

ETSI TS 102 796 V1.2.1 (2012-11)



Hybrid Broadcast Broadband TV

EBU

OPERATING EUROVISION

Reference

RTS/JTC-023

Keywords

broadcasting, DVB, HTML, internet, multimedia

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

http://portal.etsi.org/chaicor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2012.

© European Broadcasting Union 2012.

All rights reserved.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPP™ and **LTE™** are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Contents

Intellectual Property Rights	7
Foreword.....	7
1 Scope	8
2 References	9
2.1 Normative references	9
2.2 Informative references.....	10
3 Definitions and abbreviations.....	11
3.1 Definitions.....	11
3.2 Abbreviations	12
4 Overview	13
4.1 Applications	13
4.2 Architecture (informative).....	14
4.2.1 Introduction.....	14
4.2.2 System overview.....	14
4.2.3 Functional terminal components.....	15
4.3 Terminal capabilities and extensions.....	16
4.4 Specification overview	16
5 User experience (informative).....	18
5.1 Visual appearance of interactive applications	18
5.1.1 Balance of video and application	18
5.1.2 Service selection and event change.....	19
5.2 User input	20
5.3 Access to interactive applications	21
5.3.1 Overview of ways of access.....	21
5.3.2 Inaccessibility of applications.....	21
5.3.3 Starting broadcast-related autostart applications	22
5.3.3.1 Possible states of an autostart application	22
5.3.3.2 "Red Button" applications.....	22
5.3.4 Starting digital teletext applications.....	23
5.3.5 Starting broadcast-independent applications	24
5.4 Exiting and hiding broadcast-related applications.....	25
5.5 User interface issues.....	25
5.5.1 Advertising broadcast applications	25
5.5.2 Co-existence with CI and CI+ MMI	25
5.5.3 Encrypted channels.....	26
6 Service and application model	26
6.1 Application model	26
6.2 Application lifecycle	27
6.2.1 Introduction.....	27
6.2.2 Starting and stopping applications	27
6.2.2.1 Summary	27
6.2.2.2 Behaviour when selecting a broadcast service	28
6.2.2.3 Behaviour while a broadcast service is selected.....	30
6.2.2.4 Other general behaviour	31
6.2.2.5 Simultaneous broadcast/broadband application signalling.....	33
6.2.2.5.1 Priority.....	33
6.2.2.5.2 Not currently operational broadband connection.....	33
6.2.2.5.3 Currently operational broadband connection and error accessing initial page	33
6.2.2.6 Broadcast-independent applications.....	33
6.2.2.7 Suspension of access to broadcast resources.....	34
6.2.2.8 Behaviour on encrypted broadcast services	35
6.2.3 Application lifecycle example (informative)	35
6.3 Application boundary.....	36

7	Formats and protocols	37
7.1	General formats and protocols.....	37
7.1.1	Graphic formats	37
7.1.2	Audio description.....	37
7.2	Broadcast-specific format and protocols	38
7.2.1	System, video, audio and subtitle formats	38
7.2.2	Protocol for application transport	38
7.2.3	Signalling of applications	38
7.2.3.1	Broadcast signalling	38
7.2.3.2	Broadcast-independent application signalling.....	40
7.2.4	Synchronization	41
7.2.5	DSM-CC carousel.....	42
7.2.5.1	Mounting related constraints.....	42
7.2.5.2	Initial carousel mounting.....	42
7.2.5.3	Subsequent carousel mountings (during the lifecycle of an application)	42
7.2.5.4	Constraints	42
7.2.6	Data Services	42
7.3	Broadband-specific format and protocols.....	43
7.3.1	System, video and audio formats	43
7.3.1.1	General requirements	43
7.3.1.2	Systems layers.....	44
7.3.1.3	Video.....	44
7.3.1.4	Audio.....	44
7.3.2	Protocols	44
7.3.2.1	Protocols for streaming	44
7.3.2.2	Protocols for download	45
7.3.2.3	Protocols for application transport	45
7.3.2.4	HTTP User-Agent header	45
7.3.2.5	HTTP Redirects.....	45
8	Browser application environment	46
8.1	DAE Specification Usage.....	46
8.2	Defined JavaScript APIs	46
8.2.1	Acquisition of DSM-CC stream events	46
8.2.1.1	Adding and removing stream event listeners	46
8.2.1.2	DSM-CC StreamEvent event	47
8.2.2	Carousel objects access with XMLHttpRequest.....	47
9	System integration.....	48
9.1	Mapping from APIs to protocols.....	48
9.1.1	Unicast streaming	48
9.1.1.1	General streaming requirements.....	48
9.1.1.2	HTTP streaming	48
9.1.2	Unicast content download.....	48
9.2	URLs	49
9.3	Other file formats	50
9.3.1	Stream event	50
9.4	Presentation of adaptive bitrate content.....	50
10	Capabilities.....	51
10.1	Display model	51
10.2	Terminal capabilities and functions.....	51
10.2.1	Minimum terminal capabilities	51
10.2.2	User input	54
10.2.3	Terminal functions.....	55
10.2.3.1	Favourites and bookmarks	55
10.2.3.2	Streaming and Download	55
10.2.3.3	PVR.....	55
10.2.4	Hybrid Broadcast Broadband TV reported capabilities and option strings.....	55
10.2.5	Terminal memory requirements.....	56
10.2.6	Parental Access Control.....	57
10.2.6.1	Broadcast channel	57
10.2.6.2	Streaming on-demand content.....	57

10.2.6.3	Downloaded content.....	57
10.2.6.4	PVR.....	57
10.2.7	Subtitles	58
11	Security.....	58
11.1	Application and service security	58
11.2	TLS and SSL Root Certificates	59
11.3	TLS client certificates (informative)	59
11.4	CI+.....	60
11.4.1	CI+ Communication	60
11.5	Protected content via Broadband.....	60
Annex A (normative): OIPF DAE Specification Profile.....		61
A.1	Detailed section by section definition	61
A.2	Modifications, extensions and clarifications	71
A.2.1	Resource management.....	71
A.2.2	Extensions to the ApplicationPrivateData class	72
A.2.3	Extensions to the oipfCapabilities embedded object.....	72
A.2.4	Extensions to the video/broadcast object.....	72
A.2.4.1	State machine and related changes	72
A.2.4.2	Access to the video/broadcast object	72
A.2.4.3	Extensions to the Configuration class for time-shift.....	73
A.2.5	Extensions to the AV Control Object.....	73
A.2.6	XHTML Profile.....	74
A.2.6.1	General.....	74
A.2.6.2	MIME type and DOCTYPE.....	74
A.2.6.3	Use of iframe Elements.....	74
A.2.6.4	Browser History.....	75
A.2.7	CSS profile	75
A.2.8	DOM profile.....	75
A.2.8.1	The Window object.....	75
Annex B (normative): Support for protected content delivered via broadband.....		76
B.1	Introduction	76
B.2	Common Encryption for ISOBMFF.....	76
B.2.1	Key Management for On Demand Content.....	76
B.2.2	Key Management for Live Content	76
B.2.3	Encryption mode	76
B.2.4	Usage of ISOBMFF boxes	77
B.2.4.1	'pssh' box.....	77
B.2.5	Extensions to ISOBMFF boxes.....	77
B.2.5.1	Constraints on the SampleAuxiliaryInformationOffsetsBox	77
Annex C (informative): Support for analogue broadcasting networks		78
C.1	Scope.....	78
C.2	AIT retrieval and monitoring	78
C.3	Tuning to a new channel	78
C.4	Other aspects	79
Annex D (informative): Server root certificate selection policy		80
D.1	Introduction	80
D.2	Background	80
D.3	Policy.....	80
Annex E (normative): Profiles of MPEG DASH.....		82

E.1	Introduction (informative).....	82
E.2	Requirements relating to the MPD.....	82
E.2.1	Profile definition	82
E.2.2	Numerical requirements	82
E.2.3	Metadata Requirements.....	83
E.2.4	Role Related Requirements	83
E.2.5	Audio Channel Configuration Requirements	84
E.2.6	Content protection signalling	84
E.3	Restrictions on Content	84
E.3.1	Restrictions on File Format	84
E.3.1.1	ISO Base Media File Format	84
E.3.2	Restrictions on Adaptation Sets.....	85
E.4	Requirements on Terminals.....	85
E.4.1	DASH Profile Support.....	85
E.4.2	Transitions between Representations	85
E.4.2.1	Video Tracks.....	85
E.4.2.2	Audio tracks.....	86
E.4.3	Buffering	86
E.4.4	ISO File Format Support	86
Annex F (informative): DRM Integration.....		87
F.1	Introduction	87
F.2	General issues.....	87
F.3	DRM Agent API.....	87
F.4	Content via the CEA-2014 A/V Object.....	87
History		88

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://ipr.etsi.org>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECTrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel: +41 22 717 21 11
Fax: +41 22 717 24 81

1 Scope

The present document defines a platform for signalling, transport, and presentation of enhanced and interactive applications designed for running on hybrid terminals that include both a DVB compliant broadcast connection and a broadband connection to the internet.

The main uses of the broadcast connection are the following:

- Transmission of standard TV, radio and data services.
- Signalling of broadcast-related applications.
- Transport of broadcast-related applications and associated data.
- Synchronization of applications and TV/radio/data services.

The main uses of the broadband connection are the following:

- Carriage of both On Demand and Live content.
- Transport of broadcast-related and broadcast-independent applications and associated data.
- Exchange of information between applications and application servers.

Applications are presented by an HTML/JavaScript browser.

The platform has the following characteristics:

- It is open and is not based on a single controlling authority or aggregator.
- Services and content from many different and independent providers are accessible by the same terminal.
- Standard functions of the terminal are available to all applications. Sensitive functions of the terminal are only available to trusted applications.
- Services and content may be protected.
- Broadcasted applications can be presented on terminals which are not connected to broadband. This includes both terminals which could be connected but have not yet been connected and terminals located where no broadband connectivity is available.
- Applications or services provided by a device manufacturer are outside the scope of the present document even if they use the same browser and features as described by the present document.
- Video, audio and system formats for the broadcast channel are outside the scope of the present document. Protocols for the broadcast channel are also outside the scope of the present document except for those relating to interactive applications.
- Applications can run on different types of terminals such as IDTVs, set-top boxes, and PVRs.
- Both broadcast-related and broadcast-independent applications are supported.

The platform combines a profile of the Open IPTV Forum specifications with a profile of the DVB specification for signalling and carriage of interactive applications and services in Hybrid Broadcast/Broadband environments. In addition, the present document defines supported media formats, minimum terminal capabilities, and the application life cycle.

The present document is intended to be usable without additional country/market-specific specifications. It is however also possible to combine it with country/market-specific specifications.

Some material contained herein is the copyright of, or has been supplied by the Digital TV Group.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

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

[1] Open IPTV Forum Release 1 specification, volume 5 (V1.2): "Declarative Application Environment".

NOTE: Available at <http://www.oipf.tv/specifications>.

[2] Open IPTV Forum Release 1 specification, volume 2 (V1.2): "Media Formats".

NOTE: Available at <http://www.oipf.tv/specifications>.

[3] ETSI TS 102 809 (V1.1.1): "Digital Video Broadcasting (DVB); Signalling and carriage of interactive applications and services in Hybrid Broadcast/Broadband environments".

[4] Open IPTV Forum Release 1 specification, volume 4 (V1.2): "Protocols".

NOTE: Available at <http://www.oipf.tv/specifications>.

[5] Open IPTV Forum Release 1 specification, volume 7 (V1.2): "Authentication, Content Protection and Service Protection".

NOTE: Available at <http://www.oipf.tv/specifications>.

[6] IETF RFC 2616: "Hypertext transport protocol - HTTP 1.1".

[7] IETF RFC 2818: "HTTP Over TLS".

[8] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".

[9] IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

[10] ETSI TS 102 851: "Digital Video Broadcasting (DVB); Uniform Resource Identifiers (URI) for DVB Systems".

[11] W3C Working Draft 19 November 2009: "XMLHttpRequest".

NOTE: Available at <http://www.w3.org/TR/2009/WD-XMLHttpRequest-20091119/>.

[12] CI Plus Forum, CI Plus Specification: "Content Security Extensions to the Common Interface", V1.2 (2009-04).

NOTE: Available at http://www.ci-plus.com/data/ci_plus_specification_v1.2.pdf.

[13] ISO/IEC 14496-3 (2009): "Information technology -- Coding of audio-visual objects -- Part 3: Audio".

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

[15] ETSI TS 102 366 (V1.2.1): "Digital Audio Compression (AC-3, Enhanced AC-3) Standard".

- [16] ETSI EN 300 468: "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".
- [17] Void.
- [18] Open IPTV Forum Release 1 specification, volume 3 (V1.2): "Content Metadata".
- NOTE: Available at <http://www.oipf.tv/specifications>.
- [19] ETSI TS 101 162 (V1.2.1): "Digital Video Broadcasting (DVB); Allocation of Service Information (SI) and Data Broadcasting Codes for Digital Video Broadcasting (DVB) systems".
- [20] IETF RFC 2246: "The Transport Layer Security (TLS) Protocol Version 1.0".
- [21] IETF RFC 4346: "The Transport Layer Security (TLS) Protocol Version 1.1".
- [22] Void.
- [23] W3C, XML Schema Part 2: "Datatypes Second Edition".
- NOTE: Available at Available at <http://www.w3.org/TR/xmlschema-2/>.
- [24] IETF RFC 6265: "HTTP State Management Mechanism".
- [25] IETF RFC 6454: "The Web Origin Concept".
- [26] IEC 62481-2 (2007-08): "Digital living network alliance (DLNA) home networked device interoperability guidelines - Part 2: Media Formats, ed1.0".
- [27] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".
- [28] W3C Recommendation (July 2002): "Exclusive XML Canonicalization - Version 1.0".
- NOTE: Available at <http://www.w3.org/TR/xml-exc-c14n/>.
- [29] ISO/IEC 23009-1 (2012): "Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats".
- [30] ISO/IEC 23001-7 (2012): "Information technology -- MPEG systems technologies -- Part 7: Common encryption in ISO base media file format files".
- [31] ISO/IEC 14496-12: "Information technology -- coding of audio-visual objects -- Part 12: ISO Base File Format".
- [32] Void.
- [33] Void.
- [34] ETSI TS 102 822-3-1 (V1.7.1): "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".
- [35] Void.

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

- [i.1] CEA-2014 revision A: "Web-based Protocol and Framework for Remote User Interface on UPnP™ Networks and the Internet (Web4CE)".
- [i.2] ETSI ES 202 130 (V2.1.2): "Human Factors (HF); User Interfaces; Character repertoires, orderings and assignments to the 12-key telephone keypad (for European languages and other languages used in Europe)".

- [i.3] ETSI TS 101 231 (V1.3.1): "Television systems; Register of Country and Network Identification (CNI), Video Programming System (VPS) codes and Application codes for Teletext based systems".
- [i.4] W3C: "How to Add a Favicon to your Site".
- NOTE: Available at <http://www.w3.org/2005/10/howto-favicon>.
- [i.5] Open IPTV Forum Release 2 Specification, Volume 5 (V.2.1): "Declarative Application Environment".
- NOTE: Available at <http://www.oipf.tv/downloads.html>.
- [i.6] HbbTV Specification (V 1.5), 1st August 2012.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

application data: set of files comprising an application, including HTML, JavaScript, CSS and non-streamed multimedia files

broadband: always-on bi-directional IP connection with sufficient bandwidth for streaming or downloading A/V content

broadcast: classical uni-directional MPEG-2 transport stream based broadcast such as DVB-T, DVB-S or DVB-C

broadcast-independent application: interactive application not related to any broadcast channel or other broadcast data

broadcast-related application: interactive application associated with a broadcast television, radio or data channel, or content within such a channel

broadcast-related autostart application: broadcast-related application intended to be offered to the end user immediately after changing to the channel or after it is newly signalled on the current channel

NOTE: These applications are often referred to as "red button" applications in the industry, regardless of how they are actually started by the end user.

digital teletext application: broadcast-related application which is intended to replace classical analogue teletext services

hybrid broadcast broadband TV application: application conformant to the present document that is intended to be presented on a terminal conformant with the present document

hybrid terminal: terminal supporting delivery of A/V content both via broadband and via broadcast

linear A/V content: broadcast A/V content intended to be viewed in real time by the user

non-linear A/V content: A/V content that which does not have to be consumed linearly from beginning to end for example, A/V content streaming on demand

persistent download: non-real time downloading of an entire content item to the terminal for later playback

NOTE: Persistent download and streaming are different even where both use the same protocol - HTTP. See clause 10.2.3.2.

progressive download: variant of persistent download where playback of the content item can start before the download of the content item has completed

NOTE: Progressive download is referred to as playable download in the OIPF DAE specification [1].

terminal specific applications: applications provided by the terminal manufacturer, for example device navigation, set-up or an internet TV portal

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

A/V	Audio Video
AD	Audio Description
AES	Advanced Encryption Standard
AIT	Application Information Table
AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
AVC	Advanced Video Coding
BFF	Base File Format
CAM	Conditional Access Module
CAS	Conditional Access System
CDN	Content Delivery Network
CEA	Consumer Electronics Association
CE-HTML	Consumer Electronics - Hypertext Markup Language
CENC	Common Encryption
CI	Common Interface
CICAM	Common Interface Conditional Access Module
CSP	Content and Service Protection
CSS	Cascading Style Sheets
CTR	Counter
DAE	Declarative Application Environment
DASH	Dynamic Adaptive Streaming over HTTP
DLNA	Digital Living Network Alliance
DOM	Document Object Model
DRM	Digital Rights Management
DSM-CC	Digital Storage Media - Command and Control
DTD	Document Type Definition
DVB	Digital Video Broadcasting
DVB-C	Digital Video Broadcasting - Cable
DVB-S	Digital Video Broadcasting - Satellite
DVB-SI	DVB Service Information
DVB-T	Digital Video Broadcasting - Terrestrial
EIT p/f	EIT present/following
EIT	Event Information Table
EPG	Electronic Program Guide
FQDN	Fully Qualified Domain Name
GIF	Graphics Interchange Format
HEAAC	High Efficiency AAC
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol - Secure
IDTV	Integrated Digital TV
IP	Internet Protocol
ISO	International Organization for Standardization
ISOBMFF	ISO Base Media File Format
JPEG	Joint Photographic Experts Group
KID	Key Identifier
LFE	Low Frequency Effect
MMI	Man Machine Interface
MPD	Media Presentation Description
MPEG	Motion Picture Experts Group
MSB	Most Significant Bit
OIPF	Open IPTV Forum
OITF	Open IPTV Terminal Function
PID	Packet Identifier

PMT	Program Map Table
PNG	Portable Network Graphics
PVR	Personal Video Recorder
RCU	Remote Control Unit
SD&S	Service Discovery and Selection
SDT	Service Description Table
SSL	Secure Sockets Layer
SVG	Scalable Vector Graphics
TLS	Transport Layer Security
TV	Television
UI	User Interface
UPnP	Universal Plug and Play
URL	Uniform Resource Locator
UTF-8	UCS Transformation Format—8-bit
XHTML	Extensible HyperText Markup Language
XML	eXtensible Markup Language

4 Overview

4.1 Applications

The web-based Hybrid Broadcast Broadband terminal as defined in the present document provides download and execution of applications which are defined as a collection of documents constituting a self-contained enhanced or interactive service. The documents of an application are HTML, JavaScript, CSS, XML and multimedia files.

The system architecture which allows for the provision of applications comprises a browser, application signalling via broadcast and broadband, application transport via broadcast and broadband, and synchronisation of applications and broadcast services (see clause 4.2 for details).

The present document addresses the following types of application:

- Broadcast-independent application (i.e. not associated with any broadcast service). This type of application is downloaded via broadband and accesses all of its associated data via broadband.
- Broadcast-related application (i.e. associated with one or more broadcast services or one or more broadcast events within a service) that may be launched automatically ("autostart") or explicitly upon user request. This type of application may be downloaded via broadband or broadcast and may access its data via either method.

The following possible uses of the browser environment are outside the scope of the present document:

- Service provider related applications as defined in the OIPF DAE specification [1].
- Using the browser environment to provide terminal specific applications such as a channel navigator or a device setup menu.
- Using the browser environment to display open Internet websites.
- Using the browser environment to support other specifications such as CEA-2014 [i.1] or the full set of Open IPTV Forum specifications.

4.2 Architecture (informative)

4.2.1 Introduction

This clause gives an overview of the system architecture and explains the necessary functional components inside a hybrid terminal. The level of detail of this explanation is general and abstract. Details about the internal structure of the components (e.g. whether the DSM-CC client has an integrated cache or not) or about their practical implementation (e.g. whether a specific component is solved in hardware or software) are omitted. Also in practice several components could be combined in one component (e.g. a browser with an integrated application manager). The primary intention of this clause is to provide an introduction and an understanding of the overall concept and the needed components. The communication between these components is outside the scope of the present document.

4.2.2 System overview

A hybrid terminal has the capability to be connected to two networks in parallel. On the one side it can be connected to a broadcast DVB network (e.g. DVB-T, DVB-S or DVB-C). Via this broadcast connection the hybrid terminal can receive standard broadcast A/V (i.e. linear A/V content), application data and application signalling information. Even if the terminal is not connected to broadband, its connection to the broadcast network allows it to receive broadcast-related applications. In addition, signalling of stream events to an application is possible via the broadcast network.

In addition the hybrid terminal can be connected to the Internet via a broadband interface. This allows bi-directional communication with the application provider. Over this interface the terminal can receive application data and non-linear A/V content (e.g. A/V content streaming on demand). The hybrid terminal may also support non-real time download of A/V content over this interface.

Figure 1 depicts the system overview with a hybrid terminal with DVB-S as the example of the broadcast connection.

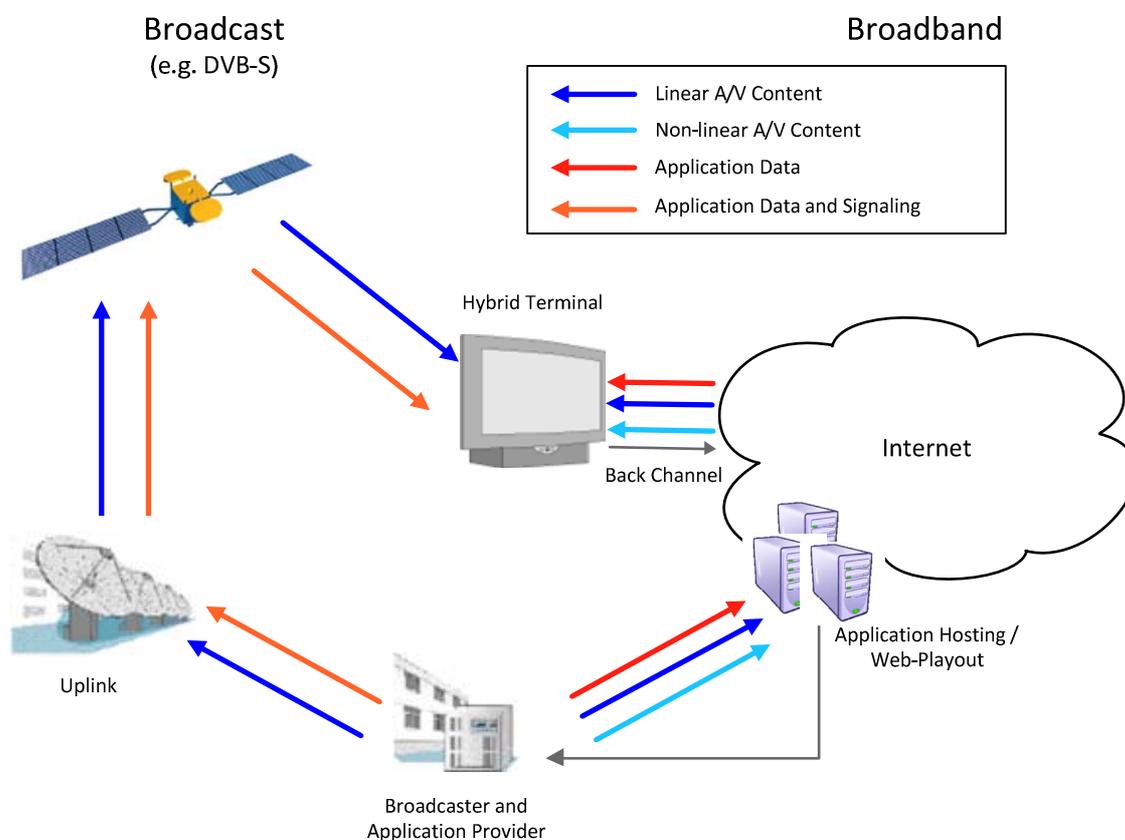


Figure 1: System Overview

4.2.3 Functional terminal components

Figure 2 depicts an overview of the relevant functional components inside of a hybrid terminal. These components are described below the figure.

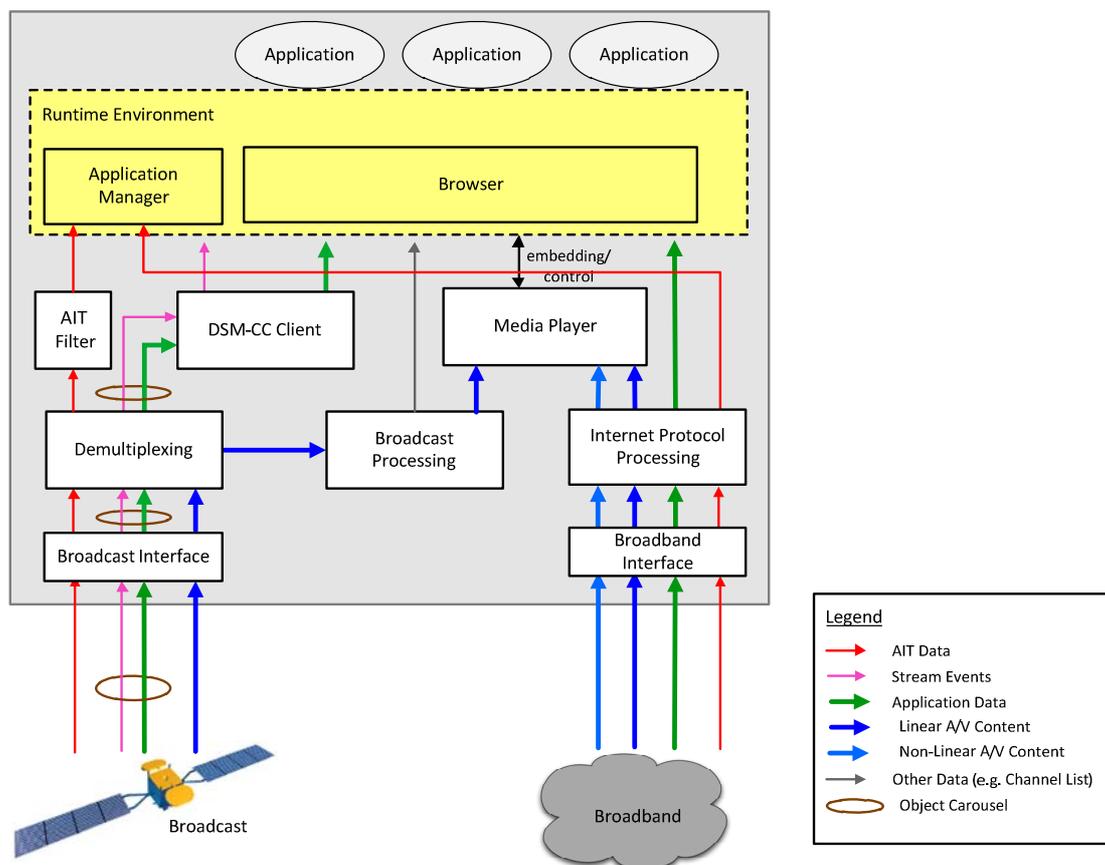


Figure 2: Functional components of a hybrid terminal

Via the **Broadcast Interface** the terminal receives AIT data, linear A/V content, application data and stream events. The last two data streams are transferred by using a DSM-CC object carousel. Therefore a **DSM-CC Client** is needed to recover the data from the object carousel and provide them to the Runtime Environment. The **Runtime Environment** can be seen as a very abstract component where the interactive application is presented and executed. The **Browser** and an **Application Manager** form this Runtime Environment. The **Application Manager** evaluates the AIT to control the lifecycle for an interactive application. The **Browser** is responsible for presenting and executing an interactive application.

Linear A/V content is processed in the same way as on a standard non-hybrid DVB terminal. This is included in the functional component named **Broadcast Processing** which includes all DVB functionalities provided on a common non-hybrid DVB terminal. Additionally some information and functions from the Broadcast Processing component can be accessed by the Runtime Environment (e.g. channel list information, EIT p/f, functions for tuning). These are included in the "other data" in figure 2. Moreover an application can scale and embed linear A/V content in the user interface provided by an application. These functionalities are provided by the **Media Player**. In figure 2 this includes all functionalities related to processing A/V content.

Via the **Broadband Interface** the hybrid terminal has a connection to the Internet. This connection provides a second way to request application data from the servers of an application provider. Also this connection is used to receive A/V content (e.g. for Content on Demand applications). The component **Internet Protocol Processing** comprises all the functionalities provided by the terminal to handle data coming from the Internet. Through this component application data is provided to the Runtime Environment. A/V content is forwarded to the **Media Player** which in turn can be controlled by the Runtime Environment and hence can be embedded into the user interface provided by an application.

4.3 Terminal capabilities and extensions

The present document defines a base level (or set of capabilities for terminals) which shall be supported in all terminals. This base level supports interactive applications:

- Which do not use video as part of their UI.
- Which use broadcast video as part of their UI.
- Which use unicast streaming content on demand as part of their UI.

In addition to this base level, the present document includes three other features which may optionally be supported by terminals:

- Support for downloading A/V content into persistent memory available locally to the terminal (both persistent download and progressive download) - this is referred to as the "download feature".
- Support for scheduling and playback of recordings and timeshifting of broadcast content using mass storage available locally to the terminal - this is referred to as the "PVR feature".
- Support for protected content via broadband is defined in annex B.

Additionally the present document defines some aspects that are mandatory for terminals supporting CI+ [12].

4.4 Specification overview

The present document specifies the technical requirements for the system described in the previous clauses. It largely references parts of already available standards and specifications and adapts these parts where necessary. The most significant referenced documents are the following:

- CEA-2014 [i.1] - Web-based Protocol and Framework for Remote User Interface on UPnP Networks and the Internet (Web4CE), also known as CE-HTML.
- Open IPTV Forum Release 1 Volume 5 [1] - Declarative Application Environment of the Open IPTV Forum.
- TS 102 809 [3] (formerly DVB Blue Book A137): "Signalling and carriage of interactive applications and services in Hybrid Broadcast Broadband environments".
- MPEG DASH - formally known as ISO/IEC 23009-1 [29]: Information technology -Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats.
- MPEG CENC - formally known as ISO/IEC 23001-7 [30]: Information technology -- MPEG systems technologies -- Part 7: Common encryption in ISO base media file format files.

The present document is the second revision of TS 102 796. It is based on the first revision of TS 102 796 with the addition of several features (DASH, CENC, etc) that have been previously published in a separate document called "HbbTV 1.5" [i.6]. It also includes errata to the original specification that have been found after the first release of the present document.

Figure 3 shows a graphical overview of the relationship between the profile defined here and the above mentioned specifications.

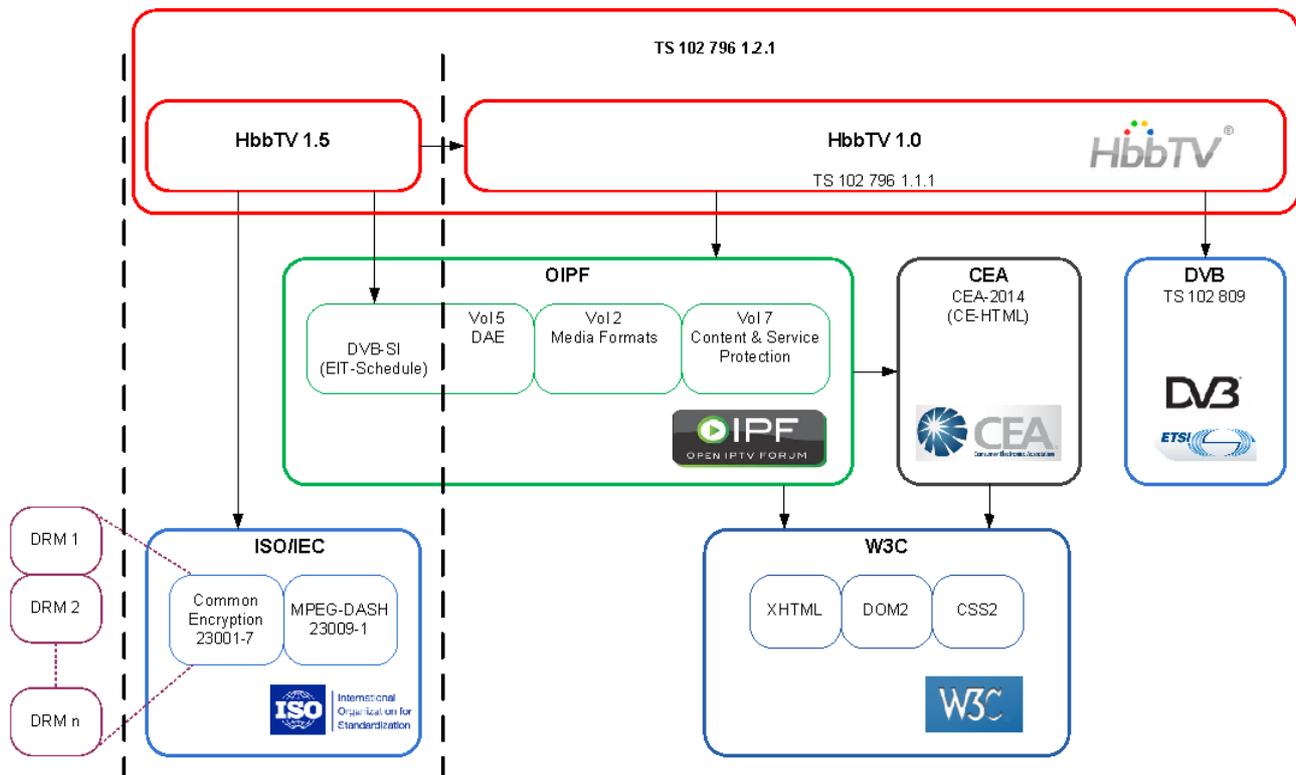


Figure 3: Specification overview

Important components provided by CEA-2014 [i.1] are:

- Definition of the application language (XHTML, CSS and JavaScript including AJAX).
- Definition of embedding non-linear A/V Content in an application.
- Definition of DOM event-handling (e.g. key events).
- Specification of still image formats.

CEA-2014 [i.1] is already profiled through the OIPF DAE specification [1]. The present document includes some additional profiling of that CE-HTML profile. Other important components provided by the OIPF DAE specification [1] are:

- JavaScript APIs for applications running in a TV environment (e.g. channel change).
- Definition of embedding linear A/V content in an application.

TS 102 809 [3] provides the following components:

- Application signalling.
- Application transport via broadcast or HTTP.

The audio and video formats are defined in the OIPF Media Formats specification [2].

In some rare cases none of the referenced standards provide an appropriate solution. In these cases the requirements are directly defined in the present document (e.g. the application lifecycle definition). Additionally the present document provides recommendations on the user experience and a description of the system overview.

The requirements in the OIPF and DVB specifications are only included if explicitly referenced in the present document or a dependency of those explicitly referenced parts. Other parts of those specifications are not required by the present document and should not be implemented unless required by another specification. The only parts of CE-HTML which are included are those explicitly required by OIPF except for those features removed as defined in clause 8.1.

5 User experience (informative)

This clause describes the behaviour of the terminal as seen by the end-user. It should be considered as usability guidelines for implementing interactivity. However, the described behaviour usually results from the functionality coded into the broadcast application, rather than the terminal.

A homogenous user experience is important to enable a successful interactive platform. To ensure this, both the manufacturer and the application developer should respect the following framework and guidelines.

5.1 Visual appearance of interactive applications

5.1.1 Balance of video and application

Table 1 illustrates the range of different visual appearances the end user might experience. Each "screen" shows a different balance between "conventional TV" content and information delivered by an interactive application.

Table 1: Typical range of programme types perceived by end users

	1. Conventional TV
	2. TV with visual prompt of available information ("Red Button")
	3. TV with information overlaid (still picture only in the overlaid information, no A/V in overlay)
	4. Information with video, audio or picture inset
	5. Just information (without A/V)

5.1.2 Service selection and event change

The end-user may see a change in appearance either when she/he changes channel or when a service changes through time.

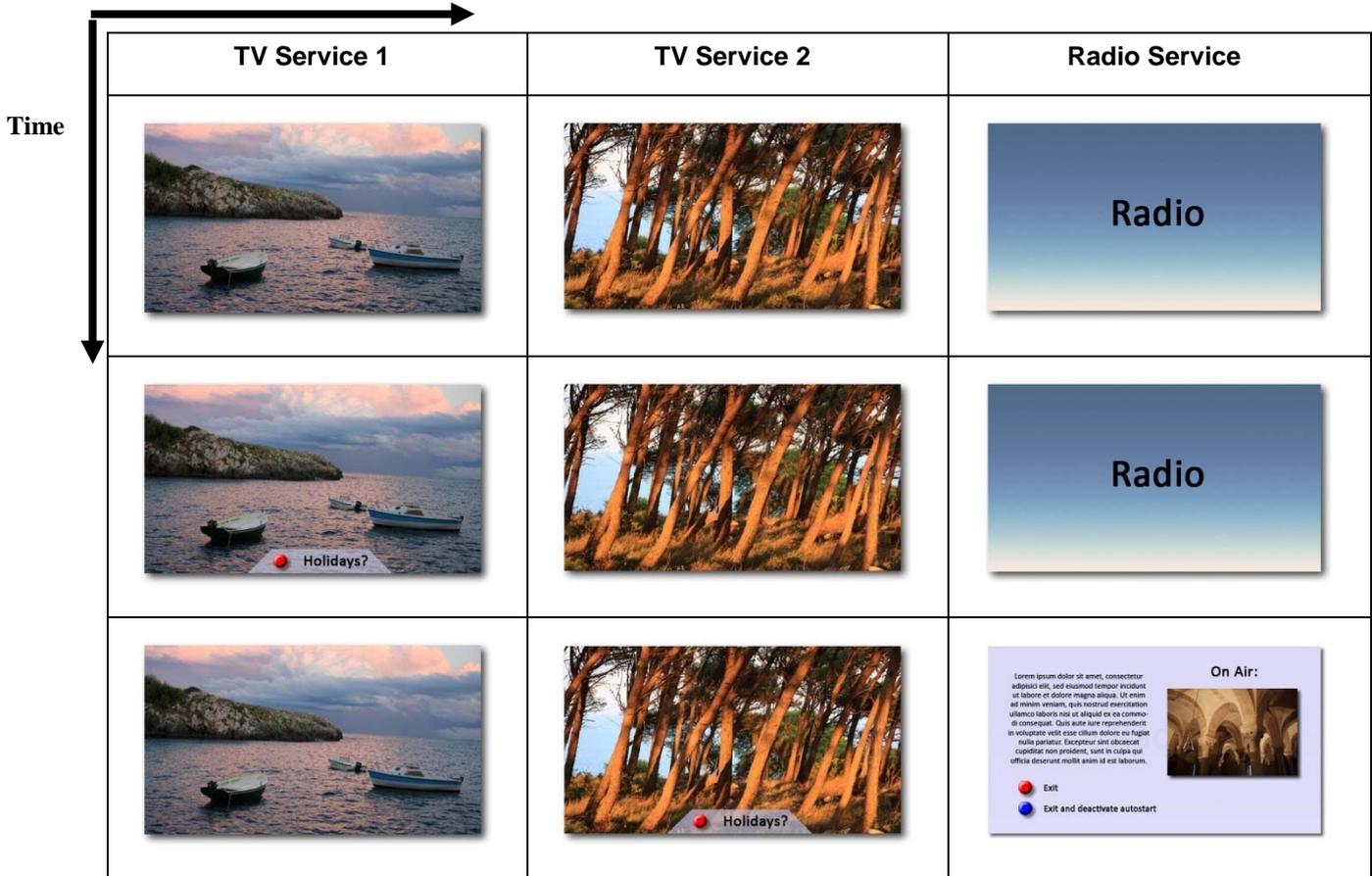


Figure 4: What might be seen across channels and through time

5.2 User input

The user controls interactive applications using a user input device typically supplied with the terminal. This may be a conventional remote control or an alternative input device such as a game controller, touch screen, wand or drastically reduced remote control.

NOTE: While the alternative input devices do not have buttons in the same way as a remote control, it is expected that implementations using these alternative input devices will include means to generate input to the application (called key events) logically equivalent to pressing buttons on a conventional remote control.

Table 2 lists the buttons or key events which are relevant for the end user when using interactive applications. Requirements on implementations are found in table 12 in clause 10.2.2.

Table 2: Relevant remote control buttons or key events for the end user when using interactive applications

Button or Key Event	Usage
TEXT or TXT or comparable button	Launches the digital teletext application and/or the standard teletext as described in clause 5.3.4
red colour button	Usually displays or hides a broadcast-related autostart application
3 additional colour buttons (green, yellow, blue)	Variable usage as defined by the application (typically short-cuts or colour-related functions)
4 arrow buttons (up, down, left, right)	Variable usage as defined by the application (typically focus movement or navigation through lists)
ENTER or OK button	Variable usage as defined by the application (typically selection of focused interaction elements or confirmation of requested actions)
BACK button	Variable usage as defined by the application (typically going back one step in the application flow)
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 6
WEBTV or comparable button	If available: opens a menu providing access to broadcast-independent applications as described in clause 5.3.5
EXIT or TV or comparable button	If available: terminates a running application and returns to last selected broadcast service

5.3 Access to interactive applications

5.3.1 Overview of ways of access

The end user can access interactive applications via the following ways:

- Accessing a typical broadcast-related autostart application by pressing the visually indicated "Red Button" (see clause 5.3.3.2).
- Starting a digital teletext application by pressing the TEXT button (see clause 5.3.4).
- Starting a broadcast-independent application through the Internet TV portal of the manufacturer if one is offered (see clause 5.3.5).
- Starting an application via a link in the currently running application.
- Selecting a broadcast channel which has a broadcast-related autostart application which starts in full-screen mode (usually only used on radio or data services).

5.3.2 Inaccessibility of applications

If a non-autostart application (e.g. a digital teletext application) is not available via the broadcast channel but only via broadband and the terminal is not connected to a broadband network, the terminal should display a suitable error message encouraging the end user to connect the device to one. Technical error messages (e.g. HTML status code 404) or a black screen should be avoided.

Despite the device having an active broadband connection, failure to access the initial page of an autostart broadband service should not cause any message (error or otherwise) to be displayed on the screen and disturb the TV watching experience.

5.3.3 Starting broadcast-related autostart applications

5.3.3.1 Possible states of an autostart application

Broadcast-related autostart applications are usually associated with a broadcast channel or an event (or part of an event) on that channel. In the first case, they start as soon as the channel is selected. In the second case, they start through an AIT update (usually co-incident with the start of the event).

Broadcast-related autostart applications may be in one of the following states when they start:

- 1) Displaying a "Red Button" notification to inform the user that the application is available.
- 2) Displaying no user interface.
- 3) Displaying their full user interface (usually only used on radio and data services).

In general, autostart applications on TV services should not display their full user interface (i.e. state 3) automatically. Instead, the user is informed of their availability by the "Red Button" icon (i.e. state 1). Further parts of the application should not be started unless the end-user presses the "Red Button".

Applications will start with a window covering the entire display in order that they can position the "Red Button" notification where they wish. Since the browser rendering canvas default color is device-dependent, applications should explicitly set the background of their `<body>` element to transparent using (for example) the following CSS rule:

```
body {
  background-color: transparent;
}
```

This ensures that the video for the current service is visible in those areas of the screen where the "Red Button" notification is not displayed.

On some services (e.g. radio), a broadcast-related autostart application may start by displaying its full user interface (i.e. state 3) immediately without displaying a "Red Button" icon beforehand.

When an application changes from state 1 or 3 to state 2, it should:

- Remove all graphics on screen.
- Stop presenting any kind of streaming audio or video.
- Restart the broadcast service (if it is a broadcast-related application and the broadcast service has been stopped).
- Rescale/reposition video to "full screen mode" (if video has been scaled/positioned).
- Unmute audio (if audio has been muted).
- Stop consuming any key events apart from the "Red button" (which should be used to change back to state 3).

When an application changes from state 2 to state 1 or 3, it should:

- Show new application graphics as appropriate.
- Inform the terminal which key events it wishes to consume in its new state.

For some use cases e.g. interactive radio applications, some of these may not apply.

5.3.3.2 "Red Button" applications

This type of broadcast-related autostart application indicates its availability by displaying a "Red Button" icon on the screen. This icon is displayed for a time period and then it may disappear. Pressing the "Red Button" on the RCU always displays the full user interface of the application (see figure 5), whether the "Red Button" icon currently being displayed or not. If there is no broadcast-related autostart application, pressing the "Red Button" has no effect (see figure 6).

NOTE: The "Red Button" icon is generated by the broadcast-related autostart application and therefore it is also designed by the owner of the application.

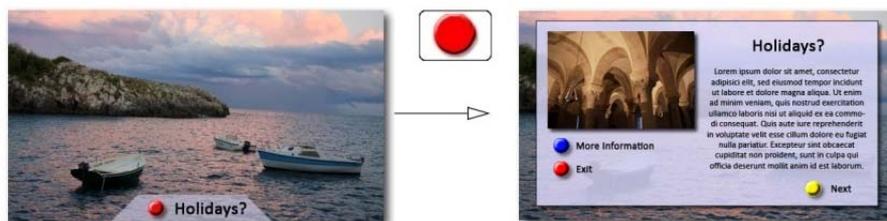


Figure 5: Service with associated broadcast-related autostart application



Figure 6: Service without associated broadcast-related autostart application

The end user may be able to control a setting to disable the initial drawing of the "Red Button" indication. If the end user selects this setting then this broadcast autostart application will display its full user interface when it starts, without drawing a "Red Button" indication. Support for this setting is provided entirely by the application. If such a setting is available, it should be easy for the end user to find and its purpose should be clear to the end user.

5.3.4 Starting digital teletext applications

A digital teletext application is a special broadcast-related application which is started by pressing the TEXT button on the RCU. Depending on the provision of a digital teletext application and of standard teletext the reaction on pressing the TEXT button differs.

Case A: If only the standard teletext is available on the current service, the standard teletext is displayed.

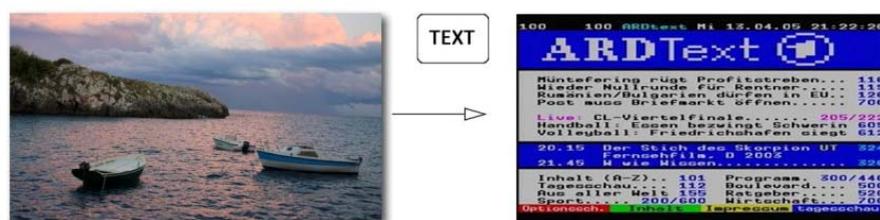


Figure 7: Service with standard teletext only

Case B: If only a digital teletext application is available on the current service, this application is started. Pressing the TEXT button a second time terminates the application and causes the AIT to be re-parsed and any autostart application to be restarted.

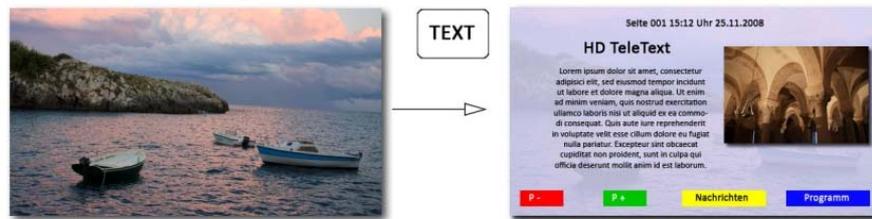


Figure 8: Service with digital teletext application only

Case C: If both a digital teletext application and standard teletext are available on the current service, an easy to use mechanism should be implemented to toggle between the different teletext modes.

EXAMPLE: Pressing the TEXT button for the first time could start the digital teletext application, pressing it for the second time would close the digital teletext application and start the standard teletext, and pressing it for the third time would close the standard teletext and rerun AIT parsing and start the autostart application if provided.



Figure 9: Example of service with digital teletext application & standard teletext

Case D: If a digital teletext application is signalled but not available (because the digital teletext application is only reachable via broadband and the terminal is not connected appropriately) but standard teletext is available, the standard teletext would be displayed (see also figure 7).

Case E: If no digital teletext application is signalled and standard teletext is not available, nothing should happen.



Figure 10: Service without associated teletext

Case F: If a digital teletext application is signalled but not available (because the digital teletext application is only reachable via broadband and the terminal is not connected appropriately) and standard teletext is not available, the terminal would display an informative message encouraging the end user to connect the terminal to the internet.

5.3.5 Starting broadcast-independent applications

Broadcast-independent applications are started via a running application or an Internet TV Portal. An Internet TV Portal is an application which provides a type of start page where broadcast-independent applications are sorted and offered in an appropriate and useful way to the end user. The Internet TV Portal may be opened by pressing a dedicated Internet TV Button on the RCU. The type of interactive applications that are listed in the Internet TV Portal is the responsibility of the manufacturer. There may be an option for the user to add broadcast independent applications via manual URL entry or similar means like apps on mobile phones. The structure and the design of the start page is the responsibility of the manufacturer and out of the scope of the present document. Broadcast-independent applications are described in more detail in clause 6.2.2.6.

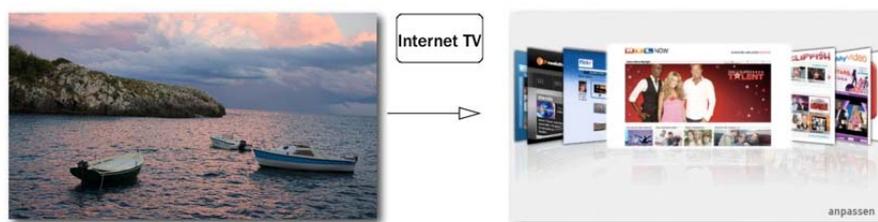


Figure 11: Internet TV Portal

5.4 Exiting and hiding broadcast-related applications

According to the technical definitions of the application lifecycle in clause 6, applications may be stopped when they launch other applications or a channel change is performed. Applications may also kill themselves, either as a result of a request by the end-user or as a consequence of some internal logic.

If the input device comprises an EXIT button or a comparable button, pressing this button terminates the application.

Applications may disappear from view automatically on some actions of the end-user which cause the application to move to state 2 (as defined in clause 5.3.3.1). "Red Button" applications should always provide this function and should use the "Red Button" to toggle between state 2 and state 3 (as defined in clause 5.3.3.1). Applications should use the `Application.hide()` method to hide their user interface, or may use an alternative approach.

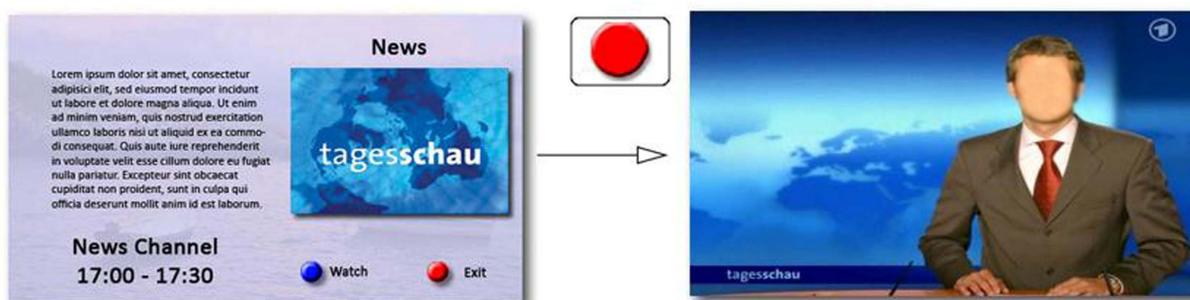


Figure 12: Application selects TV channel

5.5 User interface issues

5.5.1 Advertising broadcast applications

The user interface displayed on channel change (and when the "Info" button is pressed) is the responsibility of the terminal manufacturer but typically includes the title and synopsis of the current event. It is recommended that the presence of HbbTV applications signalled in the broadcast is indicated to the user in this UI.

5.5.2 Co-existence with CI and CI+ MMI

A CAM may request the terminal to display an MMI screen or dialogue at any time. The terminal has to respect the mandatory requirements of the CI and CI+ specifications (see sections 12.3.3 and 12.6.1.1 of CI+ [12]). Working within those constraints, the terminal should endeavour to present a consistent and uncomplicated user interface at all times. On occasion, this may result in the HbbTV application at least losing focus and possibly being terminated.

If any interaction between the CAM and the user is required, application authors are strongly recommended to use the `oipfDrmAgent` APIs to allow communication between the CAM and the HbbTV application, which can then act as a proxy for any interaction with the user.

5.5.3 Encrypted channels

Terminals may wish to display a message to the user that the channel is encrypted and cannot be displayed (see clause 6.2.2.8). If they do so, they should be aware that applications may wish to present some relevant information for this scenario. Hence any native UI should not remain on screen permanently or should give the user a way to remove it.

6 Service and application model

6.1 Application model

The present document defines a model which supports one Hybrid Broadcast Broadband TV application visible at one time.

Two types of applications are supported:

- Broadcast-related applications. These are signalled as part of a broadcast channel as defined in clause 7.2.3.1 and follow the lifecycle rules defined in clauses 6.2.2.2 and 6.2.2.3.
- Broadcast-independent applications. These are either not signalled at all or are signalled as in clause 7.2.3.2. They follow the lifecycle rules defined in clause 6.2.2.6.

Applications may transition between these two types as described later in the present document.

Terminal specific applications like navigators, channel list management, terminal specific EPGs or PVR control applications are out of scope of the present document.

It is optional for a terminal to support background preloading and rendering of applications other than the visible one.

No mechanism is defined to allow the visible application to interact with other running applications.

Terminal specific applications may be temporarily displayed on top of Hybrid Broadcast Broadband TV applications. This shall not affect the state of the Hybrid Broadcast Broadband TV application but during this time, if the terminal specific application takes focus, the Hybrid Broadcast Broadband TV application shall not receive any key event. Calls to `application.show()` while a terminal specific application is visible shall either:

- cause the Hybrid Broadcast Broadband TV application to be visible behind the terminal specific application;
or
- cause the Hybrid Broadcast Broadband TV application to become visible once the terminal specific application stops being visible assuming that the Hybrid Broadcast Broadband TV application is still running and that `application.hide()` has not been called.

6.2 Application lifecycle

6.2.1 Introduction

The application lifecycle is determined by the following four factors:

- 1) The application model.
- 2) The currently selected broadcast service (if any) and changes to it.
- 3) The applications signalled as part of the currently selected broadcast service.
- 4) The signalled application control code (as defined in clause 7.2.3.1 of the present document and clause 5.2.4 of TS 102 809 [3]).

6.2.2 Starting and stopping applications

6.2.2.1 Summary

Starting an application may be initiated in the following ways:

- Directly by the end-user (e.g. by using dedicated buttons on the remote control or an equivalent menu provided by the terminal).
- In response to signalling in a broadcast service (e.g. automatically starting a broadcast-related autostart application).
- By an already running application (via the JavaScript method `createApplication()`).

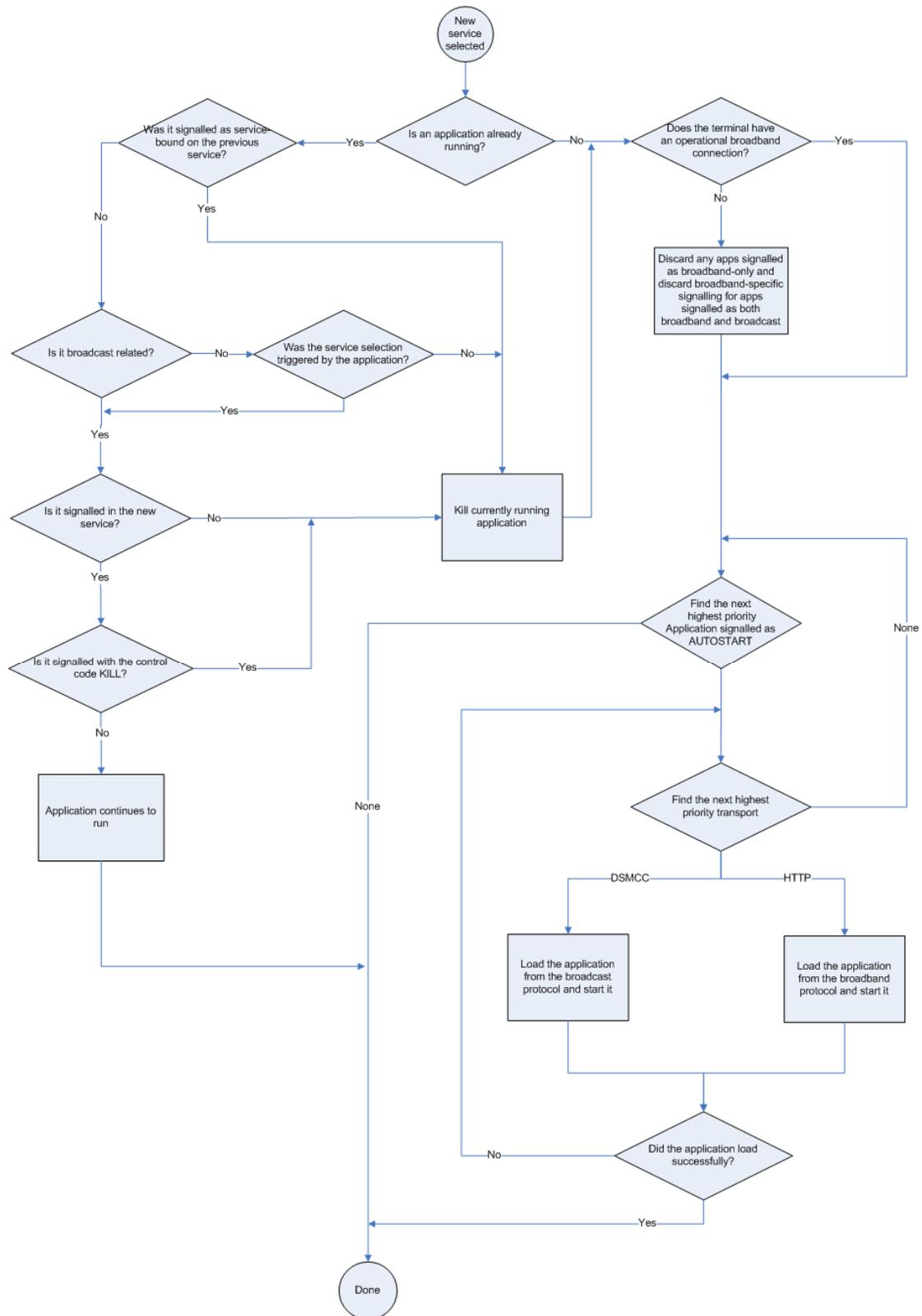
Starting applications in response to the playback of recorded or downloaded content is not supported.

An application may be stopped in the following ways:

- As defined in the flowcharts in clauses 6.2.2.2 and 6.2.2.3.
- By calling `Application.destroyApplication()`.
- By the terminal, under error conditions.
- Directly by the end-user.

6.2.2.2 Behaviour when selecting a broadcast service

Figure 13 shows the rules that shall apply when the selected broadcast service changes.



NOTE: It is strongly recommended that broadcasters only signal one autostart application per broadcast service.

Figure 13: Behaviour when selecting a broadcast service

Figure 13 shall not apply when selecting an MPEG program which is not a broadcast DVB service. If a transport stream does not include an SDT actual then none of the MPEG programs in that stream are broadcast DVB services. If the SDT actual in a transport stream does not include an entry corresponding to a PMT in that transport stream then the MPEG program described by that PMT is not a broadcast DVB service. There is no requirement for a terminal to check again either for an SDT or that a service is listed in the SDT if it has already done so, e.g. in order to acquire the service name when creating the channel list.

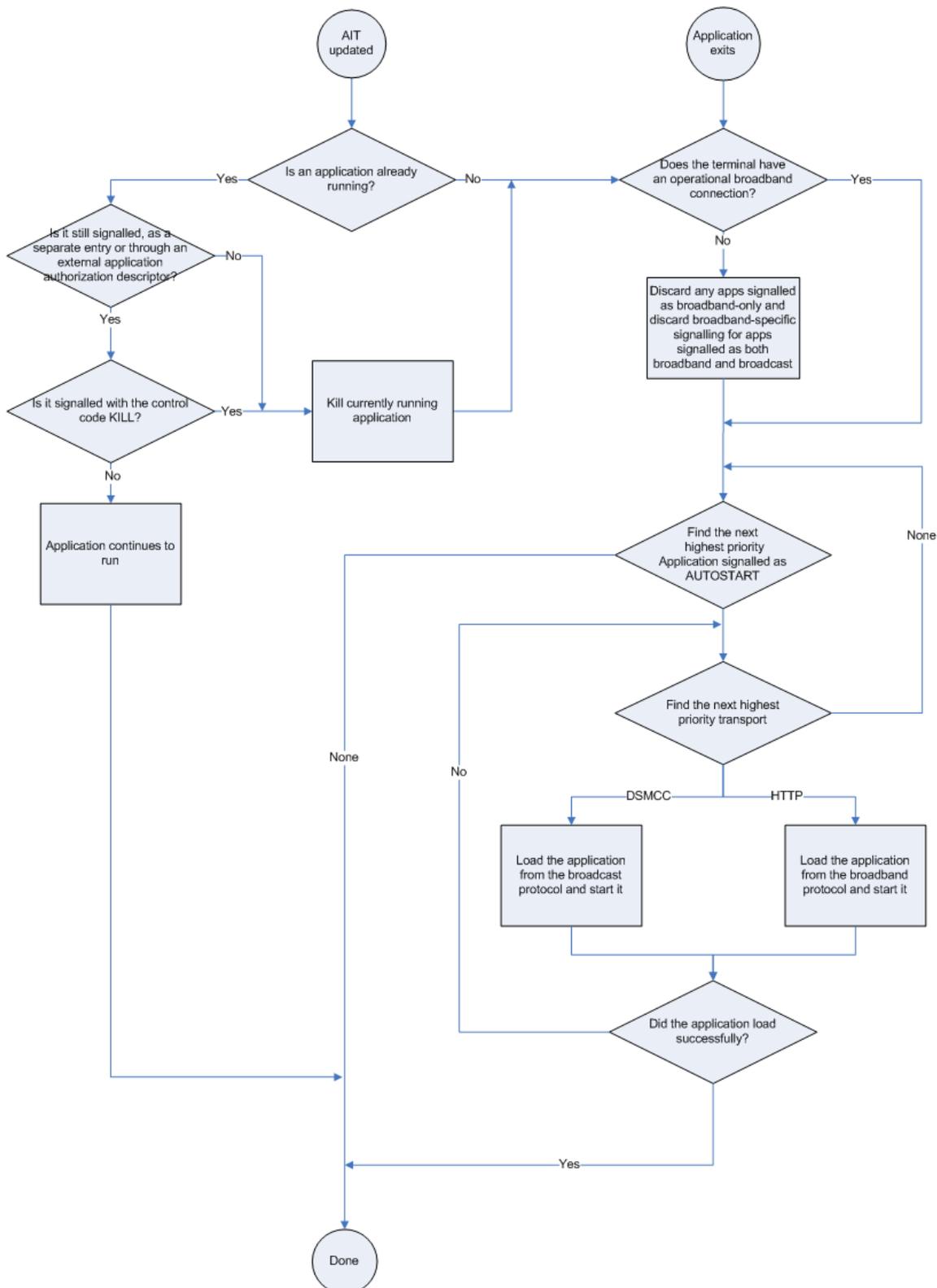
NOTE: If broadcasters or operators change programs in a multiplex from being a broadcast service to a non-broadcast service or vice-versa, they should use new program numbers/service_ids and should not re-use the old program numbers/service_ids.

As a consequence of selecting such an MPEG program:

- No applications shall be started.
- No applications shall be stopped except for broadcast-related applications with `service_bound_flag` set to '1' which are stopped when leaving the current broadcast service.
- The value of the `currentChannel` property on the video/broadcast object and the `ApplicationPrivateData.currentChannel` property shall reflect the MPEG program selected.
- Figure 13 shall not apply when selecting an MPEG program that is not a broadcast DVB service.

6.2.2.3 Behaviour while a broadcast service is selected

Figure 14 shows the rules that shall apply if the AIT changes or a broadcast-related application exits while a broadcast service is selected.



NOTE: By "operational broadband connection", it is meant that at the time of the operation, the connection to the Internet is functional.

Figure 14: Behaviour while a broadcast service is selected

In figure 14, the following clarifications shall apply:

- For the purposes of deciding whether an application is already running or is still signalled, only the `organization_id` and `application_id` fields from the AIT shall be used. Other information (e.g. the URL of the first page) shall not be used.
- Other than `organization_id` and `application_id`, the only other field in the AIT which is relevant when the AIT is updated is the application control code. Changes in other fields shall be ignored for already running applications.

NOTE 1: As a result of the above, changes to fields in the AIT other than `organization_id`, `application_id` and application control code will only take effect for newly started applications. In order for those changes to effect an already running application, the application needs to exit and re-start. It is up to the broadcaster and/or application provider to arrange for this to happen.

NOTE 2: A change in the version number of an AIT subtable is an indication to the terminal to retrieve a new version of the AIT. It does not imply or require any changes in the content of the AIT itself. For example, adding an application to the AIT would be an update to the AIT without changing the AIT entries for any existing applications.

If the only running broadcast-related application exits without starting a broadcast-independent application or without the terminal changing channel, the AIT shall be re-parsed and any autostart application shall be re-started following the rules defined in the previous clause. It may be that the restarted application is the same one as the one that just exited. If an application exits when an MPEG program that is not a broadcast DVB service is selected and that MPEG program does not include an AIT then the behaviour is implementation specific.

This flowchart shall not apply while MPEG programs are selected which are not a broadcast service, (i.e. not listed in the SDT of the transport stream carrying them or are carried in a transport stream that does not include an SDT) and which do not include an AIT.

Terminals shall include a mechanism to start and stop digital teletext applications, For example, the TEXT key on an RCU could be used to start the digital teletext application (which would require any other running application to be killed); pressing the TEXT key again causes the running application to be stopped as long as it is signalled as a digital teletext application. Digital teletext applications are identified with an `application_usage_descriptor` in the AIT with `usage_type` equal to 1.

NOTE 3: The digital teletext application is intended to be distinct from the autostart application(s) in the AIT. Care is needed if a teletext application is started by means other than the TEXT key.

6.2.2.4 Other general behaviour

Any application shall be stopped under the following circumstances:

- The application itself exits using the `Application.destroyApplication()` method (as defined in clause 7.2.2 of the OIPF DAE specification [1]).
- In response to changes in the application signalling as defined in clauses 6.2.2.2 and 6.2.2.3 for broadcast-related applications.
- The terminal has run out of resources for executing the application and therefore has to terminate it in order to keep operating correctly.

By default, newly started broadcast-related applications shall not be visible to the end user. These applications shall call the `Application.show()` method in order to display their user interface and accept user input. Newly started broadcast-independent applications shall be visible and active without needing to call this method.

Terminals may be configurable (either by the user or by the manufacturer) to not load or not start applications in spite of other requirements in the present document.

The requirements in the present document on starting and stopping Hybrid Broadcast Broadband TV applications may be modified for markets where other application formats are already deployed. For example, a static priority (one format always having priority over another where both are present) or a dynamic priority based on broadcast signalling may be used.

When one application requests a second application be started, the first application shall continue to run until the initial HTML document of the second application has been loaded - i.e. until after an `ApplicationLoadError` event would be generated (if any listener was registered). Only then shall the first application be stopped.

Failing to parse the initial page of an application shall be regarded as a loading failure when evaluating if the application successfully loads in figures 13 and 14.

If the terminal initiates time-shifting of the currently selected broadcast service, an application may get out of sync with the presentation of the audio-video components of this service. An HbbTV application shall be terminated if it is not safe to run it on a time-shifted broadcast service. An application is safe to run in time shift mode, if it is signaled in the AIT with an `applicaton_recording_descriptor` and both the `trick_mode_aware_flag` and the `time_shift_flag` set to '1' as described in clause 7.2.3.1. If an application is killed due to a broadcast service being time-shifted, the procedure defined in clause 6.2.2.2 for selecting an autostart application to run shall be followed except that only applications that are time-shift safe shall be considered.

After starting time-shift a terminal shall:

- Dispatch a `RecordingEvent` to signal a state change to state 11 "time-shift mode has started" of the PVR state machine
- Update the `recordingState`, `playPosition` and `playSpeed` properties of the video/broadcast object

After stopping time-shift a terminal shall dispatch a `RecordingEvent` to signal a state change to state 0 "unrealized" of the PVR state machine.

The present document defines two implementation options for support of applications when video is time-shifted - depending on whether the terminal can or cannot maintain synchronization between applications and the A/V components of a service. Which of these two options is implemented by a terminal is indicated by the `timeShiftSynchronized` property.

When a terminal can maintain synchronization between applications and the A/V components of a service, all of the following shall apply:

- DSMCC stream event descriptors shall be recorded with the A/V components keeping the timing relation and shall be delivered during playback of the time-shift
- The AIT shall be monitored, any changes shall take effect preserving the correct timing with respect to the A/V components
- The service information shall be recorded with the A/V components keeping the timing relation and the properties of the video broadcast object (e.g. programmes, `AVComponent` as defined in clause 7.13.4 of the OIPF DAE specification [1]) changes at the proper time of the playback of the time-shift
- The `timeShiftSynchronized` property shall be set to true (see clause A.2.4.3)

If a terminal is not able to maintain synchronization between applications and the A/V components of a service:

- The application may receive some (or all) broadcast resources from the live broadcast signal instead of the time shift playback
- It shall set the `timeShiftSynchronized` property to false

NOTE: When an application accesses service information or receives stream events, it may check if it is synchronized with the A/V component of the service by reading the values of the properties `recordingState` and `timeShiftSynchronized`.

When an application selects a new broadcast channel, there is a period of time between the channel change having been completed (when the `onChannelChangeSucceeded` event is triggered) and the AIT having been received and parsed. During this period, the application shall retain its type (broadcast-related or broadcast-independent) and trust level (trusted or untrusted). Hence, while a broadcast-independent application is transitioning to become broadcast-related, access to features limited to broadcast-related applications will continue to fail as they did before the transition started until the AIT has been received and parsed.

6.2.2.5 Simultaneous broadcast/broadband application signalling

6.2.2.5.1 Priority

Both broadcast and broadband transport protocols may be specified simultaneously for a given application. The priority by which the transport protocols shall be used is determined as specified in clause 5.3.5.3 of TS 102 809 [3].

6.2.2.5.2 Not currently operational broadband connection

Where a terminal does not have a currently operational broadband connection and an application to be launched is signalled to be:

- Available both through broadcast and broadband: the terminal shall disregard the signalling for the broadband transport protocol.
- Available only through broadband: the terminal shall ignore the request to launch the application (and return an error if the application was launched by a call to `createApplication()`).

6.2.2.5.3 Currently operational broadband connection and error accessing initial page

Where a terminal has a currently operational broadband connection but there is an error (asynchronous due to the nature of the HTTP protocol) accessing the initial page of a broadband application and an application to be launched is signalled as:

- Available through broadband as top priority and then through broadcast: the terminal shall revert to the broadcast version.
- Available only through broadband: the terminal shall not display an error message for applications which were either launched as autostart (e.g. following a channel selection or AIT update) or which were launched by another application.

If the application cannot ultimately be loaded from either broadcast or broadband and the application was launched by a call to `createApplication()`, an `ApplicationLoadError` shall be dispatched. Once the initial page of an application has been successfully loaded, the present document does not specify how terminals should behave if a page from that application subsequently fails to load.

6.2.2.6 Broadcast-independent applications

A broadcast-independent application can be created in one of the following ways:

- By calling the `Application.createApplication()` method with either an HTTP or an HTTPS URL. The URL shall refer to either an HTML page or an XML AIT (see clause 7.2.3.2).
- Optionally from a terminal specific application like an Internet TV Portal or following manual URL input as described in clause 5.3.5.

Where the URL refers to an HTML page directly, the broadcast-independent application shall be created without an `organization_id` or `application_id`.

Where the URL refers to an XML AIT, the broadcast-independent application shall be created with the `organization_id` and `application_id` specified in the XML AIT. In both cases, the application shall be associated with an application boundary as defined in clause 6.3.

When a broadcast-related application starts a broadcast-independent application, the application is started but the broadcast service shall cease to be selected - logically equivalent to selecting a "null service" as described above. Access to broadcast resources shall be lost and the object shall transition to the `unrealized` state.

A broadcast-related application can transition to a broadcast-independent application by calling the `setChannel()` method on the video/broadcast object with a value of `null` for its `channel` argument. Access to broadcast resources shall be lost and the object shall transition to the `unrealized` state. A `ChannelChangeSucceededEvent` shall be dispatched to the video/broadcast object that caused the transition with a value of `null` for the `channel` property.

NOTE: Applications that wish to become broadcast-independent and later transition back to broadcast-related should remember the current channel before transitioning to broadcast-independent.

When a broadcast-independent application successfully selects a broadcast service using a video/broadcast object, that application shall be killed unless all the following conditions are met:

- The broadcast-independent application has an `organization_id` and `application_id` (whether obtained through a broadcast AIT or an XML AIT).
- An application of the same `organization_id` and `application_id` is signalled in the broadcast channel to be selected with control code `AUTOSTART` or `PRESENT`.
- The application signalled in the broadcast channel with the same `organization_id` and `application_id` includes a `transport_protocol_descriptor` with `protocol_id` equal to 3.
- The URL of the entry point document of the broadcast-independent application has the same origin as at least one of the URLs signalled in the broadcast for that `organization_id` and `application_id`.
- The URL of the page currently loaded in the broadcast-independent application is inside the application boundary of the application as defined in clause 6.3.

If these conditions are met, the application shall transition to be a broadcast-related application as defined in clause 6.2.2.2. The application should be authored to follow the behaviour defined in clause 5.3.3.

6.2.2.7 Suspension of access to broadcast resources

This clause shall apply to terminals which do not have the hardware capability to present broadband delivered video at the same time as demultiplexing MPEG-2 sections from the broadcast.

Attempting to present broadband delivered video using the AV Control object may result in suspension of access to broadcast resources, including but not limited to:

- AIT monitoring being paused.
- Files in a carousel no longer being accessible.
- DSM-CC stream event monitoring being paused.
- Broadcast video presentation being stopped.
- Not dispatching `ProgrammesChanged` events.

Suspension of access to broadcast resources shall be treated as a transient error as defined in table 11 - "State transitions for the video/broadcast embedded object" of the OIPF DAE specification [1]. The `PlayStateChange` Event that is dispatched shall have the error code 11.

When playback of broadband delivered video terminates for any reason and no broadband-delivered media item is queued and access to broadcast resources was previously suspended due to the presentation of broadband-delivered video, the following actions shall be taken by the terminal:

- AIT monitoring shall resume.
- Access to files in a broadcast carousel shall be automatically restored.
- DSM-CC stream event monitoring shall resume.
- Broadcast video presentation shall resume.
- Dispatching `ProgrammesChanged` events shall resume.

When access to broadcast resources is restored following earlier suspension of access, this shall be treated as recovery from a transient error as defined in table 11 - "State transitions for the video/broadcast embedded object" of the OIPF DAE specification [1].

For consistent behaviour, broadcast-related applications which wish to present long items of broadband delivered video should either:

- a) make themselves broadcast-independent as defined in clause 6.2.2.6; or
- b) be permanently signalled in the AIT by the broadcaster.

6.2.2.8 Behaviour on encrypted broadcast services

Some channels may have the broadcast content encrypted, preventing those terminals without the appropriate CAS and rights from decoding and presenting the content. In these cases, clauses 6.2.2.2 and 6.2.2.3 remain applicable even when the terminal fails to decode some or all of the components.

In particular, terminals shall behave as follows:

- Failure to decrypt the AIT is identical to having no AIT present on that channel.
- Failure to decrypt the carousel containing the application is identical to failing to load the application from broadcast protocol.

NOTE: The present document is intentionally silent about requirements for terminals to support decryption of encrypted AITs, object carousels and other data components.

Applications associated with channels which may be encrypted are advised to check whether the content is being presented (using the error parameter provided in the onPlayStateChange method of the video/broadcast object) and to modify their behaviour accordingly. For instance, if the content is not being presented, the application may wish to display some advertising message indicating how the user may gain access to this channel. Applications should not remain hidden or show a mainly transparent screen.

6.2.3 Application lifecycle example (informative)

Figure 15 and table 3 illustrate the application model defined above.

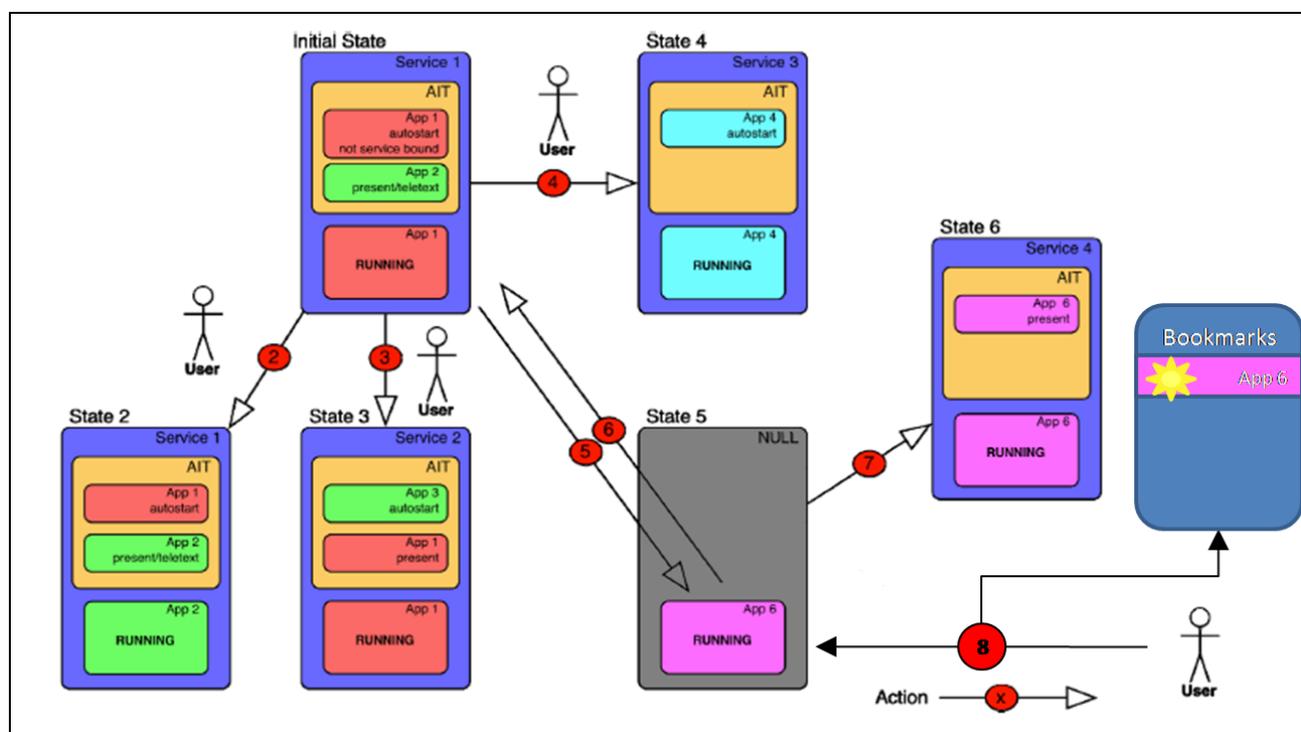


Figure 15: Application model examples

Table 3: Descriptions of actions and resulting state changes

Starting state	Action	Resulting state
Initial State: Application 1 is running	2: User presses "TEXT" key	State 2: Application 2 will be started due to TELETEXT signalling.
Initial State: Application 1 is running	3: User selects service 2	State 3: Application 1 keeps running assuming it is not service-bound.
Initial State: Application 1 is running	4: User selects service 3	State 4: Application 1 will be killed and Application 4 will be started due to AUTOSTART signalling.
Initial State: Application 1 is running	5: Application call to <code>createApplication()</code> with an XML AIT to start a broadcast-independent application	State 5: Broadcast-independent application 6 is running. Any former presentation of service components will be stopped. The application has an application identifier as it was started from an XML AIT. See also action #7.
State 5: Application 6 is running	6: User selects Service 1	State 1: Application 6 will be stopped and Application 1 will be started due to AUTOSTART signalling.
State 5: Application 6 is running	7: Application 6 selects service 4	State 6: Presentation of service 4 starts. Application 6 is signalled on service 4. It transitions to broadcast-related and keeps running.
	8: User enters URL of XML AIT or initial page to start application and to store it in his bookmarks. Terminal takes application title and logo for bookmark entry as signalled in HTML header.	State 5: same as for action 5.

6.3 Application boundary

Every application is associated with an application boundary. This is defined as follows:

- An application boundary is a set of URL origins and object carousels.
- If the origin of a URL is the same as one of the origins in the application boundary, that URL is said to be inside the application boundary.
 - The origin for URLs shall be as defined in "The Web Origin Concept" specification [25].
- If an object carousel is identical to one of the carousels in the application boundary, that carousel is said to be inside the application boundary:
 - The requirements for two object carousels to be identical shall be as defined in clause B.2.10 of TS 102 809 [3].

NOTE 1: For carousels delivered by different transport streams, the terminal compares the two `carousel_ids`. The use of the broadcaster's `organization_id` in the 24 MSBs of the two `carousel_ids` is a means to obtain unique `carousel_ids` and is not visible to the terminal.

- For applications loaded via HTTP or HTTPS, the application boundary shall include the origin of the URL used to launch the application e.g. as signalled in the AIT or XML AIT or passed as argument of `createApplication()`.

NOTE 2: This means that the default boundary is the tuple (scheme, host, port) of the application URL before any redirect, where the port component is the default port if not otherwise specified.

- For applications loaded via object carousel, the application boundary shall include the carousel from which the first page of the application was loaded.

- A `simple_application_boundary_descriptor` may be present in the AIT or an `applicationBoundary` element may be present in the XML AIT. As described in clauses 7.2.3.1 and 7.2.3.2 of the present document, these may include:
 - one or more HTTP or HTTPS URLs prefixes. The application boundary shall be extended to include also the origins of such prefix if this will not result in having origins from more than one host in the boundary. Otherwise the additional origin shall be ignored.

NOTE 3: this means that the boundary cannot be extended to cover more than one FQDN.

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

Launching a new application by using the method `createApplication()` (with an arbitrary new start page) or killing the current application and starting a new one via application signalling shall result in losing the association with the current application boundary (i.e. the new application will have a new boundary as defined in this clause).

Documents loaded from outside the application boundary shall be untrusted (in the sense of the word "trusted" as defined in clause 11), for example documents loaded in an `<iframe>` element or documents loaded as a result of following a link or an HTTP redirect. Following a link or an HTTP redirect from outside the application boundary back inside the application boundary shall restore the trust level to the original trust level of the application.

NOTE 4: An application being broadcast-related or broadcast-independent is not impacted by this change in trust level.

For files requested with `XMLHttpRequest`, the Same-Origin Policy shall be extended using the application boundary i.e. any origin in the application boundary will be considered of same origin.

7 Formats and protocols

7.1 General formats and protocols

7.1.1 Graphic formats

The graphics formats used shall comply with clause 9.1 of the OIPF media formats specification [2].

Table 4 lists the graphics formats that shall be supported.

Table 4: Graphics formats

Image Format	MIME Type
JPEG	image/jpeg
GIF	image/gif
PNG	image/png

7.1.2 Audio description

For the broadcast connection, signalling of audio description is defined by the appropriate specifications for each market where the terminals are to be deployed. Signalling of audio description for MPEG-2 transport streams delivered by the broadband connection shall follow the specification for the broadcast connection (if any).

NOTE: Typically most countries will use one of the 3 mechanisms from clause 8.4.2 of the OIPF DAE specification [1] but the present document does not require that.

For ISO format files, signalling is only defined to identify audio description streams when these are delivered using DASH. In this case, the signalling is defined in clause E.2.4, "Role Related Requirements".

Presenting a broadcast-mix audio description stream is supported since this is no different from presenting any other alternative audio stream.

Presenting receiver-mix audio description streams is not required by the present document.

To the extent that audio description is supported, it shall be exposed to applications as defined in clause 8.4.5 of the OIPF DAE specification [1].

7.2 Broadcast-specific format and protocols

7.2.1 System, video, audio and subtitle formats

The present document does not contain any requirements for system, video, audio and subtitle formats for the broadcast channel. These requirements are defined by the appropriate specifications for each market where the terminals are to be deployed.

7.2.2 Protocol for application transport

DSM-CC object carousel as defined in clause 7 of TS 102 809 [3] shall be supported.

Broadcasters shall ensure that the DSM-CC sections for a carousel are distributed over 3 or fewer elementary streams. StreamEvent sections may be carried in additional elementary stream(s).

Support for the `catching_priority_descriptor` as defined in clause B.2.2.4.2 of TS 102 809 [3] is not included.

The use of the `deferred_association_tags_descriptor` for the purpose of referencing an elementary stream (TS 102 809 [3], clauses B.3.1.1 and B.3.2) is not required by the present document. However this signalling may be present in a broadcast transport stream and acted upon by receivers that support this. Consequently, authors/broadcasters/operators should not expect this signalling to be ignored if it is present in the broadcast transport stream.

If elementary streams present in other services are to be referenced, then that elementary stream will also be required to be present in the current services PMT.

The use of the `deferred_association_tags_descriptor` to support the `BIOP_PROGRAM_USE` tap (TS 102 809 [3], clause B.3.1.2) is required by the present document.

7.2.3 Signalling of applications

7.2.3.1 Broadcast signalling

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].

Table 5: Supported application signalling features

Section	Page	Status	Notes
5.2.2 Application types	14	M	The application type shall be 0x0010.
5.2.3 Application identification	15	M	<code>application_ids</code> for trusted applications (as defined in the present document) shall be in the range for signed applications (as defined in TS 102 809 [3]). Applications signalled with an <code>application_id</code> in the range of unsigned application shall be started as untrusted. Applications signalled with an <code>application_id</code> in ranges other than signed and unsigned are outside the scope of the present document. If not otherwise required by other specifications, these applications shall not be started and discarded by the platform.
5.2.4 Application control codes	16	M	The following control codes shall be supported: 0x01 AUTOSTART 0x02 PRESENT 0x04 KILL 0x07 DISABLED The application life cycle shall follow the rules defined in TS 102 809 [3] and in the present document.

Section	Page	Status	Notes
5.2.5 Platform profiles	17	M	<p>For applications that only require the basic profile, the <code>application_profile</code> shall take the value <code>0x0000</code>. The following bits can be combined to express profiles corresponding to additional features that applications may require:</p> <p><code>0x0001</code> A/V content download feature <code>0x0002</code> PVR feature</p> <p>The 3 most significant bits of the <code>application_profile</code> are reserved for future use</p> <p>As defined in clause 5.2.5.1 of TS 102 809 [3], terminals shall be able to run all applications where the signalled application profile is one of the profiles supported by the terminal. All terminals shall support the basic profile (<code>0x0000</code>) in addition to profiles corresponding to the other features supported by the terminal.</p> <p>The <code>version</code> fields shall be set as follows: <code>version.major = 1</code> <code>version.minor = 2</code> <code>version.micro = 1</code></p>
5.2.6 Application visibility	18	See Notes	VISIBLE_ALL shall be signalled. Values other than VISIBLE_ALL are not included in the present document.
5.2.7 Application priority	18	M	
5.2.8 Application icons	19	O	The icon locator information shall be relative to the base part (constructed from the <code>URL_base_bytes</code>) of the URL as signalled in the <code>transport_protocol_descriptor</code> .
5.2.9 Graphics constraints	21	NI	
5.2.10 Application usage	22	M	Usage type <code>0x01</code> shall be supported as described in clauses 5.3.4 and 6.
5.2.11 Stored applications	23	NI	
5.2.12 Application Description File	26	NI	
5.3.2 Program specific information	28	M	
5.3.4 Application Information Table	29	M	<p>A maximum of one PID per service shall be used to carry the AIT sub-table defined by the Hybrid Broadcast Broadband TV application type.</p> <p>All sections of the Hybrid Broadcast Broadband TV AIT sub-table shall be transmitted at least once every second.</p>
5.3.5.1 Application signalling descriptor	33	M	If more than one stream is signalled in the PMT for a service with an <code>application_signalling_descriptor</code> , then the <code>application_signalling_descriptor</code> for the stream containing the AIT for the HbbTV application shall include the HbbTV <code>application_type</code> (<code>0x0010</code>).
5.3.5.2 Data broadcast id descriptor	33	O	The value to be used for the <code>data_broadcast_id</code> field of the <code>data_broadcast_id_descriptor</code> for Hybrid Broadcast Broadband TV carousels shall be <code>0x0123</code> . The <code>id_specific_data</code> are not defined. By supporting this optional feature, terminals can reduce the time needed to mount a carousel.
5.3.5.3 Application descriptor	34	M	
5.3.5.4 Application recording descriptor	35	M/NI	Support of the <code>application_recording_descriptor</code> is mandatory when the terminal has support for time-shift. Otherwise it is not included. The semantics of the <code>application_recording_descriptor</code> for HbbTV is clarified below this table.
5.3.5.5 Application usage descriptor	37	M	Usage type <code>0x01</code> shall be supported as described in clauses 5.3.4 and 6.
5.3.5.6 User information descriptors	38	M	
5.3.5.7 External application authorization descriptor	39	NI	
5.3.5.8 Graphics constraints descriptor	39	NI	

Section	Page	Status	Notes
5.3.6 Transport protocol descriptors	40	M	The following protocol_ids shall be supported: 0x0001 object carousel over broadcast channel 0x0003 HTTP over back channel (i.e. broadband connection).
5.3.7 Simple application location descriptor	43	M	
5.3.8 Simple application boundary descriptor	43	M	Only strict prefixes starting with "dvb://", "http://" or "https://" shall be supported. Only prefixes forming at least a second-level domain shall be supported. Path elements shall be ignored.
5.3.9 Service information	44	M	As modified by clause 7.2.6.
5.3.10 Stored applications	46	NI	

Table 6: Key to status column

Status	Description
M	MANDATORY The terminal shall support the referenced signalling. The signalling may be restricted to a subset specified in the "Notes" column. In that case all additional signalling is optional.
O	OPTIONAL It is the manufacturer's decision to support the referenced signalling.
NI	NOT INCLUDED The referenced signalling is not included in the present document. It should not be implemented unless required by another specification.

The semantics of the `application_recording_descriptor` are as follows;

- Applications that are safe to run in time-shift including trickmode shall set the `trick_mode_aware` flag and the `time_shift_flag` to '1'.
- The `scheduled_recording_flag` is not included.
- If applications are signalled with `trick_mode_aware` set to '0' the `timeshift_flag` shall be ignored.
- The `dynamic_flag` and `av_synced_flag` shall be used as defined by [TS102809]
- `initiating_replay_flag` is not included.
- `label_count`, `label_length`, `label_char` and `storage_properties` are not included.
- Applications shall list broadcasted data components in the component tag list. The elementary stream carrying the AIT does not need to be listed.

7.2.3.2 Broadcast-independent application signalling

The present document does not define any signalling, announcement or discovery of broadcast-independent applications. Clause 5.3.5 of the present document defines how they can be started. Broadcast-independent applications shall be identified either by the URL of the first page of the application or by the URL of a XML AIT clause 5.4 of TS 102 809 [3] and profiled in this clause. The XML file shall contain an application discovery record containing one or more application elements, all with the same `orgId` and `appId` values but with different application types. The XML file shall be delivered with HTTP or HTTPS using the "application/vnd.dvb.ait+xml" MIME type as defined in clause 5.4 of TS 102 809 [3].

The semantics of the fields and elements in the XML AIT file shall be as defined in table 7.

Table 7: Contents of XML AIT for Broadcast-independent applications

Field or element	Requirement on XML AIT file	Requirement on terminal
appName	Optional.	Optional for terminal to use.
applicationIdentifier	Mandatory.	Mandatory.
applicationDescriptor/ type/OtherApp	Shall be "application/vnd.hbbtv.xhtml+xml" for Hybrid Broadcast Broadband TV applications.	Mandatory. MIME types other than "application/vnd.hbbtv.xhtml+xml" are outside the scope of the present document.
applicationDescriptor/ controlCode	Shall be AUTOSTART.	Values other than AUTOSTART are outside the scope of the present document.
applicationDescriptor/ visibility	Shall be VISIBLE_ALL.	Values other than VISIBLE_ALL are outside the scope of the present document.
applicationDescriptor/ serviceBound	Shall be false.	Values other than false are outside the scope of the present document.
applicationDescriptor/ priority	Shall be present.	No defined semantics in the present document.
applicationDescriptor/ version	Outside the scope of the present document.	Outside the scope of the present document.
applicationDescriptor/ mhpVersion	Shall be the same values as defined for the MPEG-2 encoding of the AIT under "platform profiles" in table 5.	Values higher than those defined in table 5 shall result in the application failing to start.
applicationDescriptor/ icon	Optional.	Optional for terminal to use.
applicationDescriptor/ storageCapabilities	Outside the scope of the present document.	Outside the scope of the present document.
applicationTransport/	Mandatory. Shall be HTTPTransportType.	Mandatory.
applicationLocation/ applicationBoundary/	Mandatory. Optional.	Mandatory. Mandatory. Only strict prefixes starting with "dvb://", "http://" or "https://" shall be supported. Only prefixes forming at least a second-level domain shall be supported. Path elements shall be ignored.
applicationSpecificDescriptor/ot herDescriptor	Shall be HBBTVApplicationSpecificDescriptor as defined by the present document.	For future use.
applicationUsageDescriptor	Outside the scope of the present document.	Outside the scope of the present document.

Where a value, element or attribute is indicated as being outside the scope of the present document, the presence of this value, element or attribute in an XML AIT is not prohibited but the present document does not require any behaviour from terminals other than not suffering from a fatal error and continuing to parse the remainder of the XML AIT.

TS 102 809 [3] requires the definition of an "application specific descriptor" which is not used by the present document.

The following applicationSpecificDescriptor shall be supported;

```
<xs:complexType name="HBBTVApplicationSpecificDescriptor">
  <xs:complexContent>
    <xs:extension base="mis:ApplicationSpecificDescriptor">
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

7.2.4 Synchronization

The terminal shall support "do-it-now" events as defined in clause 8 of 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.

7.2.5 DSM-CC carousel

7.2.5.1 Mounting related constraints

A terminal shall mount a maximum of one carousel at a time for use by the running application. Mounting means that the terminal makes the latest version of the files of the carousel available to the application. Additionally a terminal may read, cache and monitor several carousels in parallel in order to decrease the loading time as experienced by the user.

Terminals shall support carousels split across up to and including three elementary streams simultaneously as defined in clause 10.2.1.

NOTE: Typically, mounting a carousel may involve reading data from the carousel into a cache and monitoring for updates to the carousel.

7.2.5.2 Initial carousel mounting

A broadcast-related application whose initial page is broadcast will cause its carousel to be mounted by the terminal (in order to be loaded and launched) unless mounting the carousel would require tuning to a transport stream other than the one carrying the current channel. If tuning would be required, the attempt to load the page shall fail as if the file did not exist.

A broadcast-related application whose initial page is not broadcast may mount a carousel on the same service using the `component_tag`, e.g. through an `XMLHttpRequest` request or a reference (e.g. from an `` element). If the elementary stream pointed to by the `component_tag` does not contain a service gateway, the mounting will fail.

The terminal shall not allow broadcast-independent applications to mount carousels. In order to mount a carousel or access any other broadcast resources, a broadcast-independent application will have to first become a broadcast-related application (see clause 6.2.2.6).

7.2.5.3 Subsequent carousel mountings (during the lifecycle of an application)

For a broadcast-related application, once a carousel has been mounted, a request that would require another carousel to be mounted shall succeed and cause the previous carousel to be un-mounted and all of its pending requests to be cancelled, unless mounting the carousel would require tuning to a transport stream other than the one carrying the current channel.

7.2.5.4 Constraints

A resolved DSM-CC object reference shall be at most 64 bytes.

7.2.6 Data Services

HbbTV services may exist that don't have any broadcast audio or video components (i.e. pure data services). Their broadcast signalling shall be as follows.

The SDT entry for the pure data service shall use a `service_descriptor` with a `service_type` of 0x0C. It shall also contain a `data_broadcast_descriptor` as defined in TS 102 809 [3] clause 5.3.9.1 with the following restrictions:

- The `data_broadcast_id` shall be 0x0123.
- The `selector_bytes` shall be present, and shall carry information about all HbbTV AUTOSTART applications that the service may carry.
- The application name and text and other private data may be present.

The signalling of the AIT and any HbbTV carousel remains the same as normal audio and video services.

Terminals shall process the `data_broadcast_descriptor` in the SDT and include, in the terminals service list, 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.

NOTE: The present document does not contain any requirements how broadcast channel lists are updated and managed. These requirements may be defined by the appropriate specifications for each market where the terminals are to be deployed.

Where an instance of the `Channel` class represents a data service, the value of the `channelType` property shall be 256.

7.3 Broadband-specific format and protocols

7.3.1 System, video and audio formats

7.3.1.1 General requirements

The system formats and their labels are specified in the OIPF Media Formats specification [2] with the restrictions in clause 7.3.1.2.

The video formats and their labels are specified in the OIPF Media Formats specification [2] with the restrictions in clause 7.3.1.3.

The audio formats are specified in the OIPF Media Formats specification [2] with the restrictions in clause 7.3.1.4.

Table 8 defines the subset of the combinations of system, video and audio formats specified in the OIPF Media Formats specification [2] that shall be supported.

Table 8: System, video and audio formats

System Format	Video Format	Audio Format	Subtitle format	MIME Type
TS	AVC_SD_25 AVC_HD_25	HEAAC E-AC3 (see note 1)	See note 2.	video/mpeg
MP4	AVC_SD_25 AVC_HD_25	HEAAC E-AC3 (see note 1)	Not defined in the present document.	video/mp4
NOTE 1: Terminals shall support E-AC3 for content received by the broadband connection when it is supported for the broadcast connection. Otherwise it is not mandated.				
NOTE 2: Terminals shall support the same subtitle formats for content received by the broadband connection as are supported for the broadcast connection.				

Table 9 defines the subset of the audio formats specified in the OIPF Media Formats specification [2] that shall be supported for audio-only services and audio clips.

Table 9: Formats for audio-only services and audio clips

Audio Format	MIME Type	Notes
MPEG1_L3	audio/mpeg	
HEAAC	audio/mp4	This is carriage of HE-AAC audio inside the MP4 system format container. This format shall comply with the requirements specified in section 8.6.35 of the DLNA media formats specification [26], except for section 8.6.35.11.

Playing WAVE audio from memory is not included in the present document. It should not be implemented unless required by another specification.

Examples of media which comply with the above supported codecs list:

- "http://myserver/myvideo.mp4", mimetype "video/mp4", container "mp4", 2,5 MBit/s, resolution 720*576 @ 25 frames per second, together with AAC LC sound @ 64 kBit/s.
- "http://myserver/myaudio.mp3", mimetype "audio/mpeg", container "mp3", 256 kBit/s.

7.3.1.2 Systems layers

The usage of the systems layer format MPEG-2 Transport Stream shall comply with clause 4 of the OIPF Media Formats specification [2]. Support for the DLNA extension "time stamped MPEG-2 transport stream" is not required.

The MP4 File Format shall comply with clause 4 of the OIPF Media Formats specification [2] and the following additions:

- For E-AC3 it shall comply with TS 102 366 [15] in addition
- The size of the moov box should not exceed 2,5 MByte

NOTE: Large moov boxes will slow down start up times especially for broadband connections with a small bandwidth.

- The large size field may be used. The size of a box should not exceed 4 GByte.

Bitrates of up to 8 MBit/sec for the stream (including protocol overheads, e.g. TCP and HTTP) shall be supported.

AIT signalling as defined in clause 7.2.3.1 shall not be processed for MPEG-2 TS delivered via unicast broadband content.

7.3.1.3 Video

The video format AVC_SD_25 shall comply with clauses 5.1.2.1 and 5.1.3 of the OIPF Media Formats specification [2].

The video format AVC_HD_25 shall comply with clauses 5.1.1.1 and 5.1.3 of the OIPF Media Formats specification [2].

7.3.1.4 Audio

Audio formats shall comply with clause 8.1 of the OIPF Media Formats specification [2] with the following additional requirements for multichannel audio:

- If the terminal supports a stereo output, it shall be capable of providing a down-mix of multichannel audio to stereo.
- If the terminal is equipped with a digital audio output then it shall be capable of providing the bitstream at this output (pass-through) and should be capable of transcoding multi-channel audio from HEAAC to AC3 format.
- The terminal shall use metadata, where provided, to control the stereo down-mix from multichannel audio, and shall use it, or pass it through, when providing bitstream output. Such metadata may be provided as described in the OIPF Media Formats specification [2] and clause 6.8 of TS 102 366 [15].

7.3.2 Protocols

7.3.2.1 Protocols for streaming

Unicast streaming using HTTP 1.1 shall be supported as defined in clause 5.2.2.2 of the OIPF protocols specification [4] with the addition that the Content-range header shall be supported in seek operations thus allowing the application to seek to any arbitrary position within the streaming video without the need of downloading the complete video first. The terminal should only buffer data equivalent to approximately 10 seconds of normal play in advance of the current play position unless the download rate is consistently lower than the consumption rate. If the Content-Length header is not provided terminals shall not make any assumptions on the size of the buffer on the server. Hence terminals which need to obtain some data from the stream, e.g. for initialisation, cannot assume that this data is still buffered on the server once they have completed their initialisation.

The accuracy of seeking to a particular point in time within an MPEG-2 transport stream is implementation dependent. Applications should avoid this except for small seeks relative to the current position in a stream that is already being played which are likely to be the least inaccurate. Seeking is likely to be more accurate in a constant bit-rate stream than a variable bit-rate one.

HTTP chunked transfer coding shall be supported as defined by section 3.6.1 of RFC 2616 [6].

NOTE: Live content delivered using HTTP chunked transfer encoding is presented using the A/V control object. There are no requirements for the <video/broadcast> object to present content delivered using HTTP.

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

7.3.2.2 Protocols for download

Where content download is supported, HTTP shall be supported as defined in clause 5.2.3 of the OIPF protocols specification [4].

7.3.2.3 Protocols for application transport

In addition to the requirements of clauses 9.1.1.1 and 9.1.1.2 of the OIPF DAE specification [1], when using HTTP over TLS the server may send a client certificate request during the TLS handshake as defined in RFC 2818 [7]. The TLS stack implementation shall support negotiation and delivery of client certificates to the server as defined in RFC 5246 [8], RFC 4346 [21] and RFC 2246 [20]. The client certificate shall comply with RFC 5280 [9]. The provision of these certificates is outside the scope of the present document as explained in clause 11.3.

7.3.2.4 HTTP User-Agent header

All outgoing HTTP requests made on behalf of an Hybrid Broadcast Broadband TV application shall include a *User-Agent* header using the syntax described in this clause.

The *User-Agent* header shall include:

```
HbbTV/1.2.1 (<capabilities>; [<vendorName>]; [<modelName>]; [<softwareVersion>];
[<hardwareVersion>]; <reserved>)
```

Where:

- The <capabilities> field consists of zero or more concatenated Hybrid Broadcast Broadband TV option strings as defined in clause 10.2.4.
- The <vendorName>, <modelName>, <softwareVersion> and <hardwareVersion> fields are the same as the one defined in the *application/oipfRemoteManagement* object in the OIPF DAE specification [1] and are optional.
- The <reserved> field is reserved for future extensions.

This *User-Agent* header may be extended with other implementation-specific information including other user agent information. In particular, it is recommended to include the browser user agent information.

Valid examples of this syntax are:

```
User-Agent: HbbTV/1.1.1 (+PVR+DL; Sonic; TV44; 1.32.455; 2.002;)
```

```
User-Agent: HbbTV/1.1.1 (;;;;)
```

7.3.2.5 HTTP Redirects

HTTP redirects as defined in [HTTP] in response to a HTTP request shall be supported as described in this clause.

- The terminal shall support responses with a status code of "302 Found" and "307 Temporary Redirect" by using the temporary URL given in the Location field.
- The terminal shall support at least one redirection.

8 Browser application environment

8.1 DAE Specification Usage

The OIPF DAE specification [1] shall be supported as defined in annex A of the present document.

8.2 Defined JavaScript APIs

8.2.1 Acquisition of DSM-CC stream events

8.2.1.1 Adding and removing stream event listeners

The following additional methods on the video/broadcast object (as defined in the OIPF DAE specification [1]) shall be supported for synchronization to broadcast events as defined in clause 7.2.4.

void addStreamEventListener(String targetURL, String eventName, EventListener listener)		
Description	<p>Add a listener for the specified DSM-CC stream event.</p> <p>When a broadcaster transmits an identical instance of the MPEG private data section carrying a stream event descriptor (including the version number), only one <code>StreamEvent</code> event shall be dispatched.</p> <p>When a broadcaster transmits different events using the same event name id (i.e. with different version numbers), one <code>StreamEvent</code> event shall be dispatched for each different stream event descriptor received.</p> <p>An event shall also be dispatched in case of error.</p> <p>Listeners can only be added while the video/broadcast object is in the Presenting or Stopped states. Calls to this function when the video/broadcast object is in other states shall have no effect.</p> <p>The terminal shall automatically unregister all listeners on the video/broadcast object in the following cases:</p> <ul style="list-style-type: none"> • A transition to the Unrealized state (e.g. when becoming broadcast-independent). • A transition to the Connecting state that is due to a channel change. <p>Listeners are not unregistered when transitioning to the Connecting state due to a transient error that does not result in a change of channel.</p>	
Arguments	targetURL	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]) describing the event.
	eventName	The name of the event (of the DSM-CC StreamEvent object) that should be subscribed to.
	listener	The listener for the event.

void removeStreamEventListener(String eventURL, String eventName, EventListener listener)		
Description	Remove a stream event listener for the specified stream event name.	
Arguments	targetURL	The URL of the DSM-CC StreamEvent object or an HTTP or HTTPS URL referring to an event description file describing the event.
	eventName	The name of the event (of the DSM-CC StreamEvent object) whose subscription should be removed.
	listener	The listener for the event.

8.2.1.2 DSM-CC StreamEvent event

<pre>interface StreamEvent : Event { readonly attribute String name; readonly attribute String data; readonly attribute String text; readonly attribute DOMString status; }</pre>		
Properties	name	The name of the DSM-CC StreamEvent's event.
	data	Data of the DSM-CC StreamEvent's event encoded in hexadecimal. EXAMPLE: "0A10B81033" (for a payload 5 bytes long).
	text	Text data of the DSM-CC StreamEvent's event as a string assuming UTF-8 as the encoding for the DSM-CC StreamEvent's event. Characters that cannot be transcoded are skipped.
	status	<p>Equal to "trigger" when the event is dispatched in response to a trigger in the stream or "error" when an error occurred (e.g. attempting to add a listener for an event that does not exist, or when a StreamEvent object with registered listeners is removed from the carousel).</p> <p>An event might be dispatched with an error status if:</p> <ul style="list-style-type: none"> the StreamEvent object pointed to by <code>targetURL</code> is not found in the carousel or via broadband; the StreamEvent object pointed to by <code>targetURL</code> does not contain the event specified by the <code>eventName</code> parameter; the carousel cannot be mounted; the elementary stream which contains the StreamEvent event descriptor is no longer being monitored (e.g. due to another monitoring request or because it disappears from the PMT). <p>Once an error is dispatched, the listener is automatically unregistered by the terminal.</p>

8.2.2 Carousel objects access with XMLHttpRequest

In order to access to the content of a carousel file, the `XMLHttpRequest` object can be used with the following constraints:

- Parameters passed to the `open()` method:
 - `method`: Shall be set to "GET".
 - `url`: Can be relative (to the location of the current page in the carousel's file system) or an absolute `dvb`: URL.
 - `async`: shall be set to `true`.
 - `user` and `password`: Ignored.
- `status`: Set to 200 when the DSM-CC object is found and to 404 if it is not present in the carousel or if the carousel has been unmounted (due to another request).
- `statusText`: Set to an empty string.
- Headers are not relevant for carousel access:
 - Calls to `setRequestHeader()` are ignored.
 - `getResponseHeader()` shall return null and `getAllResponseHeaders()` shall return an empty string.
- Values of the `responseText` and `responseXML` properties are shown in table 10.

Table 10: Values of the responseText and responseXML properties

DSM-CC object	URL example	responseText	responseXML
File	/weather/data.xml	Returns the "text response entity body" as defined in XMLHttpRequest [11]	If the file has the extension ".xml", returns the "XML response entity body" as defined in XMLHttpRequest [11]. Otherwise, returns null.
Directory	/weather	Comma-separated list of names (File name, Stream Event name or Directory name) of all objects in the directory. These names shall not include path information.	null
Stream Event	/weather/main/streamEvt1	Comma-separated list of names of all events in the Stream Event object.	null

Examples of `dvb`: URLs that may be used with the `XMLHttpRequest` object are:

```
/weather/data.xml (absolute path from the root of the carousel of the current page)
../weather/data.xml (relative path to the current page)
dvb://1..1.B8/weather/data.xml (0xB8 is the component tag)
```

9 System integration

9.1 Mapping from APIs to protocols

9.1.1 Unicast streaming

9.1.1.1 General streaming requirements

In Unicast streaming:

- Pausing playback shall cause the video to freeze and the audio to suspend.
- Stopping playback shall cause the video and audio to stop.
- When not presenting video, The AV Control object shall be rendered as an opaque black rectangle.

9.1.1.2 HTTP streaming

The mapping from the APIs for unicast streaming to the protocols shall be as defined in clause 8.2.3.1 of the OIPF DAE specification [1] for HTTP streaming.

9.1.2 Unicast content download

Where unicast content download is supported, the mapping from the APIs for unicast content download to the protocols shall be as defined in clause 8.2.1.1 of the OIPF DAE specification [1].

9.2 URLs

The `http:` and `https:` URL schemes shall be supported as defined in clause 8.3 of the OIPF DAE specification [1].

The `dvb:` URL scheme as defined in TS 102 851 [10] shall be supported and extended as follows:

- It shall be possible to use `dvb:` URLs including path references to refer to DSM-CC file objects and to DSM-CC stream event objects signalled in the current service. It shall be possible to append to URLs referring to DSM-CC file objects an optional query component or fragment component, e.g. to pass parameters to an application. Since '?' and '#' are reserved characters as defined in RFC 3986 [27], if the name of a DSM-CC file object that is part of an HbbTV application contains such characters, they shall be percent-encoded (as defined in RFC 3986 [27]) when used in URLs.
- It shall be possible to use `dvb:` URLs referring to applications signalled in the current service as defined in table 4 of TS 102 851 [10] 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. 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.
 - If only one URL contains a query component then the resulting URL shall use that query component.
 - If both URLs contain a query component then the query component of the DVB application URL is appended to the application location URL using an ampersand sign '&'. The terminal shall not parse or process the query components.
 - If only one URL contains a fragment component then the resulting URL shall use that fragment component.
 - If both URLs contain a fragment component, the fragment component of the DVB application URL takes precedence and overwrites the one in the application location URL.
 - The `window.location.href` property shall take the value of the resulting URL, including any query component. Any fragment component shall be available in the `window.location.hash` property and the query component in the `window.location.search` property.
 - Examples for a resulting URL include:
 - URL signaled in the AIT: <http://www.example.com/app1?param1=value1>
 createApplication URL: `dvb://current.ait/1.1?param2=value2#foo`
 Resulting URL: `http://www.example.com/app1?param1=value1¶m2=value2#foo`
 - URL signaled in the AIT: <http://www.example.com/app1?param1=value1#test>
 createApplication URL: `dvb://current.ait/1.1#foo`
 Resulting URL: `http://www.example.com/app1?param1=value1 #foo`
 - The application is signaled in a DSMCC Carousel with a Component Tag of 4 and a Base URL of `/index.php?param1=value1` and the current service location is `dvb://1.2.3`
 createApplication URL: `dvb://current.ait/1.1?param2=value2#foo`
 Resulting URL: `dvb://1.2.3.4/index.php?param1=value1¶m2=value2#foo`
- Use of `dvb:` URLs referring to files in a carousel carried in a different transport stream shall not cause the terminal to perform a tuning operation, and shall fail as if the file did not exist.
- Use of `dvb:` URLs referring to files in a different carousel carried in the same transport stream shall cause the terminal to unmount the currently mounted carousel and mount the new carousel, as specified in clause 7.2.5.3.
- Support for DVB URLs including the textual service identifier is not required in the present document.

9.3 Other file formats

9.3.1 Stream event

Both mechanisms for referencing sources of stream events defined in clause 8.2 of TS 102 809 [3] shall be supported.

For the XML schema defined in clause 8.2 of TS 102 809 [3] the following restrictions shall apply:

- The `stream_event_id` attribute of the type `StreamEventType` shall represent a positive/unsigned integer with a maximum value of 65535. The lexical representation of the value shall be as defined by clause 3.3.23 "unsignedShort" of [23].
- The value of the `component_tag` attribute of the type `DsmccObjectType` shall represent a positive/unsigned integer with a maximum value of 255. The lexical representation of the value shall be as defined by clause 3.3.24 "unsignedByte" of [23].
- Stream event XML files shall be served with a MIME type of "application/vnd.dvb.streamevent+xml".

9.4 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 [29] and identified by the MIME type in annex C of that document. The type attribute of the A/V object shall be set to "application/dash+xml".

NOTE: This is an intentional deviation from requirement 5.7.1.a of CEA-2014 [i.1] where the type attribute contains the type of the video or audio.

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 [29], 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 [29] is supported then terminals shall support the `<ContentURL>` element of this descriptor referencing an MPD as defined in DASH [29]. In these circumstances, the other requirements from the preceding paragraph shall apply.

If the terminal supports trick modes, the behaviour defined in clause A.2.3 shall be supported including the generation of a `PlaySpeedChangedd` event reporting the actual speed of fast forwards and fast rewind.

When media content components are delivered using DASH:

- Instances of the `AVComponent` class shall refer to `AdaptationSets` carrying audio, video or subtitles.
- The `componentTag` shall be the value of the `id` attribute on the `Adaptation Set` (if provided).

When an instance of the `AVComponent` class refers to a DASH audio media content component:

- If the audio media component is identified as being audio description (as defined in clause E.2.4 Role Related Requirements below), the `audioDescription` property of the `AVComponent` shall be set to true.
- 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.

10 Capabilities

10.1 Display model

This clause is replaced by annex H, "Display Model" of the OIPF DAE specification [1].

10.2 Terminal capabilities and functions

10.2.1 Minimum terminal capabilities

Minimum terminal capabilities which shall be available to applications are listed in table 11 for general capabilities. Additional capabilities should be signalled in the capability profile as defined by clause 9 of the OIPF DAE specification [1].

Table 11: Minimum terminal capabilities

	Value	Characteristic
Hybrid Broadcast Broadband TV application graphic plane resolution	1 280 pixels horizontally by 720 pixels vertically with a 16:9 aspect ratio.	The terminal shall have at least this graphics resolution. If it is physically higher than this then the resolution shall appear to the applications to be exactly 1 280 x 720 pixels.
Colour format	RGBA32 should be supported. If an implementation uses lower colour resolutions (e.g. RGBA16) then it shall support at least RGBA4444.	Video overlays supported.
Supported proportional font	<p>"Tiresias Screenfont" v8.03 (or equivalent) with the support for the Unicode character range "Generic Western European character set" as defined in annex C of TS 102 809 [3] but excluding the unicode character codes 0149 and 066B.</p> <p>The font shall be the default font to be used when none is explicitly specified.</p> <p>This font (even if it is an equivalent of "Tiresias Screenfont" v8.03) shall be accessible with the following CSS rule: <code>font-family: Tiresias;</code></p> <p>It shall also possible to use the "sans-serif" generic family name to point to the "Tiresias Screenfont" v8.03 font (even if other sans-serif fonts are available in the terminal), i.e. "sans-serif" shall default to the "Tiresias Screenfont" v8.03 font: <code>font-family: sans-serif;</code></p>	Sans-serif, scalable and anti-aliased font.
Supported non-proportional font	<p>"Letter Gothic 12 Pitch" (or equivalent) with the support for the Unicode character range "Generic Western European character set" as defined in annex C of TS 102 809 [3] but excluding the unicode character codes 0149 and 066B.</p> <p>This font (even if it is an equivalent of "Letter Gothic 12 Pitch") shall be accessible with the following CSS rule: <code>font-family: "Letter Gothic 12 Pitch";</code></p> <p>It shall also possible to use the "monospace" generic family name to point to the "Letter Gothic 12 Pitch" font (even if other monospace fonts are available in the terminal), i.e. "monospace" shall default to the "Letter Gothic 12 Pitch" font: <code>font-family: monospace;</code></p>	Monospace, scalable and anti-aliased font.

	Value	Characteristic
Text entry method	<p>Either multi-tap (e.g. as defined in ES 202 130 [i.2]) or an equivalent (e.g. software keyboard) where characters are input character by character in the text field.</p> <p>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.</p> <p>NOTE: The <code>input-format</code> CSS property may be used by terminals to determine which text entry method to use.</p>	NOTE: Multi-tap aka SMS-tap is not to be confused with T9 text entry which is not required.
Minimum number of DSM-CC related section filters	<p>The terminal shall allocate sufficient resources to acquire DSM-CC sections from at least 3 elementary streams simultaneously for a given DSM-CC carousel.</p> <p>In addition, a terminal shall reserve at least one section filter for monitoring DSM-CC StreamEvent's events.</p>	
Minimum DSM-CC cache size	The terminal shall reserve 3 MByte for caching objects carried in DSM-CC object carousels.	
System layer for unicast streaming using HTTP and file download	Both MPEG-2 TS and MP4 file format (as defined in clause 7.3.1.2) shall be supported.	
Video formats for unicast streaming using HTTP and file download	Both AVC_SD_25 and AVC_HD_25 shall be supported (as defined in clause 7.3.1.3).	
Audio format for unicast streaming using HTTP and file download	HEAAC, E-AC3 and MPEG1_L3 as defined in clauses 7.3.1.1 and 7.3.1.4.	
Audio format for audio from memory	HEAAC shall be supported (as defined in clause 6.3.2 of the OIPF DAE specification [1]).	
PVR management	If the PVR feature is supported, the <code>manageRecordings</code> attribute of the recording capability shall have the value "samedomain".	See clause 9.3.3 of the OIPF DAE specification [1].
Download management	If content download is supported, the <code>manageDownload</code> attribute of the download capability shall have the value "samedomain".	See clause 9.3.4 of the OIPF DAE specification [1].
Simultaneous demultiplexing of broadcast and broadband content	Not required (see clause 6.2.2.7).	
Parental rating scheme	Terminal shall at least support the scheme of a minimum recommended age encoded as per EN 300 468 [16].	

	Value	Characteristic
Video scaling	<p>Terminals shall be able to present video at sizes down to 1/8 by 1/8 of the width and height of the logical video plane - equivalent to 160 x 90 pixels in the Hybrid Broadcast Broadband TV application graphics plane.</p> <p>Terminals shall be able to scale video down to 1/4 by 1/4 and should be able to scale video down to 1/8 by 1/8. For sizes between 1/4 by 1/4 and 1/8 by 1/8, terminals which cannot scale video shall crop the video instead and display it centered in the according video object of the Hybrid Broadcast Broadband TV application graphics plane.</p> <p>Terminals shall be able to scale video up to 2 x 2 of the width and height of the logical video plane.</p> <p>Within these limits, any arbitrary scaling factor shall be allowed. The aspect ratio of decoded video shall be preserved such that all of the decoded video is visible within the area of the video/broadcast or AV Control object.</p>	
Cookie support	<p>Cookies with an expiry date shall be stored in persistent memory. Terminals shall respect the expiry date of the cookie.</p> <p>Terminal SHALL follow RFC 6265 [24] when implementing cookies support.</p> <p>Since section 6.1 of RFC 6265 [24] does not fix strict limits, the present document fix the following minimum capabilities that terminals SHALL support:</p> <ul style="list-style-type: none"> - At least 4 096 bytes per cookie (as measured by the sum of the length of the cookie's name, value, and attributes). - At least 20 cookies per domain. - At least 100 cookies total. - At least 5 120 bytes for the "Set-Cookie" header. <p>NOTE: As implied by RFC 6265, if a cookie or a "Set-Cookie" header is bigger than the maximum size supported by the terminal, it will be discarded, not truncated.</p>	

An equivalent font is one for which all the following are true:

- The line height of both fonts is the same.
- The widths of the glyphs for corresponding character points are the same in both fonts (where the character point is defined in both fonts).
- The kerning tables contain the same values for both fonts where both of the character points in the pair are present in both fonts.
- Either the appearance of the glyphs is visually similar or they are valid glyph variants as defined by unicode.

10.2.2 User input

Implementations shall provide a mechanism for the end user to generate key events as defined in table 12.

Table 12: Key events and their status

Button (for conventional remote controls)	Key event	Status
4 colour buttons (red, green, yellow, blue)	VK_RED, VK_GREEN, VK_YELLOW, VK_BLUE	Mandatory
4 arrow buttons (up, down, left, right)	VK_UP, VK_DOWN, VK_LEFT, VK_RIGHT	Mandatory
ENTER or OK button	VK_ENTER	Mandatory
BACK button	VK_BACK	Mandatory
Number keys	VK_0 to VK_9 inclusive	Mandatory
Play, stop, pause	VK_STOP and either VK_PLAY and VK_PAUSE or VK_PLAY_PAUSE	Mandatory
Fast forward and fast rewind	VK_FAST_FWD VK_REWIND	Mandatory
Record	VK_RECORD	Mandatory if the PVR feature is supported, otherwise optional.
TEXT or TXT or comparable button	Not available to applications	mandatory
2 program selection buttons (e.g. P+ and P-)	Not available to applications	Optional
WEBTV or comparable button	Not available to applications	Optional
EXIT or TV or comparable button	Not available to applications	Optional

Key events which have a key code listed in the preceding table shall be available to all applications when requested through the `keySet` object. Key events which do not have a key code listed in the preceding table shall be handled by the implementation and not delivered to applications.

Support for direct keycodes (i.e. the `keyCode` property of the DOM 2 `KeyEvent` class) is not required.

Applications shall not rely on receiving any key events not requested through a `keySet` object, for example when the end user is inputting text into an input field. However, the set of key events requested via a `keySet` object only identifies the minimum set of keys that may be sent to an application, and so applications should not rely on receiving only those key events.

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.
- 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.

NOTE: Applications shall set the NAVIGATION bit of the keyset object even if the navigation keys are only used for focus based navigation (including the CSS `nav-*` properties) and not used in javascript event handlers.

Note that `VK_*` key codes are defined as properties of `KeyEvent` interface and do not have a "global" Javascript scope. For example, if an application wants to check if a user pressed the "Enter" key, it should use Javascript like the following code fragment:

```
if (e.keyCode == KeyEvent.VK_ENTER)
//handle the user input.
```

Furthermore constant values for `VK_*` key codes defined by CEA2014-A Annex F are OPTIONAL.

10.2.3 Terminal functions

10.2.3.1 Favourites and bookmarks

The terminal should provide a feature to organize frequently used broadcast-independent interactive applications as bookmarks or favourites.

For the presentation of applications on manufacturer portals or in favourite lists the terminal may use a title and an icon specified in the HTML head section and the URL of the initial page of the application:

- The application name is defined by the HTML title element.
- The application may have multiple title elements to provide a name in different languages using the lang attribute.
- The linking to an application icon is done by an HTML link element with the following attributes. See also [i.4]:
 - rel - shall have the value 'icon';
 - type - shall contain the mime type of the image format;
 - href - shall be the URL of the image.
- The image format and mime types of the icon shall be as defined in clause 7.1.1.
- An application may have multiple icons for different aspect ratios, e.g. 4 by 3 and square. It is recommended that an application provides at least one icon with a square aspect ratio.

10.2.3.2 Streaming and Download

Terminals shall not permit persistent storage of broadband delivered content whose delivery was initiated using the streaming API (the CEA-2014 AV Control object). Service providers who want to offer content for persistent download should use the download API.

10.2.3.3 PVR

It is up to the terminal to decide whether PVR feature related calls are executed directly or if additional means to determine whether to allow the call for the application are employed, such as opening a dialog to query the user.

10.2.4 Hybrid Broadcast Broadband TV reported capabilities and option strings

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 [28] shall be equal to the canonicalized form of the following XML:

```
<profilelist>
  <ui_profile name="OITF_HD_UIPROF+DVB_S+TRICK_MODE">
    <ext>
      <parentalcontrol schemes="dvd-si">true</parentalcontrol>
    </ext>
  </ui_profile>
  <clientMetadata type="dvd-si">true</clientMetadata>
  <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"/>
  <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" />
  <audio_profile name="MPEG1_L3" type="audio/mpeg"/>
  <audio_profile name="HEAAC" type="audio/mp4"/></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 "dvd-si" may be listed in the "<parentalcontrol>" element

NOTE: There are currently no <audio_profile> elements defined which include 'dash' as the transport attribute.

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

As mentioned in table 8, 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:

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

The strings defined in table 13 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 [1].

Table 13: Hybrid Broadcast Broadband TV 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).
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 as defined in Annex F of the OIPF DAE specification [1] 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 CI+ shall be indicated using the <drm> element defined in Annex F of the OIPF DAE specification [1] and providing the protectionGateways attribute with "ci+" string. For example:

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

10.2.5 Terminal memory requirements

The terminal shall provide sufficient memory for the reference Hybrid Broadcast Broadband TV application provided along with the present document to be successfully loaded and displayed. Once it is loaded, navigation should be operable. The provided screenshot gives an indication of what it should look like.

Different Hybrid Broadcast Broadband TV applications may use memory in different ways (for instance, more dynamically through repeating XMLHttpRequest requests) than this reference application while still being compliant with the present document.

Complying with this clause is necessary but not sufficient to guarantee running compliant Hybrid Broadcast Broadband TV services.

NOTE: In particular, other reference applications may be defined in the future which exercise the memory usage in a different way and even possibly require more memory. Practically, terminal products have to be tested against the applications expected to be in the market at time of product introduction.

10.2.6 Parental Access Control

10.2.6.1 Broadcast channel

Terminals shall support parental access control for the broadcast channel as required for the markets in which the products are to be sold or deployed. The details of this are outside the scope of the present document. Typically the end user may have to enter the appropriate PIN in order to obtain access to TV content above the parental rating threshold. The following shall apply if access to broadcast TV content is blocked as a result:

- If access to broadcast TV content is blocked when changing to a channel, this shall be reported to any running Hybrid Broadcast Broadband TV application which survives the channel change and has registered a listener for a `ChannelChangeError` event as an error with `errorState` 3 ("parental lock on channel").
- If access to broadcast TV content becomes blocked while a channel is selected, this shall be reported to any running Hybrid Broadcast Broadband TV application which has registered a listener for a `ParentalRatingChange` event.

In terminals where CI or CI+ [12] is supported, the CICAM may also enforce parental access control for the broadcast channel.

10.2.6.2 Streaming on-demand content

Applications offering access to streaming on-demand content shall obtain the parental rating system threshold set on the terminal and only stream appropriate content to the terminal.

10.2.6.3 Downloaded content

Broadcasters and service providers offering content for download shall populate the otherwise optional `<parentalRating>` element in the content access descriptor with the correct value for each content item downloaded. When playing back a downloaded content item, terminals shall compare the value in the `<parentalRating>` element in the content access descriptor used to download the content item with the current parental rating system threshold and only play appropriate content.

NOTE: The definition of what content is appropriate is outside the scope of the present document. Typically this could be any content under the threshold or content above the threshold where the end-user has entered a PIN.

If playback which was initiated by an Hybrid Broadcast Broadband TV application is blocked following such a comparison, the A/V object shall enter `playState` 6 error with the error property set to 7 ("content blocked due to parental control").

10.2.6.4 PVR

Broadcasters and service providers whose applications create `Programme` objects and pass them to the `record(Programme programme)` method of the application/`oipfRecordingScheduler` object shall populate the `parentalRating` property of the `Programme` object. Terminals shall obtain the parental rating information from DVB-SI at the time of recording and store this with the scheduled recording in the system and copy it to the in-progress recording once the recording process starts. Where a recording is scheduled using the `recordAt()` method, the parental rating assigned to the recording shall be the most restrictive value encountered during the recording process.

Before playing back a recording, terminals shall compare the parental rating stored with the recording with the current parental rating system threshold and shall only play appropriate content.

NOTE: The definition of what content is appropriate is outside the scope of the present document. Typically this could be any content under the threshold or content above the threshold where the end-user has entered a PIN.

If playback which was initiated by an Hybrid Broadcast Broadband TV application is blocked following such a comparison, the AV Control object shall enter playState 6 (error) with the error property set to 2 ("unknown error").

When playing back an in-progress recording, if the parental rating value of the recording changes, the terminal shall:

- Dispatch a `ParentalRatingChange` event.
- Compare the new parental rating value with the current parental rating threshold and, if the content has become inappropriate, the AV Control object shall enter playState 6 (error) with the error property set to 7 ("content blocked due to parental control").

10.2.7 Subtitles

Terminals shall support a method for the user to enable and disable subtitles and to select at least one preferred subtitle language. Terminals shall use this information when playing content to determine whether to present subtitles and to select between multiple subtitles when they are available.

Applications may change the terminal derived subtitle component selection and presentation status. The terminal shall maintain such changes made by an application until one of the following occurs:

- the application terminates,
- the application makes a further change,
- the video broadcast object or the A/V control object (as appropriate) is destroyed,
- the user makes a change using the terminal's subtitle selection mechanism,
- in the case of a video/broadcast object, the broadcast channel is changed either by an application as defined in the present document or by a mechanism outside the scope of the present document (e.g. the end-user pressing P+ or P- on a remote control).

If the subtitle components available in the content change and the previously selected component is no longer available, then the terminal may re-evaluate the subtitle component selection based on the user preferences.

11 Security

11.1 Application and service security

The present document defines two levels of trust for applications - trusted and not trusted. The features only available to trusted applications are listed in table A.1.

By default, broadcast related applications shall be trusted and broadcast-independent applications shall not be trusted. This may be modified as follows:

- Terminals may include a mechanism to allow the end-user to configure specific broadcast-independent applications as trusted or to configure broadcast-related applications from a particular service or channel as not being trusted.
- Terminals supporting reception of non-regulated channels should not automatically trust all applications from those channels.

EXAMPLE 1: In terminals supporting reception of satellite channels, for example, Hybrid Broadcast Broadband TV applications from adult channels on satellite should not be trusted except following explicit end-user approval and in compliance with appropriate regulation.

EXAMPLE 2: In terminals supporting reception of cable or terrestrial channels, if the markets addressed have the possibility of local or community access channels then Hybrid Broadcast Broadband TV applications from these channels are not required to be trusted.

The details of how regulated and non-regulated channels are identified are outside the scope of the present document.

- Terminals supporting cable or terrestrial reception of Hybrid Broadcast Broadband TV applications are not required to automatically trust all applications from all channels if different regulatory requirements apply to different channels. For example, Hybrid Broadcast Broadband TV applications from lightly or non-regulated local or community access channels which may be found in some markets are not required to be trusted. The details of how this could be achieved are outside the scope of the present document.
- Manufacturers may be able to configure specific broadcast-independent applications as being trusted and specific broadcast-related applications as being not trusted.
- Local regulation may impose additional requirements.

The security and permission mechanisms defined in clause 10.1 of the OIPF DAE specification [1] are not included in the present document. If they are included in a particular implementation then permissions should only be granted to an application where all mandatory parts of the feature or API covered by the permission are available.

NOTE: The set of features defined as available to trusted applications in the present document cannot be perfectly mapped onto the permissions defined in the OIPF DAE specification [1].

11.2 TLS and SSL Root Certificates

A list of root certificates is maintained at <http://www.hbbtv.org/spec/certificates.html>. The policy by which this list has been derived is outlined in annex D.

Terminals shall trust all root certificates identified as mandatory and may support those certificates identified as optional on that list, subject to the conditions in this clause.

Terminals should not trust any other root certificates.

NOTE: Including root certificates that are not on the list increases the risk of a man in the middle attack if those root certificates have not been audited to a similar or greater level than those on the list.

Terminals shall cease to trust any root certificates with RSA keys of less than 2048 bits after 31st December 2013.

Terminals shall support a means by which the device manufacturer can remove or distrust root certificates after manufacture. This may be handled either via a firmware upgrade mechanism or preferably via a specific root certificate update mechanism that could allow more timely updates.

A manufacturer may choose to remove or distrust a mandatory root certificate in the Terminal in response to a security threat.

Terminals should support a means of securely adding new root certificates after manufacture in order to maintain interoperability with servers over time.

11.3 TLS client certificates (informative)

In HTTP over TLS, the use of a client certificate authenticates the client to a service provider. Some business models require that an Hybrid Broadcast Broadband TV application is delivered exclusively to trusted Hybrid Broadcast Broadband TV terminal implementations.

NOTE: A compliance and certification regime is being defined which will include issuing formal Hybrid Broadcast Broadband TV client certificates to client devices.

11.4 CI+

11.4.1 CI+ Communication

Terminals supporting CI+ for protected content via broadcast shall support the following mapping from the application/oipfDrmAgent embedded object to the CI+ protocol as defined by clause 4.2.3 "CI+ based Gateway" of the OIPF CSP specification [5]:

- 4.2.3.1 Mandatory.
- 4.2.3.2 Mandatory.
- 4.2.3.3 Mandatory.
- 4.2.3.4 Mandatory, except for 4.2.3.4.1.2 and 4.2.3.4.3 which are Not Included.
- 4.2.3.5 N/A.
- 4.2.3.6 Not Included.
- 4.2.3.7 Mandatory using URI (Usage Rule Information) as defined in section 5.7 of CI Plus [13] if the PVR feature is supported otherwise 'Not Included'. The PVR resource as defined in section 15 of CI Plus [13] is Not Included.
- 4.2.3.8 Mandatory using URI (Usage Rule Information) as defined in section 5.7 of CI Plus [13] if the PVR feature is supported otherwise 'Not Included'. The PVR resource as defined in section 15 of CI Plus [13] is Not Included.
- 4.2.3.9 Not Included.
- 4.2.3.10 N/A.

Terminals supporting CI+ shall accept CI+ CICAMs that do not support the OIPF extensions defined by clause 4.2.3 'CI+ based Gateway' of the OIPF CSP specification [5]. Specifically, the failure for any reason to set up the SAS connection with the Open IPTV Forum `private_host_application_ID` shall not stop other CI+ functionality, that does not depend upon this connection, from working normally.

Terminals supporting an embedded CA solution should support a mapping from the application/oipfDrmAgent to the embedded CA system to provide the same functionality as defined above.

11.5 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 [30] and constrained by annex B of the present document.

Annex A (normative): OIPF DAE Specification Profile

A.1 Detailed section by section definition

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

Section, sub-section	Reference in DAE [1]	Status in Hybrid Broadcast Broadband TV	Notes	Security
Gateway Discovery and Control	4.2	NI		
Application Definition				
Application definition	4.3 excluding sub-clauses	M(*)	Modified by the present document concerning the application boundary and access to privileged capabilities.	
Similarities between applications and traditional web pages	4.3.1	M		
Difference between applications and traditional web pages	4.3.2	NI	The present document defines a model supporting one application executing at one time and does not include background applications. See clause 6.1 of the present document.	
The application tree	4.3.3	NI		
The application display model	4.3.4	M(*)	The present document requires a different application visualization mode from those referred to here.	
The Security model	4.3.5	NI	See clause 11.1 of the present document.	
Inheritance of permissions	4.3.6	NI		
Privileged applications APIs	4.3.7	NI	Not applicable.	
Active applications list	4.3.8	NI	Not applicable.	
Resource Management				
Application lifecycle issues	4.4.1	M(*)	Behaviour related to multiple applications loaded in the browser at the same time may not be applicable. <code>ApplicationUnloaded</code> events are not included.	
Caching of application files	4.4.2	NI	See clause 6.1 of the present document concerning "background preloading" of applications.	
Memory usage	4.4.3	M	The <code>gc()</code> method is not included.	
Instantiating embedded object and claiming scarce system resources	4.4.4	M		
Media control	4.4.5	M(*)	Shall be modified as defined in clause A.2.1.	
Use of the display	4.4.6	M(*)	The present document defines a different application visualization mode than those in clause 4.4.6.	
Cross-application event handling	4.4.7	NI	Not applicable in the present document.	
Browser History	4.4.8	M(*)	See clause A.2.6.4 of the present document.	

Section, sub-section	Reference in DAE [1]	Status in Hybrid Broadcast Broadband TV	Notes	Security
Parental access control	4.5	M	- Approach A shall be supported for streaming on demand content. - Approach B shall be supported where CI+ is supported. - Approach C shall always be supported. See clause 10.2.6.	
Content Download				
Download manager	4.6.1	M-D(*)	The application/oipfStatusView embedded object is not included.	Trusted
Content Access Download Descriptor	4.6.2	M-D		Trusted
Triggering a download	4.6.3	M-D		Trusted
Download protocol(s)	4.6.4	M-D		Trusted
Streaming CoD				
Unicast streaming	4.7.1	M(*)	Method 2 using an HTTP URL shall be supported. Method 3 shall be supported if the DRM feature is supported. Otherwise not included.	
Multicast streaming	4.7.2	NI		
Scheduled content				
Conveyance of channel list	4.8.1	M	Clause 4.8.1.2 is optional in DAE and not included in the present document.	Broadcast-related
Conveyance of channel list and list of scheduled recordings	4.8.2	M-P		Trusted
Display Model	4.9	M		
Application lifecycle				
Web applications	5.1.1.2	M	Web applications are equivalent to broadcast-independent applications in the present document.	
Using the Application.createApplication API call	5.1.1.3	M	See clauses 6.2.2.6 and 9.2 of the present document.	
CE-HTML third party notifications	5.1.1.4	NI		
Starting applications from SD&S Signalling	5.1.1.5	NI		
Applications started by the DRM agent	5.1.1.6	NI	Terminals should not start Hybrid Broadcast Broadband TV applications triggered by the DRM agent in order to avoid killing a currently running Hybrid Broadcast Broadband TV application which is trying to present the protected content. Instead it is recommend that applications trying to present protected content should handle DRM-specific UI themselves. Note that CI+ application MMI (see clause 5.5.2 of the present document) has some conceptual similarities with this but uses a different presentation technology.	
Applications provided by the AG through the remote UI	5.1.1.7	NI		
Stopping an application	5.1.2	M		

Section, sub-section	Reference in DAE [1]	Status in Hybrid Broadcast Broadband TV	Notes	Security
Application Boundaries	5.1.3	NI	This subject is addressed in substantially more detail by clause 6.3 of the present document.	
Application announcement and signalling	5.2	NI		
Event Notification				
Event Notification Framework based on CEA 2014 - Notif Socket	5.3.1.1	NI		
Event Notification Framework based on CEA 2014 - XMLHttpRequest	5.3.1.1	M		None
Out of Session event notification	5.3.1.2	NI		
IMS Event Notification Framework	5.3.2	NI		
Formats				
CE-HTML	6.1	M(*)	See clause A.2.6 of the present document.	
CE-HTML Referenced Formats	6.2	M		
Media formats	6.3	M(*)	See clause 7 of the present document.	
SVG	6.4	NI		
APIs				
Object Factory API	7.1	M(*)	Methods for creating objects not required by the present document are not included.	None
Applications Management APIs				
The application/oipfApplicationManager embedded object	7.2.1	M(*)	The <code>getOwnerApplication()</code> method, <code>onLowMemory</code> and <code>onApplicationLoadError</code> properties (and corresponding DOM 2 events) shall be supported. All other properties, methods and DOM 2 events are not included.	None
The Application class	7.2.2	M(*)	The following properties and methods shall be supported: - the property "privateData" - <code>createApplication (URI, false)</code> - <code>destroyApplication ()</code> - <code>show ()</code> - <code>hide ()</code> (broadcast independent applications should not call this method. Doing so may result in only the background being visible to the user) All other properties and methods are not included.	None
The ApplicationCollection class	7.2.3	NI		
The ApplicationPrivateData class	7.2.4	M(*)	The following properties and methods shall be supported: - <code>keyset</code> - <code>currentChannel</code> (see clause A.2.2 below) - <code>getFreeMem ()</code> All other properties and methods are not included.	None
The KeySet class	7.2.5	M(*)	The <code>otherKeys</code> and <code>maximumOtherKeys</code> properties are not included.	None
New DOM events for application support	7.2.6	NI		None

Section, sub-section	Reference in DAE [1]	Status in Hybrid Broadcast Broadband TV	Notes	Security
Configuration and Setting APIs				
The application/oipfConfiguration embedded object	7.3.1	M(*)	The configuration property shall be supported. All other properties, methods and events are not included.	None
The Configuration class	7.3.2	M(*)	Support for read-only access to the following properties is mandatory: - preferredAudioLanguage - preferredSubtitleLanguage - preferredUILanguage - countryID All other properties and methods are optional.	None
The LocalSystem class	7.3.3	NI		
The NetworkInterface class	7.3.4	NI		
The AVOutput class	7.3.5	NI		
The NetworkInterfaceCollection class	7.3.6	NI		
The AVOutputCollection class	7.3.7	NI		
Content Download APIs				
application/oipfDownloadTrigger embedded object	7.4.1	M-D(*)	The checkDownloadPossible() method is not included. For the other methods, the downloadStart parameter shall be ignored by terminals.	Trusted
Extensions to application/oipfDownloadTrigger	7.4.2	NI		
application/oipfDownloadManager embedded object	7.4.3	M-D(*)	The discInfo property is not included.	Trusted
The Download class	7.4.4	M-D		Trusted
The DownloadCollection class	7.4.5	M-D		Trusted
The DRMControllInformation class	7.4.6	M-D+ M-M		Trusted
The DRMControllInfoCollection class	7.4.7	M-D+ M-M	Mandatory if both Download and DRM features are supported - even if the supported DRM systems do not use the <DRMControllInformation> element inside the content access download descriptor. If the Download feature is supported and the terminal supports CI+ and if the terminal is capable of providing downloaded content to the CI+ CAM then these classes shall be supported - even if the CAS brought by a CI+ CAM do not use the <DRMControllInformation> element inside the content access download descriptor.	Trusted
Content On Demand Metadata APIs				
Content Service Protection API	7.6	M-C, M-M	Mandatory if the DRM feature is supported or if the terminal supports CI+.	Trusted
Gateway Discovery and Control APIs	7.7	NI		
IMS Related APIs	7.8	NI		
Parental access control APIs				
application/oipfParentalControl Manager embedded object	7.9.1	M(*)	The parentalRatingSchemes property shall be supported. Other properties and methods are not included.	None

Section, sub-section	Reference in DAE [1]	Status in Hybrid Broadcast Broadband TV	Notes	Security
The ParentalRatingScheme class	7.9.2	M	A scheme supporting DVB-SI age based rating shall be supported.	None
The ParentalRatingSchemeCollection class	7.9.3	M(*)	The addParentalRatingScheme () method is not included.	None
The ParentalRating class	7.9.4	M		None
The ParentalRatingCollection class	7.9.5	M(*)	The addParentalRating () method shall be supported if the PVR feature is supported and is otherwise not included. All other features of the class shall be supported.	None
Scheduled Recording APIs				
application/oiptRecordingScheduler embedded object	7.10.1	M-P		Trusted
The ScheduledRecording class	7.10.2	M-P(*)	"Only the following properties shall be supported: - startPadding - endPadding - name - description - startTime - duration - state - parentalRatings - channel All other properties are not included.	Trusted
The ScheduledRecordingCollection class	7.10.3	M-P		Trusted
Extension to application/oiptRecordingScheduler for control of recordings	7.10.4	M-P(*)	The recordings property shall be supported. Other properties, methods and events are not included.	Trusted
The Recording class	7.10.5	M-P(*)	The following properties shall be supported: - id - recordingStartTime - recordingDuration Since the Recording class implements the ScheduledRecording interface, the properties required to be supported from that interface as defined above are also required. All other properties are not included.	Trusted
The RecordingCollection class	7.10.6	NI		
The PVREvent class	7.10.7	NI		
The Bookmark class	7.10.8	NI		
The BookMarkCollection class	7.10.9	NI		
Remote Management APIs	7.11	NI		
Metadata APIs				
The application/oiptSearchManager 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

Section, sub-section	Reference in DAE [1]	Status in Hybrid Broadcast Broadband TV	Notes	Security
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 - "Programme.startTime", "Programme.name", "Programme.programmeID". These shall correspond to the properties of the same name. The addRatingConstraint, addCurrentRatingConstraint and addChannelConstraint (ChannelList) 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		Broadcast-related
The SearchResults class	7.12.4	M		Broadcast-related
The MetadataSearchEvent class	7.12.5	NI		
The MetadataUpdateEvent class	7.12.6	NI		
Broadcast video				
video/broadcast embedded object	7.13.1	M(*)	In the setChannel() method, the optional contentAccessDescriptorURL parameter may be ignored. The setVolume() and getVolume() methods are not included. The modifications in clause A.2.4 shall be supported.	See clause A.2.4
Extensions for recording and timeshift	7.13.2	M(*), M-P	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: - RecordingEvent - recordingState - playPosition - playSpeed	Broadcast-related
Access to DVB-SI EIT p/f	7.13.3	M		Broadcast-related
Extensions to video/broadcast for playback of selected components	7.13.4	M		Broadcast-related
Extensions to video/broadcast for parental ratings errors	7.13.5	M		Broadcast-related
Extensions to video/broadcast for DRM rights errors	7.13.6	M-C	Mandatory if the terminal supports CI+.	
Extensions to video/broadcast for channel scan	7.13.7	M	Access to the currentChannel property by broadcast-independent applications shall return null.	Broadcast-related

Section, sub-section	Reference in DAE [1]	Status in Hybrid Broadcast Broadband TV	Notes	Security
Extensions to video/broadcast for creating Channel lists from SD&S fragments	7.13.8	NI		
ChannelConfig object	7.13.9	M(*)	The channelList property shall be supported. Other properties, methods and events are not included.	Broadcast-related
ChannelList class	7.13.10	M(*)	The getChannelBySourceID() method is not included.	Broadcast-related
Channel class	7.13.11	M(*)	The following properties shall be supported: - channelType - ccid - dsd - onid - tsid - sid - name All other properties and methods are not included.	Broadcast-related
Favourite lists	7.13.12, 7.13.13	NI		
The CEA 2014 A/V Control embedded object				
State diagram for A/V control objects	7.14.1.1	M		None
Using an A/V control object to play streaming content	7.14.1.2	M		None
Using an A/V control object to play downloaded content	7.14.1.3	M-D		Trusted
Using an A/V control object to play recorded content	7.14.1.4	M-P		Trusted
Extensions to A/V object for playback through Content-Access Streaming Descriptor	7.14.2	O-M	The description of how a particular DRM technology integrates with the present document may make this mandatory.	None
Extensions to AV object for trickmodes	7.14.3	M(*)	Only the onPlayPositionChanged property and event are required.	None
Extensions to A/V object for playback of selected components	7.14.4	M		None
Extensions to A/V object for parental rating errors	7.14.5	O-M	The description of how a particular DRM technology integrates with the present document may make this mandatory	None
Extensions to A/V object for DRM rights errors	7.14.6	M-M		none
Extensions to A/V object for playing media objects	7.14.7	M-D, M-P	Shall be supported if either the download or PVR features are supported.	Trusted
Extensions to A/V object for UI feedback of buffering A/V content	7.14.8	NI		
DOM 2 events for A/V object	7.14.9	M		None
Playback of memory audio	7.14.10	M		None
Miscellaneous APIs				
application/oipfMDTF embedded object	7.15.1	NI		
application/oipfStatusView embedded object	7.15.2	NI		
application/oipfCapabilities embedded object	7.15.3	M	The hasCapability() method shall be supported with the profile names being the Hybrid Broadcast Broadband TV option strings as defined in clause 10.2.4.	None

Section, sub-section	Reference in DAE [1]	Status in Hybrid Broadcast Broadband TV	Notes	Security
The Navigator class	7.15.4	M		None
Debug Print API	7.15.5	M		None
The StringCollection class	7.16.1	M		None
The Programme Class				
Basics	7.16.2.1, 7.16.2.2	M(*)	<p>The following properties are required:</p> <ul style="list-style-type: none"> - name - programmeID - programmeIDType - description - longDescription - startTime - duration - channelID - parentalRatings <p>All other properties and methods are not included.</p> <p>The constants defined in clause 7.16.2.1 shall be supported however support for CRIDs is outside the scope of the present document.</p> <p>The following method is required for Programme objects returned by the programmes property of the video/broadcast object:</p> <ul style="list-style-type: none"> - getSIDescriptors 	Broadcast-related
Metadata extensions to Programme	7.16.2.3	NI		
DVB-SI extensions to Programme	7.16.2.4	NI		
Recording extensions to Programme	7.16.2.5	NI		
The ProgrammeCollection class	7.16.3	M		Broadcast-related
The DisclInfo class	7.16.4	NI		
Extensions for playback of selected media components	7.16.5	M		
System integration aspects				
HTTP User-Agent header	8.1.1	NI	See clause 7.3.2.4.	
Mapping from APIs to Protocols				
Network (Common to Managed and Unmanaged Services)	8.2.1	M-D		
OITF-IG Interface (Managed Services Only)	8.2.2	NI		
Network (Unmanaged Services only)	8.2.3	M(*)	Clause 8.2.3.1 shall be supported for the HTTP protocol only. Clause 8.2.3.2 is not included.	
URI Schemes and their usage	8.3	M	The http, https and dvb URL schemes shall be supported as defined in this clause.	
Mapping from APIs to Content Formats				
Character Conversion	8.4.1	M		
AVComponent	8.4.2	M(*)	Only for properties that are required by the present document	
Channel	8.4.3	M(*)	Only the requirements about channels of type ID_DVB_* applies and only then for properties that are required by the present document.	

Section, sub-section	Reference in DAE [1]	Status in Hybrid Broadcast Broadband TV	Notes	Security
Programme, ScheduledRecording, Recording and Download	8.4.4	M(*)	Only for properties that are required by the present document.	
Exposing Audio Description Streams as AVComponent objects	8.4.5	M(*)	This only applies to the extent that the terminal supports audio description.	
Capabilities				
Minimum DAE capability requirements	9.1	NI	See clause 10.2.1 in the present document.	
SSL/TLS Requirements	9.1.1	M(*)	9.1.1.1 and 9.1.1.2 are required. 9.1.1.3 is replaced by clause 11.2 of the present document.	
Default UI profiles	9.2	NI		
CEA-2014 capability negotiation and extensions				
Tuner/broadcast capability indication	9.3.1	M		
Broadcasted content over IP capability indication	9.3.2	NI		
PVR capability indication	9.3.3	M-P		
Download Cod capability indication	9.3.4	M-D		
Parental ratings	9.3.5	M		
Extended A/V API support	9.3.6	M		
OITF Metadata API support	9.3.7	M		
OITF Configuration API support	9.3.8	M		
IMS API Support	9.3.9	NI		
DRM capability indication	9.3.10	M		
Media profile capability indication	9.3.11	M		
Remote diagnostics support	9.3.12	NI		
SVG	9.3.13	NI		
Third party notification support	9.3.14	NI		
Multicast Delivery Terminating Function support	9.3.15	NI		
Other capability extensions	9.3.16	M		
Security				
OITF requirements	10.1.1	NI		
Server requirements	10.1.2	NI		
Specific security requirements for privileged Javascript APIs	10.1.3	NI		
Permission names	10.1.4	NI		
Loading documents from different domains	10.1.5	M		
User Authentication	10.2	M(*)	HTTP Basic and Digest Authentication as defined in clause 5.4.1 of the OIPF CSP specification [5] shall be supported. Other forms of user authentication from clause 5 of the OIPF CSP specification are not included.	
CE-HTML Profiling				
5.2 Additional value	B	NI		
5.2 name	B	NI		
5.2 new UI profiles	B	NI		
5.2 video and audio profile elements	B	NI		
5.2 element pointer	B	NI		
5.3a - 5 Content-Encoding Header	B	M		
5.3a - 12 User-Agent	B	NI		
5.4 CSS3 image rotation	B	M		
5.4 W3C obsolete DOM 2 features	B	M		
5.4 Compatibility with CEA-2027-A	B	M		
5.4 Window scripting object changes	B	M(*)	See clause A.2.8.	None
5.4 Omit Window.download()	B	M		
5.4 HTML5 cross document	B	NI		

Section, sub-section	Reference in DAE [1]	Status in Hybrid Broadcast Broadband TV	Notes	Security
messaging				
5.4 Keypress events	B	M		None
5.4 change to 5.4.a.3.a	B	M		None
5.4 change to 5.4.a.3.c	B	M		None
5.4 change to 5.4.a.3.d	B	M		None
5.4 change to 5.4.a.3.e	B	M		None
5.4 change to 5.4.a.6.b	B	M		
5.4 change to 5.4.a.7	B	M		None
5.4 change to 5.4.1.f	B	M		None
5.4 change to 5.4.1.m	B	M		None
5.4 add requirement 5.4.1.p	B	M		
5.4 add requirement 5.4.1.q	B	M		
5.4 add requirement 5.4.1.r	B	M		None
5.4 add requirement 5.4.1.s	B	M		None
5.6.2 section is optional	B	M		
5.6.2 extended requirement 5.6.2.a	B	NI		
5.7 addition to 5.7.1.f	B	M		
Annex C	B	M		
Annex F additional KeyCode	B	M		None
Annex G onkeypress events	B	M		None
Annex H image rotation CSS property not supported	B	M		
Annex H clarification for CSS font property	B	M		
Annex I onkeypress intrinsic event handler	B	M		None
Annex I charCode attribute support	B	NI		None
Annex I DOM 2 Event clarification	B	M		None
Annex I Full support except interfaces	B	M		None
Annex I added DocumentView interface	B	M		None
Content Access Descriptor Syntax and Semantics				
Content Access Download Descriptor Format	E.1	M-D		
Content Access Streaming Descriptor Format	E.2	O-M	The description of how a particular DRM technology integrates with the present document may make this mandatory.	
Abstract Content Access Descriptor Format	E.3	M-D, O-M	Shall be supported if the download features is supported. The description of how a particular DRM technology integrates with the present document may make this mandatory.	
Capability Extensions Schema	F	M		
Client Channel Listing Format	G	NI		
Display Model	H	M		

Table A.2: Key to security column

Security	Description
none	All applications shall have access to the referenced API.
trusted	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 <code>SecurityError</code> (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 <code>HTMLObjectElements</code> . Security restrictions are enforced only when the application attempts to access properties or execute functions on the objects.
broadcast-related	Broadcast-related applications shall have access to the referenced API regardless of whether they are trusted or not. If other applications or web pages try to use this API, the terminal shall throw an error with the name property set to <code>SecurityError</code> (see clause 10.1.1 of the OIPF DAE specification [1]). Note that for embedded objects, untrusted broadcast-independent applications may acquire instances of them without restrictions, either through the object factory or by using <code>HTMLObjectElements</code> . Security restrictions are enforced only when the application attempts to access properties or execute functions on the objects.
n/a (for optional APIs)	The security level for optional APIs is the manufacturer's decision. If such APIs are provided, they should have at least a security level of "trusted". Further restrictions may be added.

Table A.3: Key to status column

Status	Meaning
M	Mandatory.
M-C	Mandatory if CI+ is supported for protected content via broadcast. Support of the related section/sub-section in table A.1 is not expected if CI+ support is not indicated according to clause 10.2.4.
M-D	Mandatory if the download feature supported otherwise not included.
M-M	Mandatory if the DRM feature is supported otherwise not included. Support of the related section/sub-section in table A.1 is not expected if the support of the DRM feature is not indicated according to clause 10.2.4. See note 2.
O-M	Optional in the present document but may be made mandatory by the definition of how a particular DRM solution integrates with the present document.
M-P	Mandatory if the PVR feature is supported otherwise not included.
NI	Not included.
NOTE 1: Any of the above may be post-fixed with (*) where only some parts of the section or sub-section are required in the present document.	
NOTE 2: A device supporting CI+ is not expected to support all the APIs required for the DRM feature.	

A.2 Modifications, extensions and clarifications

A.2.1 Resource management

In clause 4.4.5 of the OIPF DAE specification [1], the statement that, "If insufficient resources are available to present the media, the attempt to play the media shall fail except for" shall have an extra exception in addition to those listed in that document - suspension of access to broadcast resources (see clause 6.2.2.7 of the present document).

A.2.2 Extensions to the ApplicationPrivateData class

This class shall be extended with the following additional property.

<p><code>readonly Channel currentChannel</code></p> <p>For a broadcast-related application, the value of the property contains the channel whose AIT is currently controlling the lifecycle of this application.</p> <p>If no channel is being presented, or if the application is not broadcast-related, the value of this property shall be <code>null</code>.</p>
--

A.2.3 Extensions to the oipfCapabilities embedded object

The former contents of this clause are now included in clause 7.15.3 of the OIPF DAE specification [1].

A.2.4 Extensions to the video/broadcast object

A.2.4.1 State machine and related changes

This clause describes a set of changes to the state machine for the video/broadcast object defined in clause 7.13.1.1 of the OIPF DAE specification [1].

- Calling the `setChannel()` method from any state of the video/broadcast object with a null argument shall cause the application to transition to a broadcast-independent application (as described in clause 6.2.2.6). This is in addition to what is required by OIPF - e.g. causing the video/broadcast object to transition to the unrealized state and releasing any resources used for decoding video and/or audio. Hence the `setChannel(null)` and `release()` methods do not have the same behaviour in the present document.
- Suspension of access to broadcast resources as defined in clause 6.2.2.7 of the present document shall be treated as a transient error.

A.2.4.2 Access to the video/broadcast object

The following rules and clarifications shall apply to the video/broadcast object.

Broadcast-related applications shall have full access to the video/broadcast object. If a new broadcast service is selected then this may result in the broadcast-related application being killed as defined in clause 6.2.2.4. Access to MPEG programs which are not broadcast services and which do not contain an AIT will not have these consequences.

Broadcast-independent applications shall be able to use the video/broadcast object as follows.

- The following properties and methods shall have no restrictions: `createChannelObject`, `onChannelChangeSucceeded`, `onChannelChangeError`, `onPlayStateChange`, `addEventListener`, `removeEventListener`, `width` and `height`.
- The `setChannel` method shall trigger the behaviours defined in clause 6.2. If the method is used to select a broadcast service then this may result in the application becoming a broadcast-related application. If the `setChannel` method is used to access an MPEG program which is not a broadcast service and which does not contain an AIT, then there are no restrictions and no consequences for the application lifecycle.
- The following methods shall always throw a "Security Error" (as defined in clause 10.1.1 of the OIPF DAE specification [1]): `getChannelConfig`, `bindToCurrentChannel`, `prevChannel` and `nextChannel`.
- The following methods shall have no effect: `setFullScreen`, `release`, and `stop`.
- The object shall always be in the unrealized or connecting states unless connected to an MPEG program which is not a broadcast service and which does not contain an AIT.

Terminals shall only support one active instance of a video/broadcast object at any time. "Active" means here that the video/broadcast object is either in the `connecting` or the `presenting` state. Trying to activate an instance of a video/broadcast object (through a call to `bindToCurrentChannel()` or a `setChannel()` call) while another instance is already active shall fail and result in an error returned to the application through a `ChannelChangeError` event.

A.2.4.3 Extensions to the Configuration class for time-shift

The following property is added to the Configuration class.

<code>readonly Boolean timeShiftSynchronized</code>
Returns a boolean indicating if the terminal is capable of maintaining synchronization between applications and A/V components during time-shift. A definition of synchronization between applications and A/V components can be found in clause 6.2.2.4.

A.2.5 Extensions to the AV Control Object

The following method shall be added to the AV Control embedded object.

<code>Boolean queue(String url)</code>			
Description	<p>Queue the media referred to by <code>url</code> for playback after the current media item has finished playing. If a media item is already queued, <code>url</code> will not be queued for playback and this method will return false. If the item is queued successfully, this method returns true. If no media is currently playing, the queued item will be played immediately.</p> <p>If <code>url</code> is null, any currently queued item will be removed from the queue and this method will return true.</p> <p>If an AV Control object is an audio object (as defined by clause 5.7.1.b.1 of CEA-2014 [i.1]) then queued media items shall only contain audio. If an AV Control object is a video object (as defined by clause 5.7.1.b.2 of CEA-2014 [i.1]) then queued media items shall always contain video and may also contain audio and other media components.</p> <p>When the current media item has finished playing, the AV Control object shall transition to the finished state, update the value of the <code>data</code> property with the URL of the queued media item and automatically start playback of the queued media item. The AV Control object may transition to the connecting or buffering states before entering the playing state when the queued media item is being presented. Implementations may pre-buffer data from the queued URL before the current media item has finished playing in order to reduce the delay between items.</p> <p>Play speed is not affected by transitioning between the current and queued media item.</p> <p>To avoid race conditions when queueing multiple items for playback, applications should wait for the currently queued item to begin playback before queueing subsequent items, e.g. by queueing the subsequent item when the AV Control object transitions to the connecting, buffering or playing state for the currently queued item.</p>		
Arguments	<table border="1"> <tr> <td><code>url</code></td> <td>The media item to be queued, or null to remove the currently-queued item.</td> </tr> </table>	<code>url</code>	The media item to be queued, or null to remove the currently-queued item.
<code>url</code>	The media item to be queued, or null to remove the currently-queued item.		

Calling `stop()`, modifying the `data` property or entering the error state shall cause any queued media item to be discarded.

Play control keys (OK, play, stop, pause, fast forward, fast rewind and other trick play keys) shall not be handled by the AV Control object and no action shall be taken by the terminal for these keys when they have been requested by an application. DOM 2 events shall be generated for these keys whether the AV Control object is focused or not.

The timing of automatic transitions from the error state to the stopped state is implementation dependent; applications should not rely on the AV Control object remaining in the error state after an error has occurred and should listen for play state change events in order to detect errors.

If the AVControl object's `play()` method returns true then at least one play state change event shall be generated

The error property shall be available in the stopped state. After an automatic transition from the error state to the stopped state, the value of the error property shall be preserved.

The following value shall be added to the list of valid values for the error property:

- undefined - no error has occurred;
- 7 - content blocked due to parental control.

A.2.6 XHTML Profile

A.2.6.1 General

The XHTML profile defined in sections 6.1 and 6.2 of the OIPF DAE specification [1] shall apply with the following restrictions and extensions:

- Support for [Req. 5.4.1.o] of CEA2014-A (use of the `accesskey` attribute for standardized key-codes) is not included.
- The `video/local` object is not included.

A.2.6.2 MIME type and DOCTYPE

All XHTML documents of an Hybrid Broadcast Broadband TV application shall include either:

- The Strict XHTML doctype (for documents that are conformant with the subset of the XHTML 1.0 Strict DTD defined in the present document).
- The Transitional XHTML doctype (for documents that are conformant with the subset of the XHTML 1.0 Transitional DTD defined in the present document).
- The following "doctype" declaration:


```
<!DOCTYPE html PUBLIC "-//HbbTV//1.1.1//EN" "http://www.hbbtv.org/dtd/HbbTV-1.1.1.dtd">
```
- The following "doctype" declaration:


```
<!DOCTYPE html PUBLIC "-//HbbTV//1.2.1//EN" "http://www.hbbtv.org/dtd/HbbTV-1.2.1.dtd">
```

Terminals implementing the 1.1.1 version of the present document may reject documents with the 1.2.1 doctype. Hence this doctype shall only be used for applications which are so dependent on features in the present document that it would be meaningless for a 1.1.1 terminal to even start them.

It shall be followed by an `<html>` tag declaration including the `xmlns` attribute as follows:

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

Where a browser supports both a "Standards Mode" and a "Quirks Mode" for rendering documents, any documents of an Hybrid Broadcast Broadband TV application with the doctypes specified above shall be rendered in "Standards Mode" regardless of the presence of an XML declaration before the doctype declaration.

All XHTML documents of an Hybrid Broadcast Broadband TV application shall be served with the MIME content type `"application/vnd.hbbtv.xhtml+xml"`. All pages loaded from a carousel shall be handled as if they had this MIME type. When loading an Hybrid Broadcast Broadband TV document, a terminal shall not use the suffix from the filename to determine the MIME type.

Terminals are not required to load or run documents which are served with a MIME type other than `"application/vnd.hbbtv.xhtml+xml"` or which do not include one of the doctype declarations defined above.

A.2.6.3 Use of iframe Elements

This clause is replaced by clause 10.1.5 of the OIPF DAE specification [1].

A.2.6.4 Browser History

The terminal should not offer a history UI for Hybrid Broadcast Broadband TV applications.

The behaviour of the history mechanism when an Hybrid Broadcast Broadband TV application transitions between broadcast-independent and broadcast-related (or vice-versa) is outside the scope of the present document.

Implementations may record and reproduce these transitions when the history mechanism is used but are not required to do so.

A.2.7 CSS profile

This clause is replaced by requirements in annex B of the OIPF DAE specification [1].

A.2.8 DOM profile

A.2.8.1 The Window object

The window object shall be supported as defined in annex B of the OIPF DAE specification [1] except as follows. The following properties shall be supported on the window object:

`document`, `frames`, `history`, `innerHeight`, `innerWidth`, `location`, `name`, `navigator`, `oipfObjectFactory`, `onkeypress`, `onkeydown`, `onkeyup`, `parent`, `self`, `top`, `window`, `XMLHttpRequest`, `onblur`, `onfocus`, `frameElement`

The following methods shall be supported on the window object:

`close()`, `debug()`, `setTimeout()`, `setInterval()`, `clearTimeout()`, `clearInterval()`, `addEventListener()`, `removeEventListener()`

All other methods and properties are not included.

Annex B (normative): Support for protected content delivered via broadband

B.1 Introduction

When content protection is being used, the type of content protection in use shall be signalled:

- as defined in clause 9.3.10 of the OIPF DAE specification [1] and in table 10 of the OIPF Metadata specification [18];
- using DVB-CA identifier codepoints (CA_System_ID) allocated as usual by the DVB Project and found in TS 101 162 [19] for the DRMSystemID.

Some issues that need to be considered when defining how a particular content protection technology is integrated with implementations of the present document are described in annex F.

B.2 Common Encryption for ISOBMFF

Support for MPEG common encryption as defined in CENC [30] is optional in the present document. If it is supported then the following requirements shall apply.

B.2.1 Key Management for On Demand Content

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

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

B.2.2 Key Management for Live Content

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

NOTE 1: Periods are typically used for separate programs in a live broadcast.

NOTE 2: In cases where it is desired to use different keys for different Representations or media components, 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 alive 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.

B.2.3 Encryption mode

Media data shall be encrypted using AES 128-bit in CTR mode (AES-CTR) as defined in section 9 of CENC [30].

B.2.4 Usage of ISOBMFF boxes

This clause specifies relevant parameters of existing ISOBMFF boxes used with CENC [30].

B.2.4.1 'pssh' box

An ISOBMFF file may contain multiple Protection System Specific Header ('pssh') boxes (as defined in CENC [30]). 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.

B.2.5 Extensions to ISOBMFF boxes

B.2.5.1 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 shall be such that they point to data that is located before the sample media data to which this sample auxiliary information corresponds.

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 [30] 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 C (informative): Support for analogue broadcasting networks

C.1 Scope

The main target of the Hybrid Broadcast Broadband TV specification is to combine services delivered via a DVB compliant broadcast network and a broadband connection to the Internet. Many of the conceptual and technical aspects of Hybrid Broadcast Broadband TV, however, are also applicable to a combination of an analogue Broadcast network and a broadband Internet connection. Analogue TV distribution may for some years still be of relevance for some markets.

If a terminal includes an analogue front end, the Hybrid Broadcast Broadband TV concept may be applied to analogue channels as described in this annex. If the Hybrid Broadcast Broadband TV concept is not applied to analogue channels then they would be treated in the same way as DVB channels without an AIT.

C.2 AIT retrieval and monitoring

As the AIT cannot be provided within the analogue broadcast channel, it has to be retrieved via the Internet connection. When tuning to an analogue service the hybrid terminal can send an http request to a server hosting AIT information as following.

```
http:// [AIT_server]/service?CNI=xxx
http:// [AIT_server]/service?name=xxx
```

This request will return the AIT of the corresponding service encoded in XML format as defined in TS 102 809 [3]. The AIT is contained in a single application discovery record.

The IP address or the base URL of the AIT server may be market or manufacturer specific. It could be part of the default settings of the terminal and may allow for changes by the user.

For the identification of the service the CNI code as registered in TS 101 231 [i.3] should be used. As an alternative the name of the service may be used.

AIT monitoring while being tuned to a specific service can be done by repeating the http requests defined above. The xml document that contains the AIT carries a version attribute within the <ServiceDiscovery> element. If present the version attribute is used in the request as follows:

```
http:// [AIT_server]/service?CNI=xxx&version=YY
http:// [AIT_server]/service?name=xxx&version=YY
```

where YY are two hexadecimal digits. If the recent version on the server is the same as in the request the server returns the HTTP status code 204 with no message body.

The repetition rate should not be more frequent than once per 30 seconds.

C.3 Tuning to a new channel

The video/broadcast embedded object defined in the OIPF DAE specification [1] can be used to determine available analogue broadcast services and to tune between them as described in this clause.

An analogue broadcast service is represented by a channel object with an idType of ID_ANALOG including the properties `cni` and/or `name`. The `cni` property contains the CNI of the service when it is available in the broadcast signal. The `name` property is available when the CNI is not broadcast. For CNI and name see clause C.2.

The channel lineup of the Hybrid Broadcast Broadband TV terminal is available to the application in order to be able to retrieve channel objects for a CNI or name.

The `currentChannel` property on the video/broadcast object and the `ApplicationPrivateData.currentChannel` property returns the channel object for the analogue service currently presented.

C.4 Other aspects

EIT access, application transport with DSM-CC, stream events, etc are not available on analogue channels. Method calls related to these features cause exceptions with a message "not supported". Properties related to these features have the value `undefined`.

Annex D (informative): Server root certificate selection policy

D.1 Introduction

This informative annex describes the policy that is adopted for the selection of root certificates for inclusion in terminals compliant with the present document. A list of such certificates is published at <http://www.hbbtv.org/spec/certificates.html>.

D.2 Background

There are over 150 root certificates in web browsers at the time of publication.

- This list changes frequently over time.
- The larger the list of root certificates the more likely it is to change.

The security of TLS against man-in-the-middle attacks is dependent on the weakest root certificate trusted by a terminal.

The security of various key lengths changes with time as computing power increases. Specifically 1 024 bit RSA keys are no longer recommended for use.

Service providers need to know which root certificates are trusted by terminals to achieve interoperability. Service providers are often not in control of the servers delivering their content (e.g. delivery via a CDN).

Service providers may also wish to make use of third party web services that are not under their control.

Maintaining an independent list of root certificates that are validated requires significant resources.

D.3 Policy

The Mozilla list of approved root certificates has been selected as the authoritative source for the mandatory and optional list of root certificates for inclusion in terminals compliant with the present document. This was chosen because:

- The approved root certificate list is publicly available.
- The process for inclusion in the list is open.
- Anyone can take part in the acceptance process.
- The acceptance process itself happens in public.
- Metadata is provided to differentiate root certificates for web server authentication, e-mail and code signing.
- The procedure for requesting a root certificate for inclusion in the list requires a test website be provided which uses that certificate.

The Mozilla list of approved root certificates is published on their website at <http://www.mozilla.org/projects/security/certs/>. Each certificate marked as approved for web server authentication is automatically an optional root certificate as specified in clause 11.2.

The present document will rely upon the Mozilla list for verifying the trustworthiness of Certificate Authorities.

A list of root certificates that are mandatory will be maintained which will be a subset of the certificates specified above.

- The list will be updated periodically.
- The list will only include certificates that use algorithms mandated by clause 7.3.2.3.
- The mandatory list of certificates will be determined based on the requirements of service providers and the Certificate Authorities that are in widespread use.
- The list will be compiled relying upon published statistics to determine how widespread a Certificate Authority is.
- Certificate Authorities may be excluded from the mandatory list if they impose requirements that are deemed unreasonable.
- A revision history of changes to the mandatory list will be maintained and published.

This policy is subject to change.

Annex E (normative): Profiles of MPEG DASH

E.1 Introduction (informative)

This annex starts from MPEG DASH [29] 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 ISO BMFF 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.

E.2 Requirements relating to the MPD

E.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 ISO BMFF 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 [29]) (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 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. Additionally:

- the size of a MPD shall not exceed 100 kbytes, and
- the content referenced by the profilespecific MPD shall only be encoded using the audio and video codecs defined in clause 7.3.1 of the present document.

E.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 shall 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 shall be labelled with a Role@value of "main" to allow the terminal to identify the default adaptation set. There shall be at least one video Adaptation Set per Period in an MPD.

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 E.1 defines these values for the present document:

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

Parameter	Value
N _{per}	32
N _{adset}	16
N _{rep}	16

E.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.

E.2.4 Role Related Requirements

The MPD shall adopt the DASH role scheme (urn:mpeg:dash:role:2011) as defined in MPEG-DASH [29] clause 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. There is no requirement for a terminal to 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 using the `Role` and `Accessibility` descriptors as defined in the following table. Furthermore for receiver mix AD the associated audio stream shall use `dependencyId` to point out the dependency to the main representation and hence also point out that the associated audio stream shall not be provided as a representation on its own. Terminals shall ignore audio streams with other `Role` and `Accessibility` descriptor attributes that they do not understand.

Table E.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 as defined in [34]
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>
```

E.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 TS 102 366 [15] 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"/>
```

E.2.6 Content protection signalling

Content protection signalling is stored within the MPD inside ContentProtection elements (see DASH [29] clause 5.8.4.1). The MPD shall contain a ContentProtection element for each content protection system used. MPD URI definitions for ContentProtection elements shall conform to DASH [29] clause 5.8.5.2 "Content protection", whereby the method of the third scheme (in the third bullet text) in DASH [29] clause 5.8.5.2 shall be applied".

E.3 Restrictions on Content

E.3.1 Restrictions on File Format

E.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 [31]:

- 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 ISO 14496-12 [31]) in order to maintain audio visual presentation synchronization.

E.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 box shall 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 and Picture Parameter Sets) used for Representations in the Adaptation Sets.

Initialization segments being common means that all representations in an adaptation set will have identically the same 'std' box. There will be one entry in the 'std' box for each representation. Representations encoded with different "parameters" will use the sample_description_index in the Track Fragment Header to identify which of the sample entries in the 'std' box is applicable to them. Each segment shall consist 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 E.4.2. Adaptation Sets may include Representations which can only be reached by transitions other than those which terminals are required to support.

E.4 Requirements on Terminals

E.4.1 DASH Profile Support

Terminals shall support the HbbTV ISO/BMFF Live profile. Other profiles may be supported.

E.4.2 Transitions between Representations

E.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 a).
- 2) Between Representations which differ by profile and/or level (note b).
- 3) Between Representations which differ by full-screen resolution (e.g. 1 920 × 1 080 and 720 × 576) (note b) (note c).

- 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 [14] (e.g. 1 920 × 1 080 and 1 440 × 1 080) (note b).
 - a) Transitions shall be seamless unless combined with other changes which do not have that requirement.
 - b) Transitions may include repeated frames but shall otherwise be seamless.
 - c) As defined in clause 10.2.1 of the present document, video shall be scaled, preserving the aspect ratio, such that all of the decoded video is visible within the area of the 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.

E.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. 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:

- 2) Between Representations where one is encoded with HE-AAC and the other is E-AC3.
- 3) Between Representations which differ in the number of audio channels.
- 4) Between Representations which differ in the sampling frequency.

E.4.3 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 the present document concerning persistent storage of streamed content shall also apply to content delivered as specified in this annex.

E.4.4 ISO File Format Support

Terminals shall support more than one sample entry in the 'std' box and shall support the use of the `sample_description_index` in the Track Fragment Header at the start of each segment to identify which of the sample entries is applicable to that segment.

Annex F (informative): DRM Integration

F.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.

F.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 showing in annex D of OIPF volume 5 [1].

A DRM System ID for the DRM system needs to be registered in as described in OIPF Volume 5 [1], 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 [1], 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 need to be in the format of the video profile capability strings indicating as defined in OIPF Volume 5 [1], section 9.3.11.

F.3 DRM Agent API

In the `sendDRMMessage` method (as defined in OIPF volume 5 [1], 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 [1], 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`.

F.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 [1], 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 [1], section 7.14.6). If it is generated, the values to be used for the `contentID` and the `rightsIssuerURL` parameters need to be defined.

History

Document history		
V1.1.1	June 2010	Publication
V1.2.1	November 2012	Publication