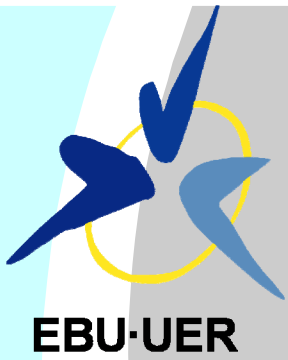
ETSI TS 102 796 V1.1.1 (2010-06)

Technical Specification

Hybrid Broadcast Broadband TV



Reference
DTS/JTC-021

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

Contents

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://webapp.etsi.org/IPR/home.asp).

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

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.

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 (V1.1): "Declarative Application Environment".

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

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

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

[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 (V1.1): "Protocols".

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

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

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

[6] Internet Streaming Media Alliance, "Implementation Specification Version 2.0, April 2005".

NOTE: Available at http://www.isma.tv.

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

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

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

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

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

[12] W3C Working Draft 19 November 2009: "XMLHTTPRequest".

NOTE: Available at http://www.w3.org/TR/XMLHttpRequest/.

[13] 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.

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

[15] ETSI TS 101 154 (V1.9.1): "Digital Video Broadcasting (DVB);Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream".

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

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

[18] W3C Working Draft 25 August 2009: "HTML5; A vocabulary and associated APIs for HTML and XHTML".

NOTE: Available at http://www.w3.org/TR/2009/WD-html5-20090825/.

[19] Open IPTV Forum Release 1 specification, volume 3 (V1.1): "Content Metadata".

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

[20] 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".

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 102 757 (V1.1.1): "Digital Video Broadcasting (DVB); Content Purchasing API".

[i.4] 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".

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 [2].

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:

AIT	Application Information Table
AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
CEA	Consumer Electronics Association
CICAM	Common Interface Conditional Access Module
CSS	Cascading Style Sheets
DAE	Declarative Application Environment
DLNA	Digital Living Network Alliance
DOM	Document Object Model
DRM	Digital Rights Management
DSM-CC	Digital Storage Media - Command and Control
DVB	Digital Video Broadcasting
EIT	Event Information Table
EIT p/f	EIT present/following
EPG	Electronic Program Guide
GIF	Graphics Interchange Format
FQDN	Fully Qualified Domain Name
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDTV	Integrated Digital TV
IP	Internet Protocol
JPEG	Joint Photographic Experts Group
MPEG	Motion Picture Experts Group
MSB	Most Significant Bit
OIPF	Open IPTV Forum
PMT	Program Map Table
PNG	Portable Network Graphics

PVR	Personal Video Recorder
RCU	Remote Control Unit
RTSP	Real Time Streaming Protocol
RTP	Real-time Transport Protocol
SSL	Secure Sockets Layer
TLS	Transport Layer Security
TV	Television
UI	User Interface
URL	Uniform Resource Locator
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 [2].
- 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.

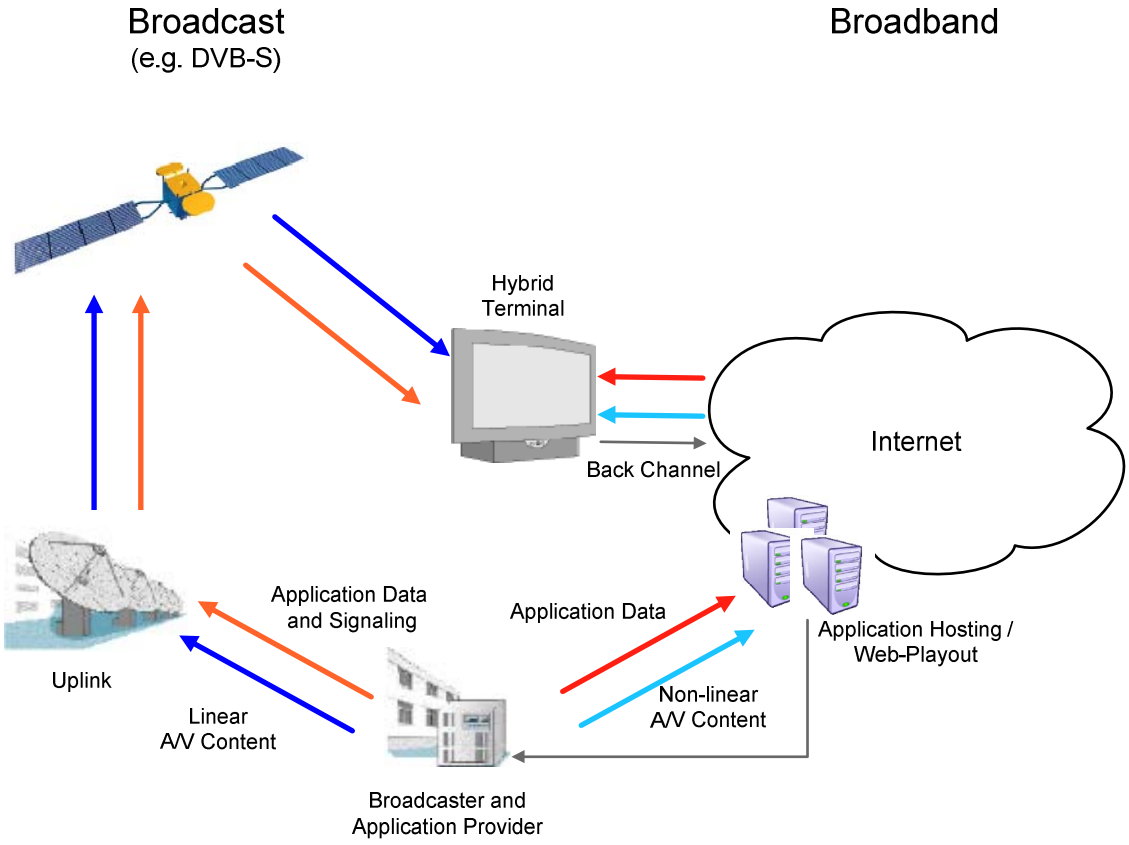


Figure 1: System Overview

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.

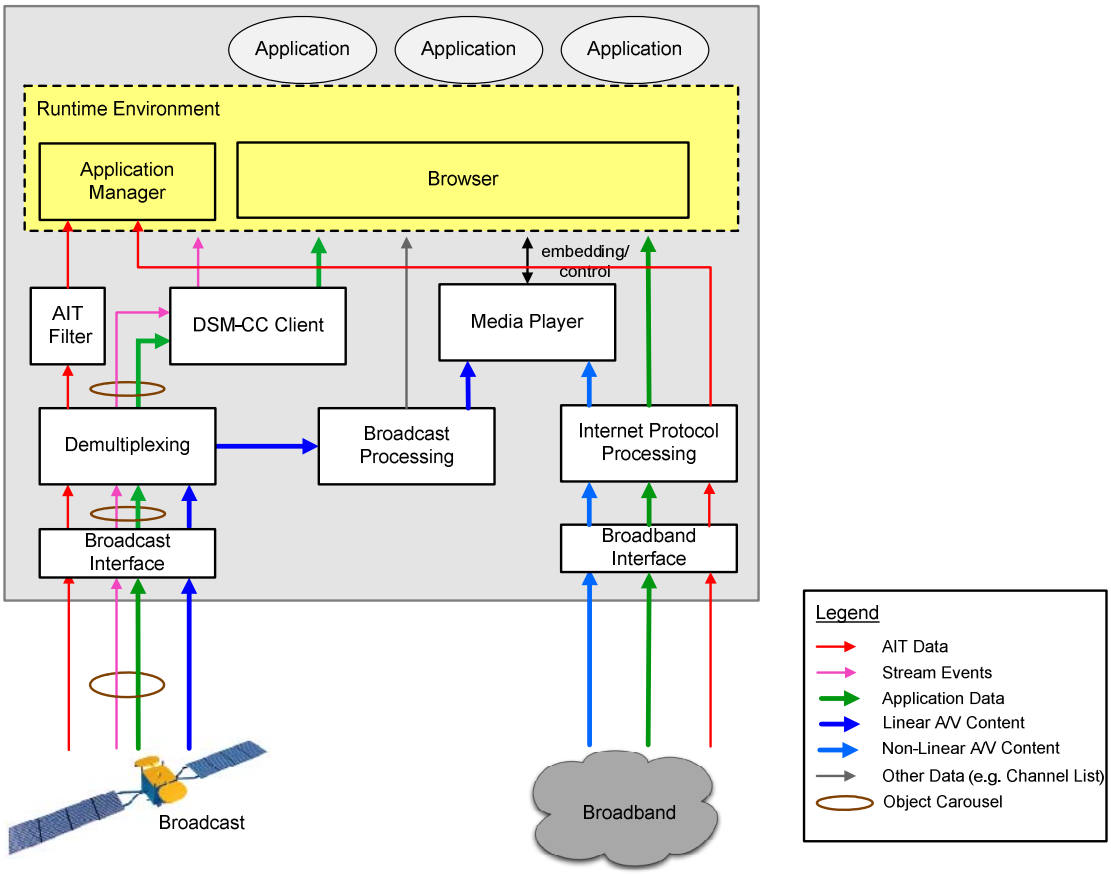


Figure 2: Functional components of a hybrid terminal

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 non-linear 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. Non-linear 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 streamed content using RTSP / RTP as defined in clauses 7.3.2.1 and 9.1.1.3 - this is referred to as the "RTSP feature".
- Support for protected content via broadband is defined in annex B.

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 - Web-based Protocol and Framework for Remote User Interface on UPnP Networks and the Internet (Web4CE), also known as CE-HTML [i.1].
- Open IPTV Forum Release 1 Volume 5 - Declarative Application Environment of the Open IPTV Forum [2].
- TS 102 809 [3] (formerly DVB Blue Book A137): "Signalling and carriage of interactive applications and services in Hybrid Broadcast Broadband environments".

The present document can be seen as a profile of these referenced documents. Figure 3 shows a graphical overview of the relationship between the profile defined here and the above mentioned specifications.

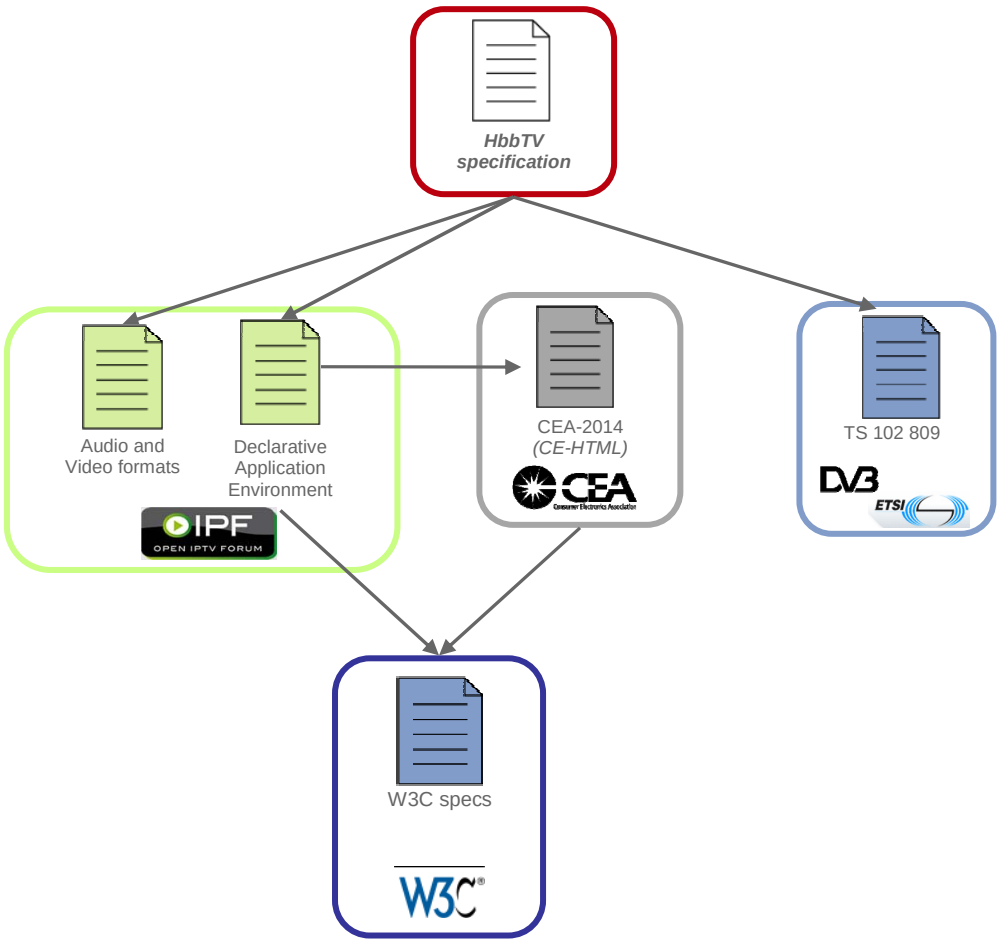


Figure 3: Specification overview

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 [2]. The present document includes some additional profiling of that CE-HTML profile. Other important components provided by the OIPF DAE specification [2] 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.

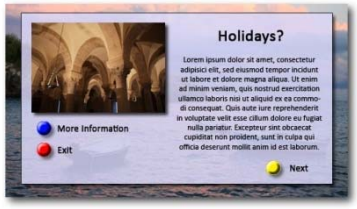
A homogenous user experience is important to enable a successful interactive platform. To ensure this, both the manufacturer and the application developer should respect the following framework and guidelines.

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.

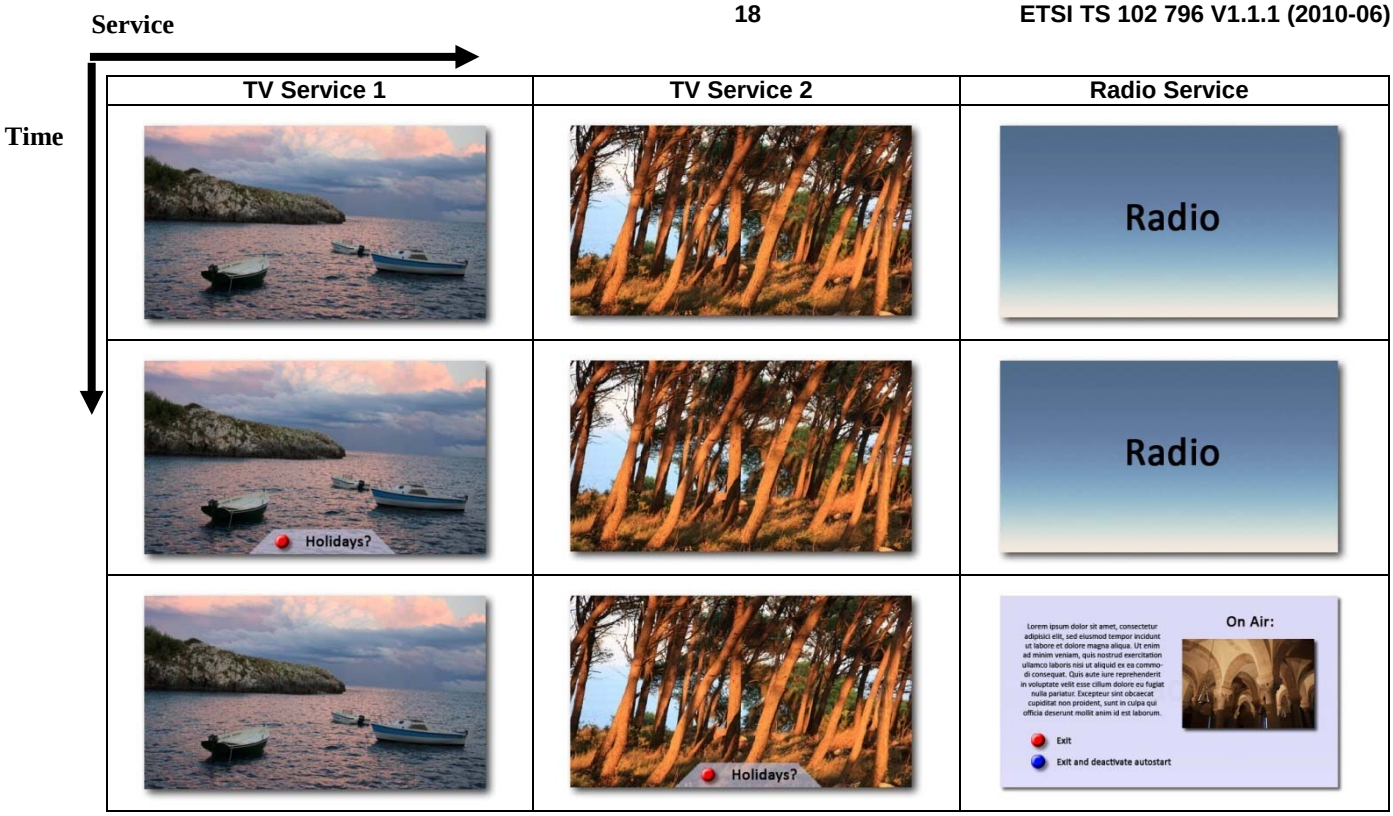


Figure 4: What might be seen across channels and through time

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 15 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. By default, the background colour of this window should be set to an opaque colour. Therefore, applications should 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.

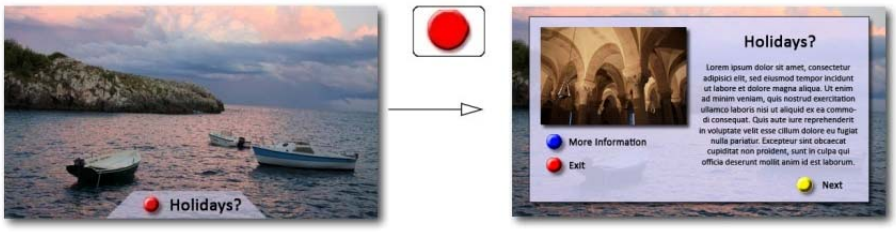


Figure 5: Service with associated broadcast-related autostart application

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.

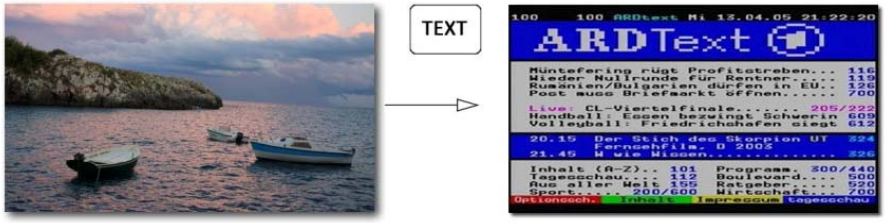


Figure 7: Service with standard teletext only

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.

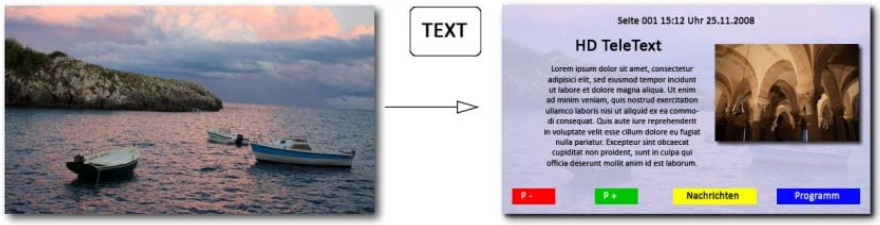


Figure 8: Service with digital teletext application only

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 13).

Case E: If neither a digital teletext application nor standard teletext is 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. 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.

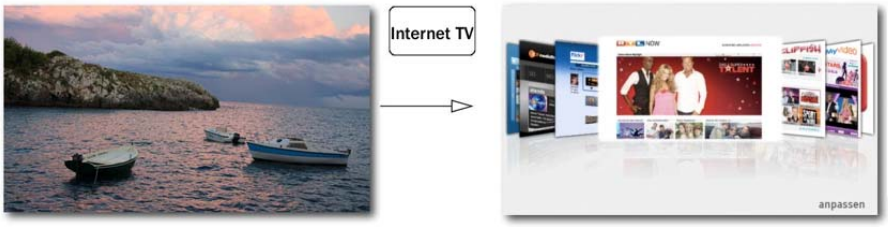


Figure 11: Internet TV Portal

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.

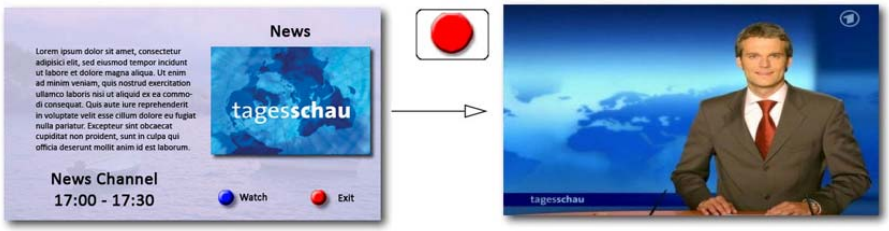


Figure 12: Application selects TV channel

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 this clause.

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:

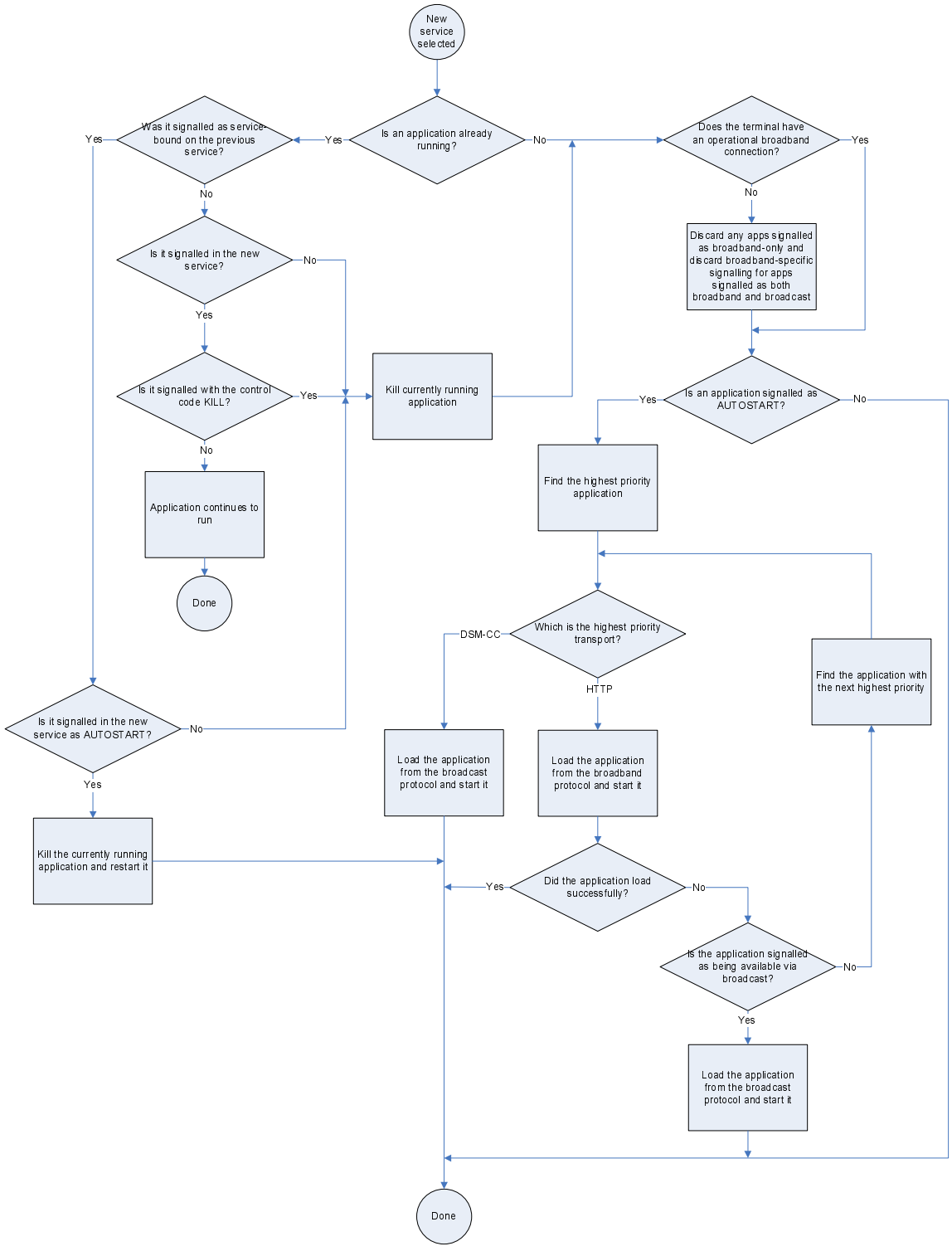
- Directly by the end-user (e.g. by using dedicated buttons on the remote control or an equivalent menu provided by the terminal).
- In response to signalling in a broadcast service (e.g. automatically starting a broadcast-related autostart application).
- By an already running application (via the JavaScript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

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.

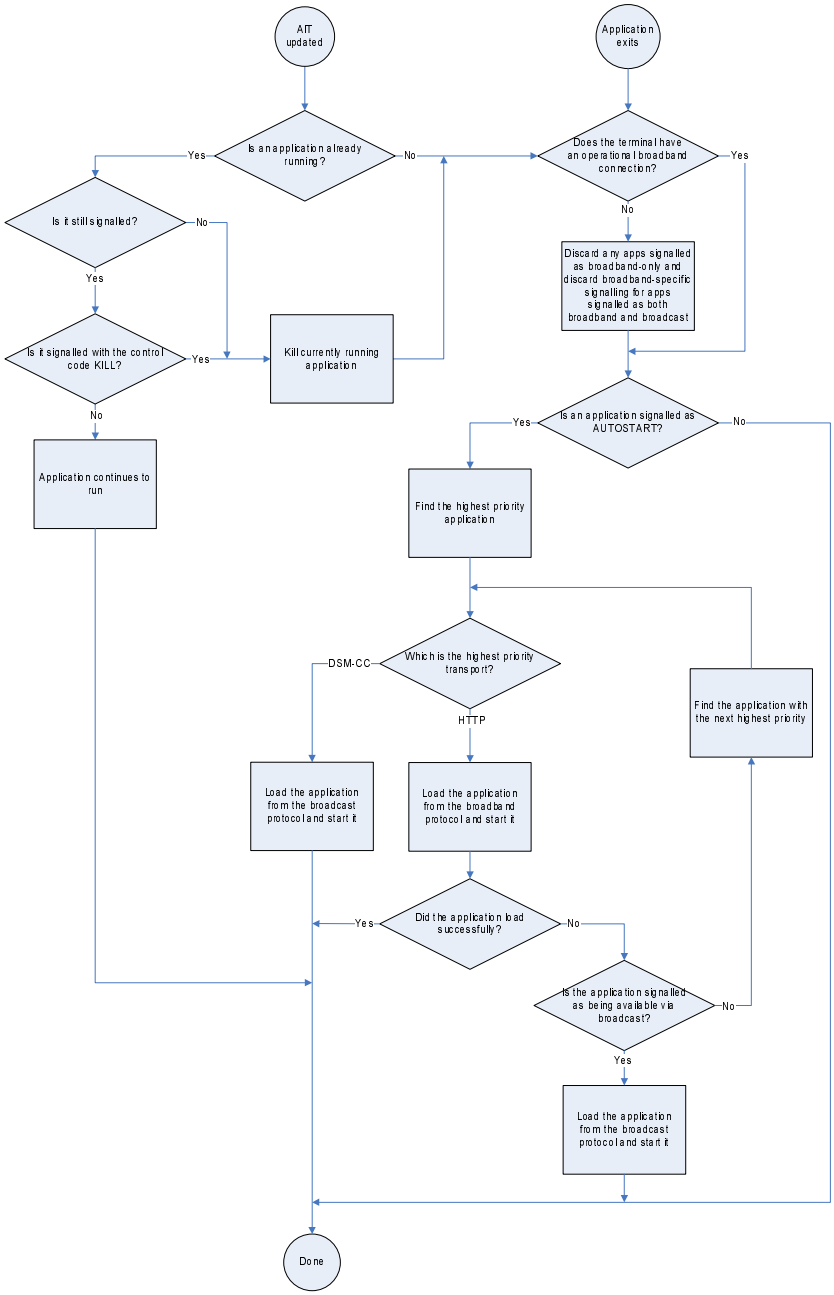
Figure 13: Behaviour when selecting a broadcast service

Figure 13 shall not apply when selecting an MPEG program which is not a broadcast service, (i.e. MPEG programs which are not listed in the SDT of the transport stream carrying them or where the transport stream carrying them does not include an SDT) and which do not include an AIT. 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.

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

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.

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: 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 [2]).
- 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.

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 display an error message unless the application was launched as autostart (e.g. following a channel selection or AIT update).

If the application cannot ultimately be loaded from either broadcast or broadband, an error shall be returned if the application was launched by a call to `createApplication()`. 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).
- From an Internet TV Portal 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 `org_id` or `app_id` and with an application domain being the fully qualified domain name of the referenced page (the first page to be loaded).

Where the URL refers to an XML AIT, the broadcast-independent application shall be created with the `org_id` and `app_id` specified in the XML AIT and an application domain derived from the contents of the XML AIT as specified 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, as described in clause 6.2.2.7.

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, as described in clause 6.2.2.7. 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 was initially referenced through an XML AIT (and hence has an `org_id` and an `app_id`).
- An application of the same `org_id` and `app_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 `org_id` and `app_id` includes a `transport_protocol_descriptor` with `protocol_id` equal to 3.
- The URL of the entry point document of the broadcast-independent application exactly matches the URL in the broadcast signalling for that `org_id` and `app_id` up to the beginning of any query or fragment string signalled.
- The page currently loaded in the broadcast-independent application is inside the application domain of the broadcast 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.

If access to broadcast resources is suspended and broadcast video is being presented by a video/broadcast object, the video/broadcast object shall transition to the `Connecting` state and a `PlayStateChangedEvent` shall be dispatched with 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, the video/broadcast object shall transition to the `Presenting` state and a `PlayStateChangedEvent` shall be dispatched.

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.

6.2.3 Application lifecycle example (informative)

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

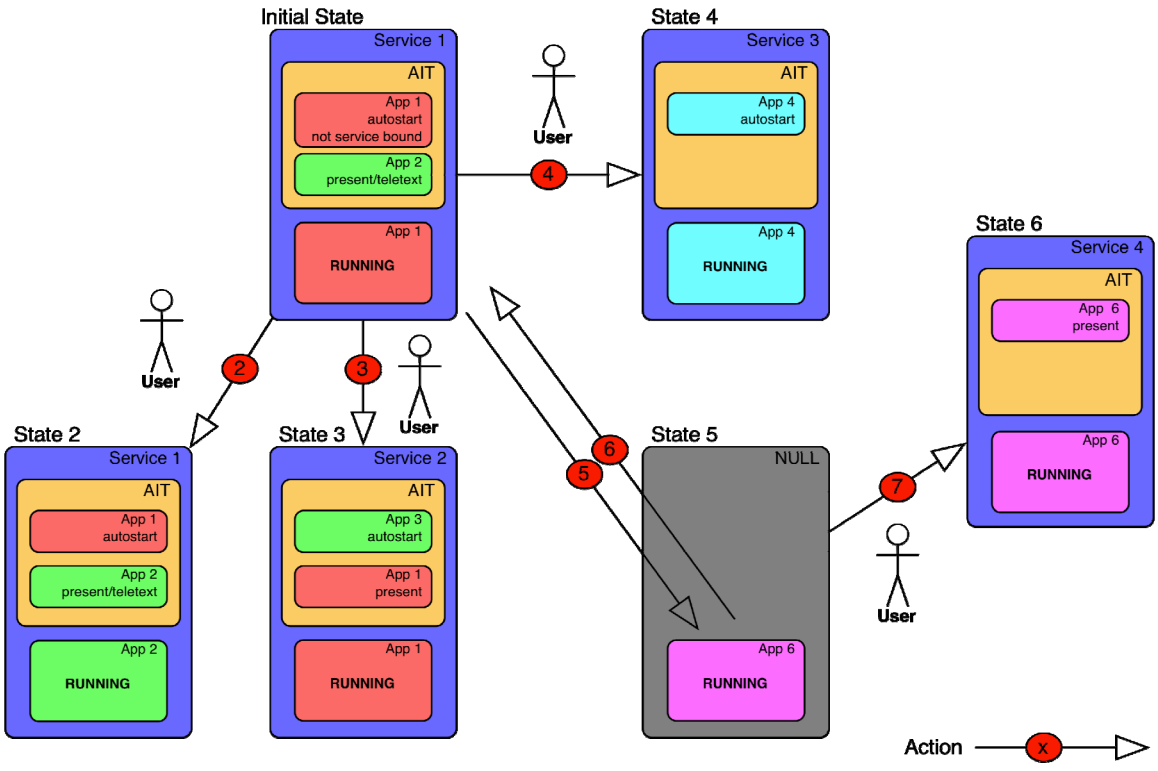


Figure 15: Application model examples

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 createApplication() to broadcast-independent application	State 5: Broadcast-independent application 6 is running. Any former presentation of service components will be stopped.
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: User selects service 4	State 6: Application 6 keeps running. Due to signalling, presentation of service components starts.

6.3 Application domain

Every application belongs to an application domain. For applications loaded via HTTP, the application domain is the "fully qualified domain name" (FQDN) of the first page of the application in the absence of an application_boundary_descriptor. For applications loaded via object carousel, the application domain is the carousel from which the first page of the application was loaded, or any identical carousels, in the absence of an application_boundary_descriptor.

Two object carousels shall be considered identical if, in the PMTs of the respective services, either of the following hold:

a) Both services are delivered within the same transport stream and:

- Both services list the boot component of the carousel on the same PID.
- The carousel_identifier_descriptor for the carousel are identical in both services (so the carousels have the same carousel_Id and boot parameters).
- All association tags used in the carousel map to the same PIDs in both services.
- The carousel is transmitted over a single path, but the services are allowed to reference the carousel via a number of routes, including deferral to a second PMT via deferred association tags.

Or:

b) The services are delivered over different transport streams and:

- The carousel_id in the carousel_identifier_descriptor is in the range of 0x100 - 0xffffffff (containing the broadcaster's OrgID in the most significant 24 MSBs of carousel_id).
- The carousel_identifier_descriptor for the carousels are identical in both services (so the carousels have the same carousel_id and boot parameters).

If an application_boundary_descriptor is present in the AIT or an applicationBoundary element is present in the XML AIT, the application domain shall also include the FQDNs or object carousels listed in the application_boundary_descriptor or applicationBoundary element. This is described in clause 5.3.8 of TS 102 809 [3].

The current application domain can be left by launching a new application by using the method createApplication() (with an arbitrary new start page) or by killing the current application and starting a new one via application signalling.

Documents loaded from outside the application domain shall be untrusted, for example documents loaded in an <iframe> element or documents loaded as a result of following a link. Following a link from outside the application domain back inside the application domain shall restore the trust level to the original trust level of the application.

For files requested with `XMLHttpRequest`,the Same-Origin Policy shall be extended using the application domain i.e. any domain in the application domain 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

7.2 Broadcast-specific format and protocols

7.2.1 System, video and audio formats

The present document does not contain any requirements for system, video and audio 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 `caching_priority_descriptor` as defined in clause B.2.2.4.2 of TS 102 809 [3] is not included.

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	`app_id`s for trusted applications (as defined in the present document) shall be in the range for signed applications (as defined in TS 102 809 [3]).

Section	Page	Status	Notes
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 1028 809 [3] and in the present document.
5.2.5 Platform profiles	17	M	For applications that only require the basic profile, the application_profile shall take the value 0x0000.The following bits can be combined for applications that require additional features: 0x0001 download feature 0x0002 PVR feature 0x0004 RTSP feature The version fields shall be set as follows: version.major = 1 version.minor = 1 version.macro = 1
5.2.6 Application visibility	18	M	
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 URL_base_bytes) of the URL as signalled in the transport_protocol_descriptor.
5.2.9 Graphics constraints	21	NI	
5.2.10 Application usage	22	M	Usage type 0x01 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	A maximum of one sub-table (i.e. using only one PID) for the Hybrid Broadcast Broadband TV application type shall be transmitted per service. All sections of the Hybrid Broadcast Broadband TV AIT sub-table shall be transmitted at least once every second.
5.3.5.1 Application signalling descriptor	33	M	
5.3.5.2 Data broadcast id descriptor	33	O	The value to be used for the data_broadcast_id field of the data_broadcast_id_descriptor for Hybrid Broadcast Broadband TV carousels shall be 0x0123. 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	NI	
5.3.5.5 Application usage descriptor	37	M	Usage type 0x01 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	M	
5.3.5.8 Graphics constraints descriptor	39	NI	
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	

Section	Page	Status	Notes
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	NI	
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.

7.2.3.2 Broadcast-independent application signalling

Broadcast-independent applications do not require any signalling. If they are signalled then this shall be done using the XML encoding of the AIT as defined in clause 5.4 of TS 102 809 [3]. The XML file shall contain an application discovery record containing exactly one application. The XML file shall be delivered with HTTP 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	Shall be "urn:hbbtv:ApplicationTypeCS:2009:HBBTV" for Hybrid Broadcast Broadband TV applications.	Mandatory. Types other than "urn:hbbtv:ApplicationTypeCS:2009:HBBTV" 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/	Mandatory.	Mandatory.
applicationBoundary/	Optional.	Mandatory.
applicationSpecificDescriptor/	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.

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 `<img>` 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 must 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.

7.3 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 and with the addition of E-AC3 as described below.

Table 8 defines a subset of the combinations of system, video and audio formats that shall be supported.

Table 8: System, video and audio formats

System Format	Video Format	Audio Format	MIME Type
TS	AVC_SD_25 AVC_HD_25	HEAAC E-AC3 (see note)	video/mpeg
MP4	AVC_SD_25 AVC_HD_25	HEAAC E-AC3 (see note)	video/mp4
NOTE: Terminals shall support E-AC3 for the broadband channel when it is supported for the broadcast channel. Otherwise it is not included.			

Table 9 defines a subset of the audio formats that shall be supported.

Table 9: Pure audio media formats

Audio Format	MIME Type
MPEG1_L3	audio/mpeg
HEAAC	audio/mp4

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 formats MPEG-2 Transport Stream and MP4 File Format shall comply with clause 4 of the OIPF Media Formats specification [2] but for E-AC3 shall comply with TS 102 366 [16]. Support for the DLNA extension "time stamped MPEG-2 transport stream" is not required.

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] with the following modifications:

- AVC baseline profile @ level 2 shall also be supported.

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 clause 4.6.8.3 of ISO/IEC 14496-3 [14], annex C of TS 101 154 [15] and clause 6.8 of TS 102 366 [16].

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 range header shall be supported in seek operations. 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.

Where unicast streaming of non transport stream based MPEG4/AVC video and MPEG/AAC audio using RTSP and RTP is supported, this shall be as defined by clauses 6, 7 and 8 of ISMA [6]. The terminal shall support the "Interleaved RTSP & RTP/AVP over TCP transport" method. For audio and video the restrictions of the present document apply.

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

HTTP as defined in RFC 2616 [7] and HTTP over TLS as defined in RFC 2818 [8] and RFC 5246 [9] shall be supported for transporting application files over broadband.

When using HTTP over TLS the server may send a client certificate request during the TLS handshake as defined in RFC 2818 [8]. The TLS stack implementation shall support negotiation and delivery of client certificates to the server as defined in RFC 5246 [9]. The client certificate shall comply with RFC 5280 [10]. The provision of these certificates is outside the scope of the present document as explained in clause 11.3.

The terminal shall support the default cipher suite of TLS as defined by clause 9 in RFC 5246 [9].

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/1.1.1 (<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 [2] 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) Bee/3.5

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

8 Browser application environment

8.1 DAE Specification Usage

The OIPF DAE specification [2] 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 [2]) shall be supported for synchronization to broadcast events as defined in clause 7.2.4.

`void addStreamEventListener(String targetURL, String eventName, EventListener listener)`		
Description	Add a listener for the specified DSM-CC stream event. When the event is found, a `StreamEvent` event with "`StreamEvent`" type shall be dispatched and passed to the event listener. An event shall also be dispatched in case of error. When a broadcaster transmits an identical instance of the MPEG private data section carrying an event (including the version number), only one event shall be dispatched. When a broadcaster transmits different events using the same event name (i.e. with different version numbers), one event shall be dispatched for each different event received.	
Arguments	`targetURL`	The URL of the DSM-CC StreamEvent object or the event description file 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 the 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

```
interface StreamEvent : Event {
    readonly attribute String      name;
    readonly attribute String      data;
    readonly attribute String      text;
    readonly attribute DOMString   status;
}
```

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`	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). An event might be dispatched with an error status if: • the `StreamEvent` object pointed to by `targetURL` is not found in the carousel or via broadband; • the `StreamEvent` object pointed to by `targetURL` does not contain the event specified by the `eventName` 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). Once an error is dispatched, the listener is automatically unregistered by the terminal.

8.2.1.3 Mapping between DOM 2 events and intrinsic event handlers

Listeners for the events described below may be registered as defined in clause 8.2.1.1 or using intrinsic event handlers. Table 10 shows the mapping between the intrinsic event handlers and the corresponding DOM 2 events.

Table 10: Mapping between DOM 2 events and intrinsic event handlers

Intrinsic event	Corresponding DOM 2 event	DOM 2 Event properties
`onStreamEvent`	`StreamEvent`	Bubbles: No Cancelable: No Context Info: `name`, `data`, `text`, `status`

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()` and `getAllResponseHeaders()` return an empty string.
- Values of the `responseText` and `responseXML` properties are shown in table 11.

Table 11: 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 [12]	If the file has the extension ".xml", returns the "XML response entity body" as defined in XMLHTTPRequest [12].
Directory	`/weather`	Comma-separated list of objects in the directory	null
Stream Event	`/weather/main/streamEvt1`	Comma-separated list of events	null

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

`/weather/data.xml`

`dvb://tf1.B8/weather/data.xml` (0xB8 is the component tag)

8.2.3 Extensions to the application/oipfDrmAgent embedded object

A terminal shall support the following extensions to the application/oipfDrmAgent embedded object type.

8.2.3.1 Properties

`function onContentPurchaseMessageReceived( Integer messageNb, String messageBytes, String DRMSystemID )`
The function that is called whenever a valid `SAS_async_msg()` APDU is received, via the SAS resource opened as defined in clause 11.4.2, from the CA system indicated by the `DRMSystemID` argument. The specified function is called with three arguments `messageNb`, `messageBytes` and `DRMSystemID` which are defined as follows: Integer `messageNb` - the value of `message_nb` from the `SAS_async_msg()` APDU. String `messageBytes` - the contents of the `message_bytes` field from the `SAS_async_msg()` APDU. This shall be encoded in hexadecimal, for example: "`0A10B81033`" (for a message 5 bytes long). String `DRMSystemID` - a string containing the URN of the DVB CASystemID which originated the message. The string is constructed by prefixing the decimal number format of the `CA_System_ID` with "`urn:dvb:casystemid:`".

`function onContentPurchase (Integer cpEventID, String DRMSystemID)`

The function that is called whenever an event listed in the table below occurs.
The specified function is called with two arguments `cpEventID` and `DRMSystemID` which are defined as follows:
Integer `cpEventID` - event identifier. Valid values are:

Event identifier	Description	Semantics
0	Session opened successfully.	The session for the SAS resource to the indicated CA system has been successfully opened.
1	Session closed.	The session for the SAS resource to the indicated CA system has been closed.
2	CA system unavailable.	The CA system has become unavailable, e.g. the CAM has been removed.
3	New CA system available.	A new CA system has become available, e.g. a CAM has been inserted.

String `DRMSystemID` - the CA system that has generated the event.

8.2.3.2 Methods

`Integer sendContentPurchaseMessage( Integer messageNb, String messageBytes, String DRMSystemID )`

Description

Send message to the CA system in the content purchasing message format. Returns a result code as follows:

Result code	Description	Semantics
0	Success.	Message passed successfully to the underlying software stack. Note that this does NOT indicate successful delivery of the message.
1	Invalid CA system.	The CA system indicated by the `DRMSystemID` does not exist on the terminal.
2	Session not ready.	The session to the selected CA system has not been fully set up yet.
3	Protocol not supported.	The CA system does not support content purchasing messages.

The `length_field()` and `message_length` fields (see section M.2.2 of CI Plus [13]) in the `SAS_async_msg()` shall be automatically generated by the terminal.

Arguments

`messageNb`	Message number as defined in section M.2.2 of CI Plus [13] that maps to `message_nb` in `SAS_async_msg()`.
`messageBytes`	Message bytes as defined in section M.2.2 of CI Plus [13] that map to `message_byte()` in `SAS_async_msg()`. This shall be encoded in hexadecimal, for example: "`0A10B81033`" (for a message 5 bytes long).
`DRMSystemID`	URN indicating the CASystemID to send the message to, as defined in clause 11.4.2.1.

8.2.3.3 Events

For the intrinsic events `onContentPurchase` and `onContentPurchaseMessageReceived`, corresponding DOM 2 events shall be generated in the following manner.

Intrinsic event	Corresponding DOM 2 event	DOM 2 Event properties
`onContentPurchase`	`ContentPurchaseEvent`	Bubbles: No Cancelable: No Context Info: `cpEventID, DRMSystemID`
`onContentPurchaseMessageReceived`	`ContentPurchaseMessageReceivedEvent`	Bubbles: No Cancelable: No Context Info: `messageNb, messageBytes, DRMSystemID`
NOTE: The above DOM 2 event is directly dispatched to the event target, and will not bubble nor capture. Applications should not rely on receiving a `ContentPurchaseEvent` event during the bubbling or the capturing phase. The `addEventListener()` method should be called on the `application/oipfDrmAgent` object itself. The third parameter of `addEventListener`, i.e. "`useCapture`", will be ignored.		

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 plane to be made transparent and the audio to stop.

9.1.1.2 HTTP streaming

The mapping from the APIs for unicast streaming to the protocols shall be as defined in clause 8.1.3.1 of the OIPF DAE specification [2] for HTTP streaming.

9.1.1.3 RTSP streaming

The mapping from the AV Control object APIs for unicast streaming to the RTSP protocol is defined as follows.

Table 12: Mapping of AV Control object methods to RTSP protocol

Method	Procedures
`play( Number speed )`	The `speed` parameter is a floating point value indicating the requested playback speed. A value of 1 represents normal playback speed, and other values are relative to this. If the terminal has not yet setup an RTSP session, i.e. the current play state is `stopped`, the RTSP DESCRIBE method and the SETUP method shall be used with the URL specified in the data attribute of the A/V embedded object. The `PlayStateChanged` events `connecting` and `buffering` shall be generated. If the connection fails, the `PlayStateChanged` event `error` shall be generated with an error code 1. The `data` attribute shall furthermore be updated with the new URI after the terminal receives a redirection request. For values not equal to 0 the terminal shall do the following procedures: • The RTSP PLAY request shall include a `Scale` header set to the value of the `speed` parameter passed to the API. • The server will play the stream at the specified speed, if supported. After a successful response to the RTSP PLAY message has been received, the terminal shall generate a `PlaySpeedChanged` event indicating the actual playback speed. A speed value of zero shall initiate the following actions: • The terminal shall initiate the RTSP PAUSE request. • The `PlayStateChanged` event `paused` shall be generated.
`stop()`	The terminal shall initiate the TEARDOWN method and generate a `PlayStateChanged` event `finished`.
`seek( Integer pos )`	Sets current play position to `pos`, by using the `Range` parameter in the RTSP PLAY method. After a successful response to the RTSP PLAY message has been received, the terminal shall generate a `PlayPositionChanged` event indicating a new playback position of `pos`.
`next()`	Not Supported. The terminal shall ignore this call.
`previous()`	Not Supported. The terminal shall ignore this call.

Table 13: Mapping of AV Control object properties to RTSP protocol

Property	Procedures
`read/write String data`	This property holds the RTSP URL for the content item.
`readonly Number playPosition`	The property holds the current play position in milliseconds of the media referenced by the `data` property. The property value should be based on the value retrieved using the RTSP GET PARAMETERS method and parameter `position` (see note). If information is not available the value shall be `undefined`. Note this may happen at the beginning of playing a video and GET_PARAMETER has not returned a value.
`readonly Number playTime`	The property holds the total duration in milliseconds of the media referenced by the data property. The property value shall be based on the value retrieved using RTSP GET_PARAMETER method and parameter `duration` (see note). If information is not available the value shall be `undefined`. Note this may happen at the beginning of playing a video and GET_PARAMETER has not returned a value.
`readonly Number playState`	No mapping defined since it is not related to protocol specification.
`readonly Number error`	No mapping defined since it is not related to protocol specification.
`readonly Number speed`	Float value indicating the actual playback speed of the player for the content referenced by the data property. The normal default playback speed is represented by value 1.
NOTE: RTSP GET_PARAMETER is defined in OIPF and not by ISMA.	

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 8.1.1.1 of the OIPF DAE specification [2].

9.2 URLs

The `http:` and `https:` URL schemes shall be supported as defined in clause 8.3 of the OIPF DAE specification [2].

The `dvb:` URL scheme as included in TS 102 809 [3] shall be supported as follows:

- It shall be possible to use `dvb:` URLs including `path_segments` to refer to DSM-CC file objects and to DSM-CC stream event objects signalled in the current service.

- 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 [11] 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. Parameters assigned to this DVB URL shall be parsed and attached to the application location URL signaled inside the corresponding AIT. The window.location.href property shall take the value of the resulting URL, including any parameters.

- 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.

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.

10 Capabilities

10.1 Display model

10.1.1 Logical plane model

Digital TV terminals typically have multiple planes for displaying graphics, subtitles, video and background color. This clause defines a logical plane model for Hybrid Broadcast Broadband TV terminals. Figure 16 shows the ordering of these logical planes.

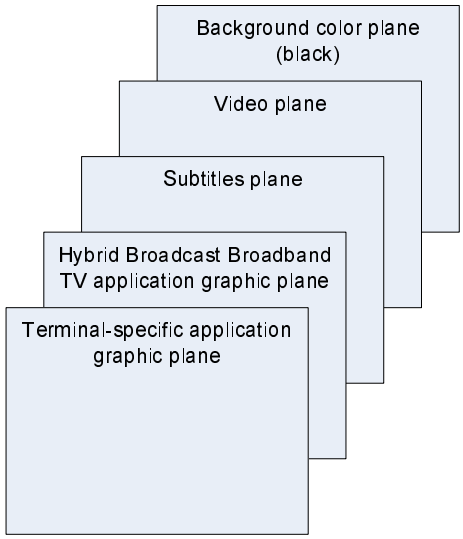


Figure 16: Logical plane model

This logical plane model does not imply any particular physical implementation. For instance, the presence of two graphic planes and a subtitle plane does not imply a requirement for three hardware graphic planes.

The logical planes are defined as follows:

- The "Background color plane" shall display a single uniform color which shall be black. This plane is at the bottom of the logical display stack.
- The "Video plane" shall be used to display video. This plane is on top of the background color plane in the logical display stack. The interaction between the "video plane" and the video/broadcast object is described in clause 10.1.2. Streamed video may appear to be presented in a plane other than the logical video plane. The present document is intentionally silent about the mechanism used by terminals to achieve this behaviour.
- The "Subtitles plane" shall be used to display subtitles. This plane is on top of the video plane in the logical display stack.

- The "Hybrid Broadcast Broadband TV application graphic plane" shall be used to display the Hybrid Broadcast Broadband TV application. This plane is on top of the subtitles plane in the logical display stack. The logical resolution of this plane is 1 280 pixels horizontally by 720 pixels vertically.

- The "Terminal specific application graphic plane" shall be used to display terminal specific application such as system menus, banners or pop-ups. This plane is on top of the Hybrid Broadcast Broadband TV application graphic plane in the logical display stack.

For subtitles, the following rules apply:

- Terminals should support simultaneous display of Hybrid Broadcast Broadband TV application and subtitles. In that case, terminals shall display the application over the subtitles (as shown in figure 16). If the video is rescaled, the subtitles shall be rescaled/repositioned appropriately or not displayed at all.

- If the presentation of subtitles is requested prior to the launch of an autostart application, then terminals which cannot support simultaneous display of Hybrid Broadcast Broadband TV applications and subtitles, shall display subtitles in preference to running the application. Terminals may offer the end-user the opportunity to disable subtitles and run the application instead.

- If the presentation of subtitles is requested while an application is running, terminals which cannot support simultaneous display of Hybrid Broadcast Broadband TV applications and subtitles shall display the application in preference to the presentation of subtitles.

NOTE: In consequence, display of subtitles with broadband delivered video is only possible on such terminals by including the subtitles as part of the video.

10.1.2 Interaction with the video/broadcast object

The behaviour of the video/broadcast object is defined in clause A.2.4. When no video/broadcast object is instantiated, or when all video/broadcast objects are in the Unrealized state, broadcast video presentation is under the control of the terminal. When video is under the control of the terminal:

- Any broadcast video being presented shall be displayed in the logical video plane.

- The complete logical video plane shall be filled.

- The terminal may scale and/or position video, for example to remove black bars.

When a video/broadcast object is in any state other than the Unrealized state, broadcast video presentation is under the control of the application. When video is under the control of the application:

- When the video/broadcast object is not in "full-screen mode", any broadcast video being presented shall be scaled and positioned to fit the video/broadcast object. The area of the video plane not containing video shall be transparent.

- When the video/broadcast object is in "full-screen mode", presented video shall be scaled to fill the logical video plane. The terminal may further scale and/or position video, for example to remove black bars.

- Depending on the Z index of the video/broadcast object with respect to other HTML elements (regardless of whether the object is in "fullscreen mode" or not), presented video may fully or partially obscure other HTML elements with a lower Z index, and may in turn be fully or partially obscured by HTML elements with a higher Z index. As a result of this, video may appear to be presented in a plane other than the logical video plane. The present document is intentionally silent about the mechanism used by terminals to achieve this behaviour.

If the `release()` method is called on a video/broadcast object, or if the object is garbage collected, control of broadcast video presentation shall be returned to the terminal and video shall be re-scaled and re-positioned (if necessary).

10.1.3 Graphic safe area (informative)

Figure 17 shows the recommended safe area for content authoring.

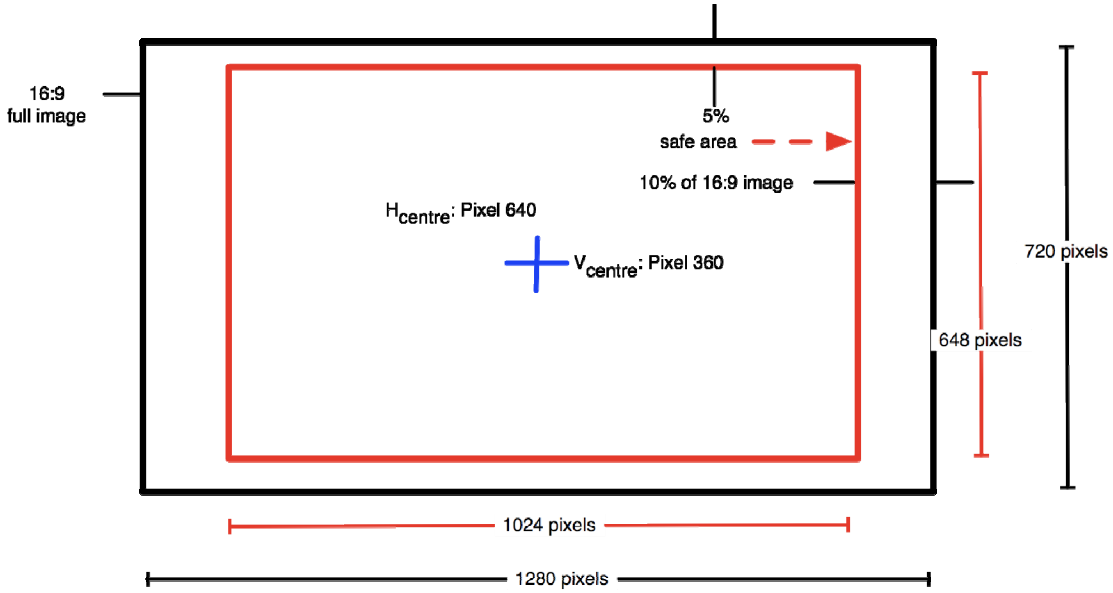


Figure 17: Graphic safe area

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 14 for general capabilities. Additional capabilities should be signalled in the capability profile as defined by clause 9 of the OIPF DAE specification [2].

Table 14: Minimum terminal capabilities

	Value	Characteristic
Screen resolution	1 280 times 720 pixels with a 16:9 aspect ratio.	Static resolution.
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	"Tiresias Screenfont" (or equivalent) with the support for the Unicode character range "Basic Euro Latin Character set" as defined in annex D of TS 102 809 [3].
	
	The font shall be the default font to be used when none is explicitly specified.
	
	This font (even if it is an equivalent of "Tiresias Screenfont") shall be accessible with the following CSS rule:
	`font-family: Tiresias;`
	
	It shall also possible to use the "`sans-serif`" generic family name to point to the "Tirerias Screenfont" font (even if other sans-serif fonts are available in the terminal), i.e. "`sans-serif`" shall default to the "Tiresias Screenfont" font:
	`font-family: sans-serif;`
		Sans-serif, scalable and anti-aliased font.

Supported non-proportional font	"Letter Gothic 12 Pitch" (or equivalent) with the support for the Unicode character range "Basic Euro Latin Character set" as defined in annex D of TS 102 809 [3].
	
	This font (even if it is an equivalent of "Letter Gothic 12 Pitch") shall be accessible with the following CSS rule:
	`font-family: "Letter Gothic 12 Pitch";`
	
	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:
	`font-family: monospace;`
		Monospace, scalable and anti-aliased font.

	Value	Characteristic
Text entry method	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. For multi-tap or other methods which use supported key events to generate characters, these intermediate key events shall not be reported to applications. Only the final character result shall be reported to applications. NOTE: The `input-format` CSS property may be used by terminals to determine which text entry method to use.	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	The terminal shall allocate sufficient resources to acquire DSM-CC sections from at least 3 elementary streams simultaneously for a given DSM-CC carousel. In addition, a terminal shall reserve at least one section filter for monitoring DSM-CC StreamEvent's events.	
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, 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 [2]).	
PVR management	If the PVR feature is supported, the `manageRecordings` attribute of the recording capability shall have the value "`samedomain`".	See clause 9.3.3 of the OIPF DAE specification [2].
Download management	If content download is supported, the `manageDownload` attribute of the download capability shall have the value "`samedomain`".	See clause 9.3.4 of the OIPF DAE specification [2].
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 [17].	

	Value	Characteristic
Video scaling	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. 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. Terminals shall be able to scale video up to 2 x 2 of the width and height of the logical video plane. 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.	

10.2.2 User input

Implementations shall provide a mechanism for the end user to generate key events as defined in table 15.

Table 15: 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
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 `charCode` 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 based which shall allow focus to be moved between navigable elements and allow all navigable elements to gain focus.

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.

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 option strings

The strings defined in this clause 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 and as parameters of a JavaScript API 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 [2].

Table 16: Hybrid Broadcast Broadband TV Option Strings

Option string	Meaning
"+DL"	Support for file download feature
"+PVR"	Support for PVR feature
"+RTSP"	Support for RTSP streaming feature

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. 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+ [13] 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").

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 [2] 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 [2].

11.2 TLS and SSL Root Certificates

At least the root certificates listed in table 17 shall be supported.

Table 17: Minimum required root certificates

Root CA certificate name	Expiry date
Thawte Personal Basic CA	1/1/2021
Thawte Personal Freemail CA	1/1/2021
Thawte Personal Premium CA	1/1/2021
Thawte Premium Server CA	1/1/2021
Thawte Server CA	1/1/2021
VeriSign Class 1-3 Public Primary Certification Authority	8/2/2028
VeriSign Class 1-4 Public Primary Certification Authority - G2	8/2/2028
VeriSign Class 1-4 Public Primary Certification Authority - G3	7/17/2036

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+ 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 4.2.3.4.1.1.5-6-7, 4.2.3.4.1.1.9-10, 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 Not Included.
- 4.2.3.8 Not Included.
- 4.2.3.9 Not Included.
- 4.2.3.10 N/A.

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.

11.4.2 CI+ support for content purchasing

11.4.2.1 General

Terminals supporting CI or CI+ shall support the API extensions defined below to enable an application to access some CA related functionality, in particular relating to content purchasing.

NOTE: This is equivalent to that provided by the MHP it.dtt.ca messages [i.3].

Terminals supporting embedded CA shall support the API extensions defined below by providing the equivalent functionality to that provided by the SAS resource.

Terminals with no CI, CI+ or embedded CA support shall support the API extensions defined below in so far as all calls to `sendContentPurchaseMessage` shall return error code 1.

11.4.2.2 Control channel

Terminals shall use the SAS resource, defined in clause 11.4 of CI Plus [13] to handle APDUs as specified in section M.2 of CI Plus [13] and in this clause.

The terminal shall create a session to each SAS resource of the available DRM systems as soon as it has completed its Application Information phase of initialisation or whenever a new DRM system becomes available. The terminal shall send a `SAS_connect_rqst()` APDU to the CICAM with the `private_host_application_ID` value defined in section M.2.1 of CI Plus [13]. If the CICAM does not correctly acknowledge the connection by sending back a `SAS_connect_cnf()` APDU with a success connection status, all calls to `sendContentPurchaseMessage()` shall fail with a suitable error code.

11.4.2.3 Sending and Receiving Messages

The terminal shall map calls to `sendContentPurchaseMessage()` to outgoing `SAS_async_msg()` APDUs and incoming SAS_async_msg() APDUs to `onContentPurchaseMessageReceived` events.

Content purchasing messages use the `SAS_async_msg()` syntax and semantics as defined in section M.2.2 of CI Plus [13].

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 [2]	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. `ApplicationUnloaded` 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 `gc()` 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 below.	
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.	
Behaviour of the BACK key	4.4.7.1	M(*)	The behaviour of the `VK_BACK` key when not used by an Hybrid Broadcast Broadband TV application shall not be used to go back in the history list.	

Section, sub-section	Reference in DAE [2]	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.	Trusted
Conveyance of channel list and list of scheduled recordings	4.8.2	M-P		Trusted
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	M-M(*)	Applications trying to present the protected content should handle DRM-specific UI. 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.	
Applications provided by the AG through the remote UI	5.1.1.7	NI		
Stopping an application	5.1.2	M(*)	The requirement that "Applications SHALL also be destroyed when following a link to a page loaded from a different domain" does not apply. See 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		

Section, sub-section	Reference in DAE [2]	Status in Hybrid Broadcast Broadband TV	Notes	Security
Event Notification Framework based on CEA 2014 - XMLHttpRequest	5.3.1.1	M		None
Out of Session even the 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 getOwnerApplication() method and onLowMemory property (and corresponding DOM 2 event) 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: - a property "privateData" (which shall have the same semantics as the private property) - `createApplication(URI,false)` - `destroyApplication()` - `show()` - `hide()` (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: - `keyset` - `currentChannel` (see clause A.2.2 below) - `getFreeMem()` All other properties and methods are not included.	None
The KeySet class	7.2.5	M(*)	The default set of key events available to broadcast-related applications shall be none. The default set of key events available to broadcast-independent applications shall be as specified in the OIPF DAE specification [1]. The `otherKeys` and `maximumOtherKeys` properties are not included.	None
New DOM events for application support	7.2.6	NI		None

Section, sub-section	Reference in DAE [2]	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. The `localSystem` property is not included.	None
The Configuration class	7.3.2	M(*)	Support for read-only access to the following properties is mandatory: - `preferredAudioLanguage` - `preferredSubtitleLanguage` - `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 DRMControlInformation class	7.4.6	M-D, M-M	Mandatory if both Download and DRM features are supported.	Trusted
The DRMControlInfoCollection class	7.4.7	M-D, M-M	Mandatory if both Download and DRM features are supported.	Trusted
Content On Demand Metadata APIs	7.5	NI		
Content Service Protection API	7.6	M-C, M-M	Mandatory if either CI+ or DRM features are supported.	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
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/oipfRecordingScheduler embedded object	7.10.1	M-P		Trusted
The ScheduledRecording class	7.10.2	M-P(*)	`ID_TVA_CRID` is not included. Support for series recording is optional.	Trusted
The ScheduledRecordingCollection	7.10.3	M-P		Trusted

Section, sub-section	Reference in DAE [2]	Status in Hybrid Broadcast Broadband TV	Notes	Security
class				
Extension to application/oipfRecordingScheduler for control of recordings	7.10.4	M-P(*)	The `recordings` property shall be supported. Other properties and methods are not included.	Trusted
The Recording class	7.10.5	M-P(*)	The following properties shall be supported: - `state` - `id` - `recordingStartTime` - `recordingDuration` - `parentalRatings` All other properties are not included.	Trusted
The RecordingCollection class	7.10.6	M-P		Trusted
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/oipfSearchManager embedded object	7.12.1	NI		
The MetadataSearch class	7.12.2	NI		
The Query class	7.12.3	NI		
The SearchResults class	7.12.4	NI		
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 is 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-P		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	NI		
Extensions to video/broadcast for channel scan	7.13.7	M(*)	The `currentChannel` property of the video/broadcast object shall be supported. The remainder of these two clauses is not included. Access to this property by broadcast-independent applications shall return `null`.	Broadcast-related
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 and methods are not included.	Broadcast-related
ChannelList class	7.13.10	M(*)	The `getChannelBySourceID()` method is not included.	Broadcast-related

Section, sub-section	Reference in DAE [2]	Status in Hybrid Broadcast Broadband TV	Notes	Security
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 ChannelScanEvent class	7.13.14	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 contol 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	M-M		None
Extensions to AV object for trickmodes	7.14.3	M-R(*)	The following properties and the corresponding DOM 2 events are required where the RTSP feature is supported: - `onPlaySpeedChanged` - `onPlayPositionChanged` Others are not included.	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	NI		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. Extensions are found clause A.2.3 below.	None
The Navigator class	7.15.4	M		None
Debug Print API	7.15.5	M		None
The StringCollection class	7.16.1	NI		

Section, sub-section	Reference in DAE [2]	Status in Hybrid Broadcast Broadband TV	Notes	Security
The Programme Class				
Basics	7.16.2.1, 7.16.2.2	M(*)	The following properties are required: - `name` - `description` - `startTime` - `duration` - `channelID` - `parentalRating` All other properties and methods are not included.	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		
The DiscInfo class	7.16.4	NI		
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. The rtsp URL scheme shall be supported if the RTSP feature is supported.	
Capabilities				
Minimum DAE capability requirements	9.1	NI	See clause 10.2.1 in 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		

Section, sub-section	Reference in DAE [2]	Status in Hybrid Broadcast Broadband TV	Notes	Security
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 messaging	B	NI		
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 suppported	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	M		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	M-M		
Abstract Content Access Descriptor Format	E.3	M-D, M-M	Shall be supported if either the download or DRM features are supported.	

Section, sub-section	Reference in DAE [2]	Status in Hybrid Broadcast Broadband TV	Notes	Security
Capability Extensions Schema	F	M		
Client Channel Listing Format	G	NI		
DVB-MCAST URI scheme for services in a MPEG-2 TS delivered over IP Multicast	H	NI		

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 `message` property set to `SecurityError` (see clause 10.1.1 of the OIPF DAE specification [2]).
broadcast-related	Broadcast-related applications shall have access to the referenced API regardless of whether they are trusted or not.
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.
M-D	Mandatory if the download feature supported otherwise not included.
M-M	Mandatory if the DRM feature is supported otherwise not included.
M-P	Mandatory if the PVR feature is supported otherwise not included.
M-R	Mandatory if the RTSP feature is supported otherwise not included.
NI	Not included.
NOTE: 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.	

A.2 Modifications, extensions and clarifications

A.2.1 Resource management

NOTE: This clause modifies clause 4.4.5 of the OIPF DAE specification [2].

If insufficient resources are available to present the media, the attempt to play the media shall fail except for the specific case of starting to play audio from memory (see below). For the video/broadcast object, this shall be indicated by a `ChannelChangeError` event with a value of 11 for the error state. For an AV Control object, the `error` property shall take the value 3.

Instantiation of a video/broadcast or AV Control object does not cause any scarce resources to be claimed. Scarce resources such as a media decoder are only claimed following a call to the `setChannel()`, `bindToCurrentChannel()`, `nextChannel()` or `prevChannel()` methods on a video/broadcast object or the `play()` method on an AV Control object. By implication, instantiating a video/broadcast or AV Control object does not cause the media referred to by the object's `data` attribute to start playing immediately.

When a video/broadcast object is destroyed (e.g. by the video/broadcast object being garbage collected), or when the `release()` method is called, control of broadcast video shall be returned to the terminal. If an application has modified the set of components being presented (e.g. changing the audio or subtitle stream being presented) then the same set of components will continue to be presented.

When a video/broadcast object is destroyed due to a page transition within an application, terminals may delay this operation until the new page is fully loaded in order to avoid display glitches if a video/broadcast object is also present in the new page. Presentation of broadcast video or audio shall not be interrupted in either case.

When an AV Control object is destroyed (e.g. by the AV Control object being garbage collected, or because of a page transition within the application), presentation of streamed audio or video shall be terminated.

In the specific case of a request to play audio from memory while broadcast or broadband streaming audio is being played and where the terminal does not support mixing the audio from memory with the already playing audio, the following shall apply:

- The audio from memory shall have priority and shall interrupt the already playing audio.
- The interrupted presentation shall be resumed automatically by the terminal when the interrupting audio ends.

The present document is intentionally silent about handling of resource use by terminal-specific applications including scheduled recordings.

A.2.2 Extensions to the ApplicationPrivateData class

This class shall be extended with the following additional property.

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

A.2.3 Extensions to the oipfCapabilities embedded object

This embedded object shall be extended with the following additional properties.

`readonly Number extraSDVideoDecodes`
This property holds the number of possible additional decodes for SD video. Depending on the current usage of system resources this value may vary. Adding an A/V object may still fail, even if `extraSDVideoDecodes` is larger than 0. In case of failure the play state for the A/V object shall be set to '`error`' with a detailed error code of 'insufficient resource'. The value of this property is likely to change if an HD video is started.

`readonly Number extraHDVideoDecodes`
This property holds the number of possible additional decodes for HD video. Depending on the current usage of system resources this value may vary. Adding an A/V object may still fail, even if `extraHDVideoDecodes` is larger than 0. In case of failure the play state for the A/V object shall be set to '`error`' with a detailed error code of 'insufficient resource'. The value of this property is likely to change if an SD video is started.

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 [2].

Figure A.1 shows the states that a video/broadcast object may be in. Dashed lines indicate automatic transitions between states. The video/broadcast object shall be in the unrealized state when it is instantiated.

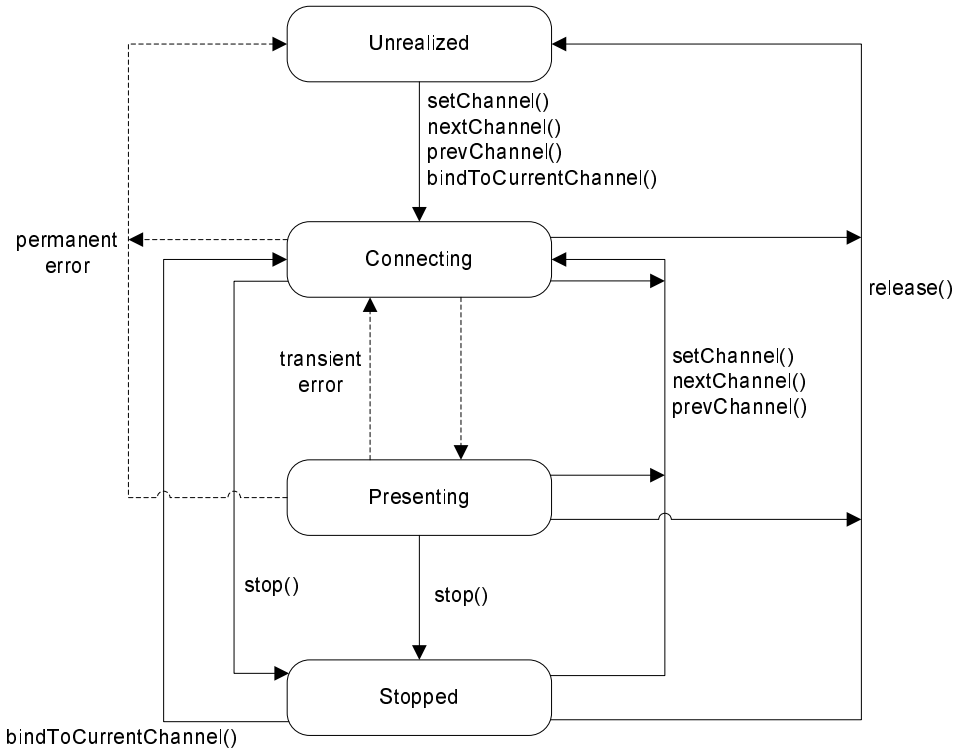


Figure A.1: State machine for the video/broadcast object

Under the following conditions, the object shall transition to the connecting state, in which the terminal attempts to connect to the broadcast stream:

- When the `setChannel()`, `nextChannel()` or `prevChannel()` method is called from the unrealized, connecting or presenting states.
- When the `bindToCurentChannel()` method is called from the unrealized or stopped states.

Calling the `stop()` method will stop video and audio presentation and cause the video/broadcast object to transition to the stopped state. This has no effect on access to non-media broadcast resources such as AIT monitoring or access to objects in a DSM-CC carousel as defined in clause 6.2.2.7. Calling the `bindToCurrentChannel()` method while in the stopped state shall result in video and audio presentation being restarted. Calling the `setChannel()`, `nextChannel()` or `prevChannel()` methods while in the stopped state shall result in the terminal attempting to select the new service.Applications can use the `playState` property of the video/broadcast object to read its current state.

Table A.4 clarifies and extends the possible values of the `playState` property and the `state` property of the `PlayStateChange` event.

Table A.4: Possible values of the playState property and state property of the PlayStateChanged event

Value	Description
0	unrealized; the application has not made a request to start presenting a channel or has stopped presenting a channel and released any resources. The content of the video/broadcast object is transparent. Control of media presentation is under the control of the terminal, as defined in clause 10.1.2.
1	connecting; the terminal is connecting to the media source in order to begin playback. Objects in this state may be buffering data in order to start playback. Control of media presentation is under the control of the application, as defined in clause 10.1.2. The content of the video/broadcast object is transparent.
2	presenting; media is currently being presented to the user. The object is in this state regardless of whether the media is playing at normal speed, paused, or playing in a trick mode (e.g. at a speed other than normal speed). Control of media presentation is under the control of the application, as defined in clause 10.1.2. The video/broadcast object contains the video being presented.
3	stopped; the terminal is not presenting media, either inside the video/broadcast object or in the logical video plane. The logical video plane is disabled. Control of media presentation is under the control of the application, as defined in clause 10.1.2. The application is still granted access to broadcast resources as defined in clause 6.2.2.7.

The following method is defined on the video/broadcast object:

`void stop()`	
Description	Stop presenting broadcast video. The video/broadcast object transitions to the stopped state. Calling this method from the unrealized state shall have no effect. See the state diagram for more information of its usage.

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`, `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 fail: `getChannelConfig`, `bindToCurrentChannel`, `prevChannel`, `nextChannel`, `setVolume` and `getVolume`.
- The following methods shall have no effect: `setFullScreen` and `release`.
- The object shall always be in the unrealized state.

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.2.5 Extensions to the AV Control Object

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

`Boolean queue(String url)`		
Description	Queue the media referred to by `url` for playback after the current media item has finished playing. If a media item is already queued, `url` 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. If `url` is null, any currently queued item will be removed from the queue and this method will return true. 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. When the current media item has finished playing, the AV Control object shall transition to the finished state, update the value of the `data` 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. Play speed is not affected by transitioning between the current and queued media item. To avoid race conditions when queueing multiple items for playback, applications should wait for the currently queued item to begin playback before queuing 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.	
Arguments	`url`	The media item to be queued, or null to remove the currently-queued item.

Calling `stop()` or modifying the data property 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 following value shall be added to the list of valid values for the error property:

- 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 [2] shall apply with the following restrictions and extensions:

- Support of the `AccessKey` attribute for standardized key-codes is not included. Services may instead use global key handling.
- The `window.download` method is not included, even in terminals supporting A/V content download.
- Direct keycodes are not included. Only the virtual keycodes (e.g. `VK_`) are required.
- 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">
```

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

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

NOTE: 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

The contents of an `<iframe>` element may be retrieved from a domain other than the one from which the top-level document is loaded. In this case, terminals shall enforce security restrictions between the contents of the iframe element and the parent document. These restrictions may be based on the nested browsing context as defined in clause 6.1.1 of HTML5 [18] and the security restrictions formalised in clause 6.3.1 of HTML5 [18] excluding the features not included in the present document.

A.2.6.4 Browser History

The terminal should not offer a history UI for Hybrid Broadcast Broadband TV applications, although Hybrid Broadcast Broadband TV applications may use the methods on the `History` object to navigate the history list. The history list shall not go back beyond the initial page of an Hybrid Broadcast Broadband TV application.

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.

The behaviour of the `VK_BACK` key when not used by an Hybrid Broadcast Broadband TV application shall not be to go back in the history.

A.2.7 CSS profile

The `text-shadow` property is not included.

A.2.8 DOM profile

A.2.8.1 The Window object

The following properties shall be supported on the window object:

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

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

The protection, by technical means, of content delivered via broadband is outside the scope of the present document.

However, implementations of the present document that need this capability are permitted to incorporate any content protection systems including or beyond those defined in the OIPF specifications when appropriate.

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 [19];
- using DVB-CA identifier codepoints (CA_System_ID) allocated as usual by the DVB Project and found in TS 101 162 [20] for the DRMSystemID.

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.4] 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 [2] 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 oject 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`.

History

Document history		
V1.1.1	June 2010	Publication