	Assigned by the terminal.	Assigned by the terminal from SDT/service descriptor/service name or undefined for Channel objects created by calls to the createChannelObject() method.
dsd	Assigned by the application.	Assigned by the application using the delivery system descriptor passed in to the createChannelObject() method, or implementation-dependent in all other cases.

A.2.1.2 The Programme class and related classes

Property name	Source	Programme Class Property	ScheduledRecording Class Property	Recording Class Property (See note 1)	Download Class Property
state	Assigned by the terminal.	N/A	N/A	Assigned and updated by the terminal as recording is carried out.	Assigned by the terminal, ref 7.4.4.1 of OIPF.
id	Assigned by the terminal.	N/A	N/A	Unique internal identifier for	Unique internal identifier for downloaded content.

Property name	Source	Programme Class Property	ScheduledRecording Class Property	Recording Class Property (See note 1)	Download Class Property
				recordings.	
startPadding	Assigned by the terminal or the application.	N/A	Default value assigned by the terminal; may be overridden by the application.	Derived by the terminal from the corresponding property on the ScheduledRecording object.	N/A
endPadding	Assigned by the terminal or the application.	N/A	Default value assigned by the terminal; may be overridden by the application.	Derived by the terminal from the corresponding property on the ScheduledRecording object.	N/A
repeatDays	Set by the application	N/A	The days on which the recording will be repeated as assigned by the application	Derived by the terminal from the corresponding property on the ScheduledRecording object.	N/A
name	Assigned by the terminal.	Assigned by the terminal from EIT/short_event_descriptor/event name	Derived from Programme object when recording is scheduled. For manual recordings, assigned by the terminal (see note).	Derived by the terminal from the corresponding property on the ScheduledRecording object.	Assigned by the terminal from CADD.Title.
description	Assigned by the terminal.	Assigned by the terminal from EIT/short_event_descriptor/description	Derived from Programme object when recording is scheduled	Derived by the terminal from the corresponding property on the ScheduledRecording object.	Assigned by the terminal from CADD.Synopsis if present.
longDescription	Assigned by the terminal.	Assigned by the terminal from EIT/extended_event_descriptor/text	Derived from Programme object when recording is scheduled	Derived by the terminal from the corresponding property on the ScheduledRecording object.	N/A
startTime	Assigned by the terminal or application.	Assigned by the terminal from EIT/event/start_time.	Derived from Programme object when recording is scheduled. Assigned by the application for recordings scheduled using the recordAt() method. For manual recordings initiated via a native UI, assigned by the terminal (see note).	Derived by the terminal from the corresponding property on the ScheduledRecording object.	Assigned by the terminal based on the startTime argument of RegisterDownload().
recording StartTime	Assigned by the terminal.	N/A	N/A	The actual start time of the recording.	N/A

Property name	Source	Programme Class Property	ScheduledRecording Class Property	Recording Class Property (See note 1)	Download Class Property
timeElapsed	Assigned by the terminal.	N/A	N/A	N/A	Assigned by the terminal.
timeRemaining	Assigned by the terminal.	N/A	N/A	N/A	Assigned by the terminal.
duration	Assigned by the terminal or application.	Assigned by the terminal from EIT/event/duration.	Derived by the terminal from the duration property of the Programme object when the recording is scheduled. Assigned by the application for recordings scheduled using the recordAt() method. For manual recordings initiated via a native UI, assigned by the terminal (see note).	Derived by the terminal from the corresponding property on the ScheduledRecording object.	Assigned by the terminal from CADD.contentURL@Duration if present.
recordingDuration	Assigned by the terminal.	N/A	N/A	The actual duration of the recording.	N/A
channel	Assigned by the terminal.	N/A	Derived by the terminal from the ccid property of the Programme object when the recording is scheduled. Derived by the terminal from the value passed by the application for recordings scheduled using the recordAt() method. For manual recordings initiated via a native UI, assigned by the terminal (see note).	Derived by the terminal from the corresponding property on the ScheduledRecording object.	N/A
channelID	Assigned by the terminal.	Populated from ccid of the channel carrying this programme.	N/A	N/A	N/A
programmeID	Assigned by the terminal.	If a programme CRID is not provided in the EIT for the programme	Derived from Programme object when recording is scheduled	Derived by the terminal from the corresponding property on the ScheduledRecordin	N/A

Property name	Source	Programme Class Property	ScheduledRecording Class Property	Recording Class Property (See note 1)	Download Class Property
		then this shall be assigned by the terminal from EIT/event_idand it shall be encoded as a decimal integer. Otherwise this is outside the scope of the present document.		g object.	
programmeIDType	Assigned by the terminal.	Assigned by the terminal.	Derived from Programme object when recording is scheduled	Derived by the termnal from the corresponding property on the ScheduledRecording object.	N/A
parentalRatings	Assigned by the terminal	Populated from EIT/parental_rating_descriptor/rating, where present.	Derived from Programme object when recording is scheduled. For manual recordings initiated via a native UI, assigned by the terminal (see note).	Derived by the terminal from the parentalRating property on the ScheduledRecording object.	Assigned by the terminal from CADD.parentalRating if present.
contentID	Assigned by the terminal.	N/A	N/A	N/A	Assigned by the terminal from CADD.contentID if present.
totalSize	Assigned by the terminal.	N/A	N/A	N/A	Assigned by the terminal from CADD.contentURL@size, then updated to actual size on disk at end of download.
contentURL	Assigned by the terminal.	N/A	N/A	N/A	Assigned by the terminal from CADD.contentURL.
drmControl	Assigned by the terminal.	N/A	N/A	N/A	Assigned by the terminal from CADD.DRMControlInformation if present.
transferType	Assigned by the terminal.	N/A	N/A	N/A	Assigned by the terminal from CADD.contentURL.transferType .
originSite	Assigned by the terminal.	N/A	N/A	N/A	Assigned by the terminal from CADD.originSite
originSiteName	Assigned by the terminal.	N/A	N/A	N/A	Assigned by the terminal from CADD.originSiteName if present.
iconURL	Assigned by the terminal.	N/A	N/A	N/A	Assigned by the terminal from CADD.iconURL if present.
Note: Manual recordings initiated via a native UI will not be accessible to HbbTV applications.					

Where there are multiple language versions of a text field derived from DVB-SI tables, the terminal should select one in accordance with pre-defined user preferences.

A.2.1.2 The AVComponent class

When an instance of the `AVComponent` class refers to a DASH audio media content component which is identified as being audio description (as defined in clause B.2.4 Role Related Requirementsbelow), the `audioDescription` property of the `AVComponent` shall be set to true.

When an instance of the `AVComponent` class refers to a DASH audio media content component, the `language` property shall be set from value of the lang attribute in the MPD – whether set explicitly for that component or inherited. If the `lang` attribute in the MPD is not set for a media content component then the value of the `language` property in the corresponding AVComponent class shall be `Undefined`. The contents of the language field in the media header “mdhd” of the track shall be ignored.

A.2.2 Extension to the Channel class

This class shall be extended with the following additional property.

`readonly Integer nid`
For channels of type ID_DVB_*, this shall be defined as follows. • If the channel is contained within a DVB Transport Stream, then this property shall be determined as follows: - If during the terminal configuration process, a network_id value was selected (either explicitly or implicitly) and the NIT subtable with that network_id value was used by the terminal to discover the correct delivery system descriptor of this channel, then the value of this property shall be that network_id value. - Otherwise, if there is exactly one NIT 'actual' subtable in the Transport Stream that is carrying the channel then the value of this property shall be the network_id in that subtable. Terminals are not required to update the value if it changes dynamically in the broadcast Transport Stream. - Otherwise the value shall be `Undefined`.

A.2.3 play speed approximation

In annex B of the OIPF DAE specification [6], where Requirement 5.7.1.f bullet 11 of CEA-2014 is modified as follows;

> If the current media can be played at the specified speed, true is returned. Otherwise, false is returned and neither the play state nor the speed is not changed.

The requirement beginning “Otherwise, false is returned” shall not apply. The terminal shall select an appropriate speed. As defined in section 8.2.3.1 of OIPF, a PlaySpeedChanged event shall be generated indicating the actual playback speed. True shall be always be returned.

Annex B – Profiles of MPEG DASH

B.1 Introduction (informative)

This annex starts from MPEG DASH [3] and defines a profile that adds additional requirements to improve testability and interoperability.

The present document references only one profile of DASH – the "ISO Base media file format live profile". This profile, on which the HbbTV profile is based, supports both live and on-demand steaming of ISOBMFF content. It supports template-based addressing of short time-aligned Segments that may be concatenated without overlap or video splicing. It supports independently addressable track fragment segments.

B.2 Requirements relating to the MPD

B.2.1 Profile definition

The document defines a sub-profile of the MPEG DASH ISO Base media file format live profile. This sub-profile is identified with the URI "urn:hbbtv:dash:profile:isoff-live:2012" and is called the "HbbTV ISOBMFF Live" profile. All of the requirements and restrictions for the MPEG DASH ISO Base media file format live profile shall apply.

Terminals may raise an error to the application when a referenced MPD does not contain this profile in the `@profiles` attribute. Terminals shall be able to play the content described by the profile-specific MPD (as defined in section 8.1 of DASH [3]) (but not necessarily other Adaptation Sets or Representations in the MPD discarded as part of the process of deriving the profile-specific MPD).

The following clauses of Annex B define the additional restrictions and requirements on an MPD identified as conforming to this profile, as well as requirements on terminals when playing such content.

The size of a MPD shall not exceed 100kbytes.

The content referenced by the profilespecific MPD shall only be encoded using the audio and video codecs defined in section 7.3.1 of TS 102 796 [1].

B.2.2 Numerical requirements

The profile-specific MPD shall conform to the following constraints:

Periods

There shall be no more than "N_{per}" Periods in an MPD that shall be temporally sequential. The behaviour of a terminal is undefined for MPDs containing more than "N_{per}" Periods.

Adaptation Sets

There shall be no more than "N_{adset}" Adaptation Sets per Period in an MPD. The behaviour of a terminal is undefined for MPDs containing Periods with more than "N_{adset}" Adaptation Sets. If there is more than one video Adaptation Set, exactly one must be labelled with a Role@value of "main" from the urn:mpeg:dash:role:2011 CS, to allow the terminal to identify the default adaptation set. Similarly if there is more than one audio Adaptation Set, exactly one must be labelled with a Role@value of "main" to allow the terminal to identify the default adaptation set.

Representations

There shall be no more than "N_{rep}" Representations per Adaptation Set in an MPD. The behaviour of a terminal is undefined for MPDs containing Adaptation Sets with more than "N_{rep} " Representations.

Table B.1 defines these values for the present document:

Table B.1: Maximum numeric requirements on HbbTV ISOBMFF Live MPD

Parameter	Value
N_{per}	32
N_{adset}	16

N_{rep}	16

B.2.3 Metadata Requirements

The profile-specific MPD shall provide the following information for all Representations, whether defined as part of the Representation or inherited.

- For video Representations: @width, @height, @frameRate and @scanType
- For audio Representations: @audioSamplingRate, AudioChannelConfiguration, @lang

Note: @lang is an attribute of the AdaptationSet element and is inherited by its Representations.

B.2.4 Role Related Requirements

The MPD shall adopt the DASH role scheme (urn:mpeg:dash:role:2011) as defined in MPEG-DASH 5.8.5.5, in order that Adaptation Sets can be uniquely differentiated.

Where there are multiple Adaptation Sets of the same component type (e.g. 2 x video Adaptation Sets), terminals shall by default select the Adaptation Set that is signalled with a `Role` element with a value of “main” from the urn:mpeg:dash:role:2011 CS. This does not imply that a terminal must render the “main” Adaptation Set if it understands the logic and signalling of other potentially more appropriate Adaptation Sets or is required by an application to select a different Adaptation Set.

The MPD shall identify audio description streams usingthe `Role` and `Accessibility` descriptors as defined in the following table. Furthermore for receiver mix AD the associated audio stream shall use `depdendencyId` to point out the depdendency to the main representation and hence also point out that the associated audio stream shall not be provided as a representation on it own. Terminals shall ignore audio streams with other `Role` and `Accessibility` descriptor attributes that they do not understand.

Table B.2 : Role and Accessibility descriptor values for Audio Description

		Role descriptor	Accessibility descriptor
schemeIdUri		**urn:mpeg:dash:role:2011**	**urn:tva:metadata:cs:AudioPurposeCS:2007**
value	Broadcast mix AD	alternate	“1”
	Receiver mix AD	commentary	“1”

For example, broadcast mix audio descriptions would be indicated as follows:

```
<Role schemeIdUri=”urn:mpeg:dash:role" value="alternate"/>
<Accessibility schemeIdUri="urn:tva:metadata:cs:AudioPurposeCS:2007" value="1"/>
```

A schematic example for receiver mix audio descriptions:

```
<!-- English Audio, main -->
<AdaptationSet ..>
   <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" />
   <Representation id="a0" bandwidth="320000"/>
</AdaptationSet>
<!-- English Audio, visually impaired for receiver mixing AD-->
<AdaptationSet ..>
   <Accessibility schemeIdUri="urn:tva:metadata:cs:AudioPurposeCS:2007” value=“1”/>
   <Role schemeIdUri="urn:mpeg:dash:role:2011" value="commentary" />
   <Representation id="a1" dependencyId="a0" bandwidth="64000"/>
</AdaptationSet>
```

B.2.5 Audio Channel Configuration Requirements

In order for the terminals to know the number of audio channels in a representation the MPD should include the Audio Channel Configuration to correctly represent the audio channel configuration

For HE-AAC the Audio Channel Configuration shall use “urn:mpeg:dash:23003:3:audio_channel_configuration:2011” schemeURI with the value set to an integer number as defined in [3]. For example, for a stream with C, L, R, Ls, Rs, LFE, the value shall be "6", as follows:

```
<AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="6"/>
```

For E-AC-3 the Audio Channel Configuration shall use the “urn:dolby:dash:audio_channel_configuration:2011” schemeURI. The value element shall contain a four digit hexadecimal representation of the 16 bit field that describes the channel assignment as defined by Table E.5 in [11] where left channel is MSB. For example, for a stream with L, C, R, Ls, Rs, LFE, the value shall be "F801" (hexadecimal equivalent of the binary value 1111 1000 0000 0001) as follows:

```
<AudioChannelConfiguration
schemeIdUri="urn:dolby:dash:audio_channel_configuration:2011" value="F801"/>
```

B.3 Restrictions on Content

B.3.1 Restrictions on File Format

B.3.1.1 ISO Base Media File Format

The following restrictions shall apply for content referenced from an profile-specific MPD and carried in the ISO base media file format as defined by ISO/IEC 14496-12 [5]:

- The movie fragment box (‘moof’) shall contain only one track fragment box (‘traf’).
- The track run box (‘trun’) shall allow negative composition offsets (as defined in [9]) in order to maintain audio visual presentation synchronization.

B.3.2 Restrictions on Adaptation Sets

The following additional restrictions shall apply across the set of Representations in an Adaptation Set in a profile-specific MPD:

- Each Representation shall contain only one media component, i.e. a single audio or video track. Other non-media components (e.g. encryption keys) may be present if applicable.
- All ISO BMFF Representations shall have the same track_ID in the track header box and track fragment header box.
- Initialization Segment shall be common for all Representations and the following shall hold:
  - For video Representations, width and height values in track header boxshall have the nominal display size in square pixels after decoding, AVC cropping, and rescaling.
  - All information necessary to decode any Segment chosen from Representations shall be provided in the Initialization Segment. For example, movie box for video Representation shall contain AVC decoder configuration records including all encoding parameters (i.e. Sequence Parameter Sets ans Picture Parameter Sets) used for Representations in the Adaptation Sets.
- Each Segment shall consists of a whole, self-contained movie fragment.
- Segments shall be at least 1s long, except for the last segment in an MPD which may be shorter.
- Each video Segment shall have a duration of not more than fifteen seconds.
- Each audio Segment shall have a duration of not more than fifteen seconds.

There is no requirement for all of the transitions between all the Representations of a media content component to be ones that terminals are required to support as defined in clause B.4.2 Transitions between Representations. Adaptation Sets may include Representations which can only be reached by transitions other than those which terminals are required to support.

B.4 Requirements on Terminals

B.4.1 DASH Profile Support

Terminals shall support the HbbTV ISOBMFF Live profile. Other profiles may be supported.

B.4.2 Transitions between Representations

B.4.2.1 Video Tracks

During playback of adaptively streamed content encoded using AVC, terminals shall support transitions between video Representations as follows:

1) Between Representations which differ by bit-rate (note 1).

2) Between Representations which differ by profile and/or level (note 2).

3) Between Representations which differ by full-screen resolution (e.g. 1920x1080 and 720x576) (note 2) (note 3).

4) Between Representations with the same full-screen resolution but different luminance resolutions as defined in table 9 “Table 9: Resolutions for Full-screen Display from 25 Hz H.264/AVC SDTV IRD and supported by 25 Hz H.264/AVC HDTV IRD, 50 Hz H.264/AVC HDTV IRD, 25 Hz SVC HDTV IRD and 50 Hz SVC HDTV IRD” and table 12 “Resolutions for Full-screen Display from H.264/AVC HDTV IRD and SVC HDTV IRD” of TS 101 154 [8] (e.g. 1920x1080 and 1440x1080) (note 2).

NOTE 1: Transitions shall be seamless unless combined with other changes which do not have that requirement.

NOTE 2: Transitions may include repeated frames but shall otherwise be seamless.

NOTE 3: As defined in clause 10.2.1 of HbbTV [1], video shall be scaled, preserving the aspect ratio, such that all of the decoded video is visible within the area of the video/broadcast or AV Control object. Clause 5.5.3.1 of MPEG DASH [3] requires all Representations in an AdaptationSet to have the same picture aspect ratio. The resolution and pixel aspect ratio can change as long as the picture aspect ratio remains the same.

Some examples of transitions between Representations which terminals may support but which are not required to support include:

1) Between Representations where one is interlaced and the other is progressive

2) Between Representations which differ in framerate, e.g. 25 and 50 fps

Terminals should not make transitions between Representations that would cause noticeable disruption to the presentation of the media at the switch point unless the transition is necessary to prevent interruption to the media presentation due to lack of data.

B.4.2.2 Audio tracks

During playback of adaptively streamed content encoded using HE-AAC or E-AC3, terminals shall support transitions between audio Representations as follows:

1) Between Representations which differ by bit-rate (note 1).

NOTE 1: Transitions shall be seamless unless combined with other changes which do not have that requirement.

Some examples of transitions between Representations which terminals may support but which are not required to support include:

1) Between Representations where one is encoded with HE-AAC and the other is E-AC3.

2) Between Representations which differ in the number of audio channels.

3) Between Representations which differ in the sampling frequency.

B.4.3 Trick Mode Support

The terminal shall support a minimum basic implementation of trick mode (the detailed implementation is outside the scope of this document).

Terminals shall support at least two different speeds for fast forward. Terminals shall support at least two different speeds for fast rewind.

B.4.4 Buffering

The terminal should not buffer more than data equivalent to approximately 300 seconds of normal play in advance of the current play position.

The requirement in clause 10.2.3.2 of HbbTV [1] concerning persistent storage of streamed content shall also apply to content delivered as specified in this annex.

Annex C: Common Encryption for ISOBMFF

Support for MPEG common encryption as defined in CENC [4] is optional in the present document. If it is supported then the following requirements shall apply.

C.1 Key Management for On Demand Assets

The HbbTV ISOBMFF Live media file shall be encrypted using a single key for all Representations in all Periods, and a single KID. As a consequence, the same key is used for all Representations of an On Demand asset, independent of its duration.

NOTE: In cases where it is desired to use different keys for different Representations, this may be done using multiple MPDs. For example, in order to target multiple groups of users or multiple device classes.

C.2 Key Management for Live Content

The HbbTV ISOBMFF Live media file shall be encrypted using a single key for all Representations within a single Period.

NOTE Periods are typically used for separate programs in a live broadcast.

NOTE: In cases where it is desired to use different keys for different Representations, this may be done using multiple MPDs. For example, in order to target multiple groups of users or multiple device classes.

The KID may be updated but not faster than every 120 seconds.

As a consequence, while the same key is used for all Representations of an On Demand asset, the key may be updated on a regular basis, hence reproducing a lower frequency key update mechanism than the one usually used to protect broadcast signals.

C.3 Encryption mode

Media data shall be encrypted using AES 128-bit in CTR mode (AES-CTR) as defined in section 9 of CENC [4]. Video tracks shall be encrypted as defined in section 9.6.2.

The terminal may implement the Full Encryption described in CENC [4].

C.4 Usage of ISOBMFF boxes

This clause specifies relevant parameters of existing ISOBMFF boxes used with CENC [4].

C.4.1 ‘pssh’ box

An ISOBMFF file may contain multiple Protection System Specific Header ('pssh') boxes (as defined in CENC [4]). The terminal shall be able to identify and use the 'pssh' box that corresponds to the DRM system that is available to the terminal. If the terminal has multiple DRM systems available with matching 'pssh' boxes, the terminal shall select between them to decrypt the content.

Usage of the ‘pssh’ by the DRM in either ‘moov’ or ‘moof’ box is optional. Normally, information in the MPD is sufficient for license acquisition by the terminal, but in live streaming situations, it may be necessary to distribute new protected keys/licenses in a ‘pssh’ box in each downloaded Track Fragment to allow encryption changes during a presentation (i.e. “key rotation”, multiple programs, interspersed advertisements, etc.).

If a DRM system uses the 'pssh' box, then the value of the SystemID field corresponding to that DRM system shall be specified as well as the encoding of the Data field.

C.5 Extensions to ISOBMFF boxes

C.5.1 Constraints on the SampleAuxiliaryInformationOffsetsBox

In order to ensure that the terminal has access to the sample auxiliary information before it is needed to decrypt a sample, the offsets in any 'saio' box must be such that they point to data that is after the start of, and before the end of, the 'moof' box that contains the 'traf' box that contains this 'saio' box.

For example, each ‘traf’ box of a track that may contain encrypted media samples may contain a Sample Encryption Information box (‘senc’) to provide the initialization vectors and subsample encryption information necessary to decrypt any encrypted media samples using the CENC [4] as defined in Section 7 of that document.

Box Type ‘senc’

Container Track Fragment Box (‘traf’)

Mandatory No

Quantity Zero or one

Syntax

```
aligned(8) class SampleEncryptionBox extends FullBox(‘senc’, version=0, flags=0) {
      unsigned int(32)  sample_count;
      {
            unsigned int(IV_size*8)  InitializationVector;
            if (flags & 0x000002)
            {
                  unsigned int(16)  subsample_count;
                  {
                        unsigned int(16)  BytesOfClearData;
                        unsigned int(32)  BytesOfEncryptedData;
                  } [ subsample_count ]
            }
      }[ sample_count ]
}
```

Annex D – DRM Integration (informative)

D.1 Introduction

This annex identifies issues which need to be considered and in most cases documented when defining how a DRM system is to be integrated with HbbTV. It is expected that solutions to these issues would form the basis of the document defining the technical integration between HbbTV and that DRM system and subsequently a test specification and test suite.

D.2 General issues

Some informative text is needed identifying how the key aspects of the DRM technology map on to the mechanisms and local interfaces showin in annex D of OIPF volume 5 [6].

A DRM System ID for the DRM system needs to be registered in as described in OIPF Volume 5 [6], Section 9.3.10.

If the DRM agent can generate user interfaces on the terminal then the interaction between these and the HbbTV system needs to be defined. This is particularly critical if these user interfaces are rendered using the same browser as is used for HbbTV applications. (See OIPF Volume 5 [6], section 5.1.1.6).

Which combinations of protocols and codecs are required to be supported with the DRM technology need to be defined. These must be in the format of the video profile capability strings indicating as defined in OIPF Volume 5 [6], section 9.3.11.

D.3 DRM Agent API

In the `sendDRMMessage` method (as defined in OIPF volume 5 [6], section 7.6.1.2), it needs to be defined which values of the `msgType` parameter are valid and what the contents of the `msg` parameter are for each message type.

In the `onDRMMessageResult` function (as defined in OIPF Volume 5 [6], section 7.6.1.1), the valid values for the `resultMsg` parameter should be defined if they are intended to be parsed by an HbbTV application. Additionally it needs to be defined which conditions in the DRM system trigger which `resultCode` values and any implications on the value of the `resultMsg`.

D.4 Content via the CEA-2014 A/V Object

If DRM is used to protect content presented via the CEA-2014 A/V object then the following need to be specified;

1) Whether the content access streaming descriptor is needed to provide information for the DRM system. If so then which of the fields are used, under what circumstances and what the requirements are on their contents need to be defined. If not then the mechanism by which DRM information is obtained needs to be defined.

2) Whether the DRM system can enforce parental access control and trigger an `onParentalRatingChange` event (as defined in OIPF volume 5 [6], section 7.14.5). If this event can be triggered then how the value of the `contentID parameter` is obtained needs to be specified. The same applies for `onParentalRatingError` event.

3) The conditions when the `onDRMRightsError` event is generated (as defined in OIPF Volume 5 [6], section 7.14.6). If it is generated, the values to be used for the `contentID` and the `rightsIssuerURL` parameters need to be defined.

Annex E – Code Examples (informative)

E.1 Access to EIT Schedule Information

```
////////////////////////////////////////////////////
//
//  Event listeners and functions to process the search results.
//  These functions are common to both examples
//
////////////////////////////////////////////////////


var mySearch;

// Listener for events relating to a metadata search
function searchUpdate(state, id) {

    if (state == 0) {
        // the search is complete

        // update the search results to make sure we have all results
        mySearch.result.update();
        processSearchResults();
    }
    else if (state == 1) {
        // more results are available

        // ignore this for now and wait for all data to be available
        // we could use this to incrementally update our UI if we wanted
    }
}


// Process the search results when the search is completed
function processSearchResults() {

    var results = mySearch.result;

    // Iterate over the results and do stuff
    var i;
    for (i=0; i < results.length; i++) {
        var name = results[i].title;
        var startTime = results[i].startTime;
        var duration = results[i].duration;

        // do stuff
    }
}



////////////////////////////////////////////////////
//
//  Example 1- Get programme schedule data for a set of channels
//
////////////////////////////////////////////////////

function getGuideData() {

    // Instantiate the search manager
    var mySearchMgr = window.oipfObjectFactory.createSearchManagerObject();
    mySearchMgr.addEventListener(“MetadataSearch”, searchUpdate);

    // Create the search object to search EPG data
    mySearch = mySearchMgr.createSearch(1);

    // Get the start time for the search.  We round down to the previous
// half-hour boundary
    var now = Math.round(new Date().getTime()/1000.0);
    var gridStart = (Math.floor(now/1800)) *1800)

    //Now use that to set the start and end times for the search
    var startQuery = mySearch.createQuery("startTime", 3, gridStart);

    // Assume the grid holds two hours worth of programmes
    var endQuery = mySearch.createQuery("startTime", 4, gridStart + 7200);

    // Now set that query
    mySearch.query = startQuery.and(endQuery);

    // Constrain the search to return only those programmes below the current
// parental rating threshold
    mySearch.addCurrentRatingConstraint();

    // Start the search, getting the first 50 results
    if (mySearch.result.getResults(0, 50)) {
        // All our data is available immediately
        processSearchResults();
    }
    else {
        // We need to wait...
    }
}



////////////////////////////////////////////////////
//
//  Example 2 - Get now/next data for a channel other
//  than the currently selected one.
```

```
//
////////////////////////////////////////////////////

function getNowNextData() {

    // Instantiate the search manager
    var mySearchMgr = window.oipfObjectFactory.createSearchManagerObject();
    mySearchMgr.addEventListener(“MetadataSearch”, searchUpdate);

    // Create the search object to search EPG data
    mySearch = mySearchMgr.createSearch(1);

    // Get the current time
    var now = Math.round(new Date().getTime()/1000.0);


    // Find data for the first channel in the channel list
    var chan = mySearchMgr.getChannelConfig().channelList[0];

    // Start the search, getting the first 2 results
    if (mySearch.findProgrammesFromStream(chan, now, 2)) {

        // All our data is available immediately, so we don’t need to call
// update() on the search results.
        processSearchResults();
    }
    else {
        // We need to wait...
    }
}
```