

Confidential



Errata 2 to TS 102 796 V1.4.1

2018-02-12

Copyright 2018 HbbTV Association

Table of Contents

1 Introduction.....	5
2 Conventions.....	5
3 Summary.....	5
4 Changes to TS 102 796 v1.4.1.....	8
4.1 Clause 1 – Scope.....	8
4.1.1 Running apps from previous spec versions.....	8
4.2 Clause 2 – References.....	9
4.2.1 Update reference to TS 102 809.....	9
4.2.2 UPnP version to be used with DIAL.....	9
4.2.3 Ignoring unsupported AIT descriptors.....	9
4.2.4 XML Parsing.....	9
4.2.5 Update inter-device sync spec from DVB blue book back to ETSI.....	9
4.3 Clause 5 – User experience.....	10
4.3.1 Update key event text in clause 5.2.....	10
4.4 Clause 6 – Service and application model.....	10
4.4.1 Channel change key behaviour with broadcast independent applications.....	10
4.4.2 Clarification to 6.2.2.9 re key events.....	10
4.4.3 video/broadcast object issues when playing A/V from broadband.....	10
4.4.4 Running a regular HbbTV app on a channel not in the terminal channel list.....	11
4.4.5 Web Storage following channel change.....	11
4.5 Clause 7 – Formats and protocols.....	11
4.5.1 Update reference to TS 102 809.....	11
4.5.2 Clarification on XML AIT example.....	12
4.5.3 Missing XML declaration in example XML AIT.....	12
4.5.4 Ignoring unsupported AIT descriptors.....	12
4.5.5 XML Parsing.....	13
4.5.6 Lack of linkage between spec version in AIT signalling and version in DOCTYPE in individual pages of app.....	13
4.5.7 Including or not including data services in the service list based on HbbTV version.....	13
4.5.8 Ambiguous use of “this”.....	13
4.5.9 Clarification on FSA: Is group priority value 0x00 valid?.....	13
4.5.10 Carousel access following channel change.....	14
4.5.11 Pending requests for FSA files when carousel is unmounted.....	14
4.6 Clause 8 – Browser application environment.....	14
4.6.1 Avoid MediaSynchroniser API leaking "secret" stream URLs.....	14
4.6.2 Clarifying media synchroniser error codes.....	15
4.6.3 Issue with companion launcher API.....	18
4.6.4 XML parsing.....	19

Confidential

- 4.6.5 Update inter-device sync spec from DVB blue book back to ETSI..... 19
- 4.6.6 Clarification on value for property "lastError" if no error occurred..... 19
- 4.6.7 errorCode(s) for onCSLaunch..... 20
- 4.7 Clause 9 – System integration..... 20
 - 4.7.1 Clarify URL scheme and MPD anchor rules..... 20
 - 4.7.2 Clarify resource usage by HTML5 media elements..... 20
 - 4.7.3 Additional cases when 250ms ad insertion transition may not be possible to meet..... 21
 - 4.7.4 Timing requirements for end of mid roll adverts..... 21
 - 4.7.5 Require the same DASH player for HTML5 video element and A/V control object?... 21
- 4.8 Clause 10 – Capabilities..... 21
 - 4.8.1 Removing support for CSS3 navigation..... 21
 - 4.8.2 XML Parsing..... 22
 - 4.8.3 Subtitle track selection - ability for an app to suppress subtitles reliably..... 22
 - 4.8.4 Update inter-device sync spec from DVB blue book back to ETSI..... 24
- 4.9 Clause 11 – Security..... 24
 - 4.9.1 Update reference to TS 102 809..... 24
 - 4.9.2 SHA-1 sunset date has passed..... 24
- 4.10 Clause 12 – Privacy..... 24
 - 4.10.1 Confusion over deviceId class..... 24
- 4.11 Clause 13 - Media synchronization..... 24
 - 4.11.1 Avoid MediaSynchroniser API leaking "secret" stream URLs..... 24
 - 4.11.2 Clarification of TEMI timeline requirements..... 26
 - 4.11.3 Update inter-device sync spec from DVB blue book back to ETSI..... 26
- 4.12 Clause 14 – Companion screens..... 26
 - 4.12.1 UPnP version to be used with DIAL..... 26
 - 4.12.2 Clarification on XML AIT example..... 27
 - 4.12.3 URL used in in DIAL example should be changed..... 27
- 4.13 Annex A – OIPF DAE Specification Profile..... 27
 - 4.13.1 Text referring to MetadataSearch in wrong location in annex A..... 27
 - 4.13.2 "channels" should be "audioChannels"..... 28
 - 4.13.3 Clarify resource usage by HTML5 media elements..... 28
 - 4.13.4 VK_RECORD key is not in any set of keys..... 28
 - 4.13.5 XML Parsing..... 29
 - 4.13.6 bindToCurrentChannel() while an A/V Control object is presenting..... 29
 - 4.13.7 Key “Label” should be “label”..... 31
 - 4.13.8 video/broadcast object issues when playing A/V from broadband..... 31
 - 4.13.9 Lack of linkage between spec version in AIT signalling and version in DOCTYPE in individual pages of app..... 32
 - 4.13.10 Two setChannel methods of video/broadcast defined with same signature..... 32
 - 4.13.11 PVR API issues..... 33

Confidential

- 4.13.12 Scope of AVAudio and AVVideoComponents.....33
- 4.13.13 Allow applications to prioritise DRM to use.....34
- 4.13.14 Unclear language in A 2.20.1 - Extensions to Represent Subtitle Presentation.....34
- 4.13.15 Clarification to parental rating.....34
- 4.13.16 Reference to NI clause 10.1.1 of the DAE specification.....35
- 4.13.17 Incoherent AVSubtitleComponent language with TTML tracks.....35
- 4.14 Annex E - Profiles of MPEG DASH.....36
 - 4.14.1 Clarify URL scheme and MPD anchor rules.....36
 - 4.14.2 Frame rate support with UHD content via DASH.....36
- 4.15 Annex G - Implementer guidelines for media synchronization.....36
 - 4.15.1 Update inter-device sync spec from DVB blue book back to ETSI.....36
- 4.16 New Annex L.....36
 - 4.16.1 Removing support for CSS3 navigation.....36

1 Introduction

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

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

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

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

2 Conventions

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

3 Summary

The following table summarises the changes included in this document.

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
3535	XML parsing	Other	2.1, 7.2.3.2, 7.2.4, 8.2.1.1, 10.2.4, A.1, A.2.6.2, E.2.1	4.2.4, 4.5.5, 4.6.4, 4.8.2, 4.13.5
4913	Clarify resource usage by HTML5 media elements	Ambiguity	9.6.2, A.2.1	4.7.2, 4.13.3
5097	Frame rate support with UHD content via DASH	Ambiguity	New clause E.4.6	4.14.2
5307	Clarifying media synchroniser error codes	Ambiguity	8.2.3.2	4.6.2
5314	Update inter-device sync spec from DVB blue book back to ETSI	Other	2.1, 8.2.3, 10.2.8, 13, G.1	4.2.1, 4.6.5, 4.8.4, 4.11.3, 4.15.1
5324	Clarify URL scheme and MPD anchor rules	Ambiguity	9.2, E.4.5	4.7.1, 4.14.1

Confidential

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
5385	Removing support for CSS3 navigation	Other	10.2.2.1, new annex L	4.8.1, 4.16.1
5499	Timing requirements for end of mid roll adverts	Ambiguity	New clause 9.6.12	4.7.4
5500	Confusion over deviceId class	Ambiguity	12.1.5	4.10.1
5575	Running apps from previous spec versions	Editorial	1	4.1.1
5582	Two setChannel methods of video/broadcast defined with same signature	Error	A.2.4.7.4	4.13.10
5583	Ignoring unsupported AIT descriptors	Ambiguity	2.2, 7.2.3.1	4.5.4, 4.2.3
5639	Issue with companion launcher API	Error	8.2.6.1	4.6.3
5738	errorCode(s) for onCSLaunch	Ambiguity	8.2.6.1	4.6.7
5824	Key "Label" should be "label"	Ambiguity	A.2.5.3	4.13.7
5862	Running a regular HbbTV app on a channel not in the terminal channel list	Editorial	6.2.2	4.4.4
5864	text in wrong location in annex A - referring to MetadataSearch	Editorial	A.1	4.13.1
5872	Clarification to 6.2.2.9 re key events	Ambiguity	6.2.2.9	4.4.2
5899	Reference to NI clause 10.1.1 of the DAE specification	Inconsistency	A.1	4.13.16
5910	UPnP version to be used with DIAL	Implementation	2.1, 14.7	4.2.2, 4.12.1
5919	VK_RECORD key is not in any set of keys	Error	A.1	4.13.4
5972	PVR API issues	Error	A.1, A.2.4.7.3, A.2.4.8.1, A.2.4.8.2	4.13.11
5984	Scope of AVAudio and AVVideoComponents	Ambiguity	A.2.4.6	4.13.12
6047	Update key event text in clause 5.2	Editorial	5.2	4.3.1
6049	Avoid MediaSynchroniser API leaking "secret" stream URLs	Other	8.2.3.2.1, 13.6.2, 13.8.2.2	4.6.1, 4.11.1

Confidential

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
6211	video/broadcast object issues when playing A/V from broadband	Ambiguity	6.2.2.7, A.2.4.1, A.2.4.5, new clause A.2.26	4.4.3, 4.13.8
6422	Clarification of TEMI timeline requirements	Ambiguity	13.4.2	4.11.2
6482	Incoherent AVSubtitleComponent language with TTML tracks	Inconsistency	A.1	4.13.17
6495	Additional cases when 250ms ad insertion transition may not be possible to meet	Other	9.6.3	4.7.3
6498	Subtitle track selection - ability for an app to suppress subtitles reliably	Ambiguity	10.2.7.1, 10.2.7.3	4.8.3
6544	Missing XML declaration in example XML AIT	Error	7.2.3.2	4.5.3
6550	Clarification on XML AIT example	Error	7.2.3.2	4.5.2, 4.12.2
6560	Channel change key behaviour with broadcast independent applications	Ambiguity	6.2.2.2	4.4.1
6564	bindToCurrentChannel() while an A/V Control object is presenting	Ambiguity	A.2.1	4.13.6
6591	"channels" should be "audioChannels"	Editorial	A.2.4.6	4.13.2
6610	Update reference to TS 102 809	Other	2.1, 7.2, 11.1	4.2.1, 4.5.1, 4.9.1
6741	Unclear language in A.2.20.1 - Extensions to Represent Subtitle Presentation	Ambiguity	A.2.20.1	4.13.14
6914	URL used in in DIAL example should be changed	Other	14.7.3.1, 14.7.3.2	4.12.3
6915	Lack of linkage between spec version in AIT signalling and version in DOCTYPE in individual pages of app	Other	7.2.3.1, A.2.6.2	4.5.6, 4.13.9
6971	Ambiguous use of “this”	Ambiguity	7.3.2.6	4.5.8
7045	Including or not including data services in the service list based on HbbTV version	Ambiguity	7.2.6	4.5.7
7046	Require the same DASH player for HTML5	Other	New clause	4.7.5

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
	video element and A/V control object?		9.1.1.3	
7088	Allow applications to prioritise DRM to use	Other	A.1, new clause A.2.27	4.13.13
7128	Clarification on FSA: Is group priority value 0x00 valid?	Ambiguity	7.2.7.2	4.5.9
7146	Pending requests for FSA files when carousel is unmounted	Ambiguity	New clause 7.2.7.5	4.5.11
7162	SHA-1 sunset date has passed	Other	11.2.4	4.9.2
7196	Carousel access following channel change	Ambiguity	7.2.5.3	4.5.10
7323	Web Storage following channel change	Other	6.3.2	4.4.5
7364	Clarification on value for property "lastError" if no error occurred	Ambiguity	8.2.3.2.1	4.6.6
7439	Clarification to parental rating	Ambiguity	A.1, new clause A.2.28	4.13.15

Key to categories

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

4 Changes to TS 102 796 v1.4.1

4.1 Clause 1 – Scope

4.1.1 Running apps from previous spec versions

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

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

been deployed.

4.2 Clause 2 – References

4.2.1 Update reference to TS 102 809

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

4.2.2 UPnP version to be used with DIAL

The following normative references are added.

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

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

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

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

4.2.3 Ignoring unsupported AIT descriptors

The following additional informative reference is added:

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

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

4.2.4 XML Parsing

The following normative reference is added;

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

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

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

In clause 2.1, reference 47 is changed from:

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

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

To

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

4.3 Clause 5 – User experience

4.3.1 Update key event text in clause 5.2

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

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

The following text is added immediately after the table.

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

4.4 Clause 6 – Service and application model

4.4.1 Channel change key behaviour with broadcast independent applications

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

The channel change mechanisms offered by the terminal (e.g. P+/P- keys, number keys) shall remain functional at all times while broadcast related applications are running, regardless of whether media is being presented and whether that originates from broadcast or broadband. The behaviour of these channel change mechanisms is implementation-dependent when a broadcast-independent application is running (and hence no broadcast channel is selected).

4.4.2 Clarification to 6.2.2.9 re key events

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

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

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

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

Broadcast related applications that wish to access information from the video/broadcast object, e.g. channelchange succeeded events or stream events, while playing broadband content, should put the video/broadcast object into the stopped state. When an application survives a channel change, e.g. caused by P+/P-, the video/broadcast object transitions from

the stopped state into the connecting state and into the presenting state if available resources permit (e.g. if additional video and audio decoders are available beyond those used for presenting the broadband content). The application is responsible to put it back into the stopped state.

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

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

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

4.4.5 Web Storage following channel change

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

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

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

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

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

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

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

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

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

4.5 Clause 7 – Formats and protocols

4.5.1 Update reference to TS 102 809

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

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

terminals are to be deployed.

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

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

4.5.2 Clarification on XML AIT example

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

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

4.5.3 Missing XML declaration in example XML AIT

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

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

4.5.4 Ignoring unsupported AIT descriptors

Text is added to clause 7.2.3.1 as shown underlined.

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

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

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

Table 5: Supported application signalling features

4.5.5 XML Parsing

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

The XML AIT must not contain an XML Document Type Definition (“<!DOCTYPE ...>”).

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

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

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

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

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

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

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

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

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

- carry a data_broadcast_descriptor that indicates the HbbTV data_broadcast_id and have selector_byte present and
- signal an HbbTV application that is supported by the terminal

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

4.5.8 Ambiguous use of “this”

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

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

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

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

- group_priority: priority 0 is the highest priority and increasing group_priority value means decreasing actual priority. In the present document there is no allocation of priority values.

The system group has no specific meaning.

4.5.10 Carousel access following channel change

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

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

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

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

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

4.5.11 Pending requests for FSA files when carousel is unmounted

A new clause 7.2.7.5 is added as follows:

7.2.7.5 File groups referenced by multiple carousels

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

- The file group cache may continue to be filled
- Pending requests for files in the group to be cached may continue or be restarted in an implementation specific manner

4.6 Clause 8 – Browser application environment

4.6.1 Avoid MediaSynchroniser API leaking "secret" stream URLs

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

```
String contentIdOverride
```

Confidential

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

4.6.2 Clarifying media synchroniser error codes

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

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

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

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

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

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

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

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

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

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

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

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

Confidential

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

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

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

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

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

The terminal shall select the components from the media object to be presented in accordance with the value of the multiDecoderMode parameter and the definitions in clause 10.2.7.

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

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

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

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

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

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

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

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

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

Confidential

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

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

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

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

Confidential

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

4.6.3 Issue with companion launcher API

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

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

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

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

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

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

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

4.6.4 XML parsing

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

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

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

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

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

The following changes are made in clause 8.2.3.2.1;

The description of the `lastError` property is changed from

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

to

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

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

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

to

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

4.6.7 errorCode(s) for onCSLaunch

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

1) The text introducing the table is changed from:

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

to

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

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

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

4.7 Clause 9 – System integration

4.7.1 Clarify URL scheme and MPD anchor rules

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

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

4.7.2 Clarify resource usage by HTML5 media elements

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

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

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

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

- the video in the two video elements either has the same frame rate, or one frame rate is an integer multiple of the other (see frame rate families in clause 10.4 of DVB-DASH)
- the video in the two video elements has the same colour primaries and transfer characteristics, e.g. BT.709 or BT.2020

4.7.4 Timing requirements for end of mid roll adverts

A new clause 9.6.12 is added as follows;

9.6.12 End of stream indication

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

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

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

A new clause 9.1.1.3 is added as follows:

9.1.1.3 Media player implementations and API behaviour

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

4.8 Clause 10 – Capabilities

4.8.1 Removing support for CSS3 navigation

Text is added in clause 10.2.2.1 as shown underlined.

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

- Allow applications to capture the events and prevent the default action (known as "JavaScript navigation").
- Handle CSS3 directional focus navigation when the nav-up, nav-right, nav-down and nav-left CSS properties are used by the application.

NOTE: Terminal support for these will be removed in a future version of the present document. Their use by application authors is discouraged. See Annex L.

- A default navigation mechanism provided by the terminal which shall allow focus to be moved between navigable elements and allow all navigable elements to gain focus.

4.8.2 XML Parsing

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

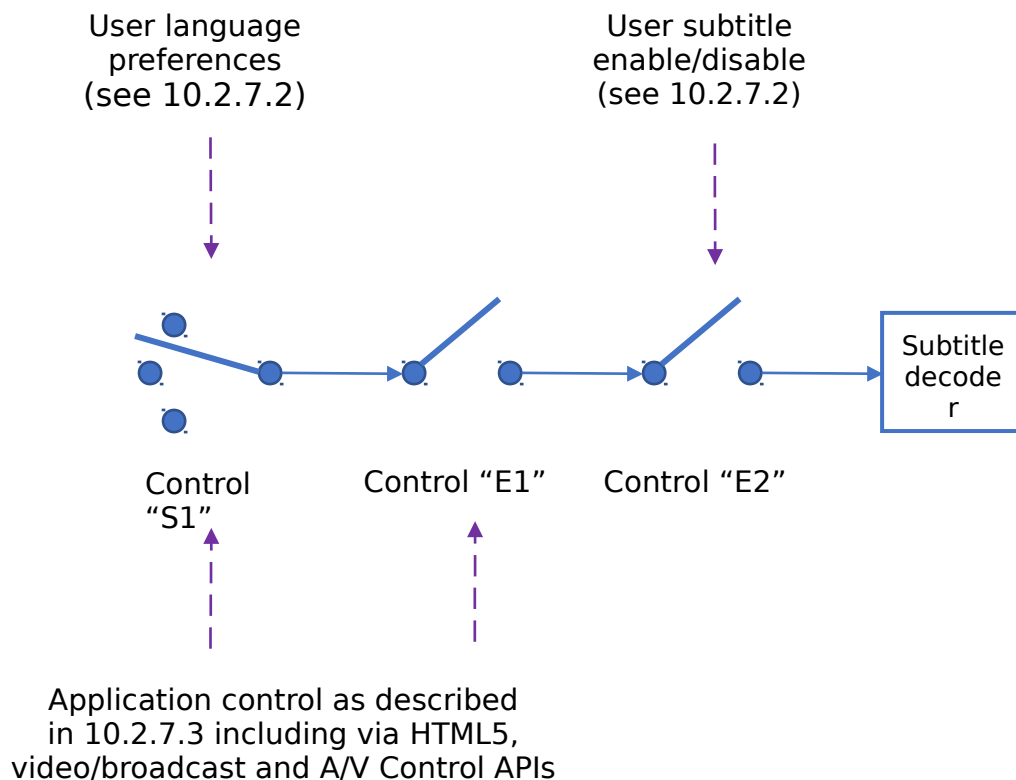
The “doctype” property of the “xmlCapabilities” property of the “application/oipfCapabilities” embedded object shall be null.

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

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

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

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



Selection control **S1** determines which of the available subtitle streams is selected for possible presentation. At any one time, this control can be is either under the control of the terminal, according to user language preferences (see clause 10.2.7.2) or it can be under

Confidential

application control and influenced by the relevant component selection APIs (see clause 10.2.7.3). The circumstances under which control passes from one to the other are described in clause 10.2.7.3.

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

Enabling control **E2** is the user's control over whether subtitles are presented. It can may be changed when the user changes terminal subtitle preferences. An HbbTV Application does not have any ability to control this setting but its current setting can may be read using the `subtitlesEnabled` property of the `Configuration` class (see A.2.20.1).

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

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

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

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

...

- a component, selected by the application, is being presented and is part of a video/broadcast object or an A/V Control object or an HTML5 media element or a `MediaSynchroniser` object (as appropriate) which is destroyed:
in which case component selection for that component type shall revert to the control of the terminal;
- a component of a particular type is being presented and the user makes a change using the terminal's subtitle/audio description (or other) selection mechanism relating to that component type:
- in which case component selection for that component type shall revert to the control of the terminal;

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

to “disabled” or “hidden”.

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

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

4.9 Clause 11 – Security

4.9.1 Update reference to TS 102 809

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

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

4.9.2 SHA-1 sunset date has passed

The following changes are made in clause 11.2:

- The sentence “Terminals shall cease to trust any signature that uses SHA-1 as the digest algorithm after 31st December 2016.” is removed.
- The two sentences “Mandatory until forbidden by SHA-1 sunset requirement specified above.” are both replaced by “Forbidden”.

4.10 Clause 12 – Privacy

4.10.1 Confusion over deviceId class

In clause 12.1.5, the following note is added between the first and second paragraphs;

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

4.11 Clause 13 - Media synchronization

4.11.1 Avoid MediaSynchroniser API leaking "secret" stream URLs

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

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

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

Confidential

- the value of the `contentId` property shall be the value of `contentIdOverride`, and
- the `contentIdStatus` shall be "final".
- When `contentIdOverride` is (or is set to) undefined or null then no override takes place and (The `contentId` and `contentIdStatus` properties shall correspond to the Content Identifier of the master media. For DVB broadcast services (and PVR recordings made from them) and MPEG DASH streams this shall be as defined in clause 5.2 of DVB Bluebook A167-2 [47]. For ISOBMFF and MPEG2 TS delivered via broadband:
 - the value of the `contentId` property shall be the absolute version of the URL provided by the HbbTV[®] application to specify the location of the media stream, before any redirect that may occur, and
 - the `contentIdStatus` shall be "final".

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

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

The other numbered notes in 13.6.2 are renumbered accordingly.

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

13.8.2.2 Synchronization timeline availability

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

The requested Synchronization Timeline shall be available if the requirements for determining the availability defined in clause 9.7.3 of the present document and clause 9.2 of DVB Bluebook A167-2 [47] are met and the requested Timeline is supported by the master terminal (see clause 13.4.2) and the master terminal has sufficient resources to decode the requested Timeline (see clause 13.4.2).

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

override takes place.

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

4.11.2 Clarification of TEMI timeline requirements

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

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

- Any component that is supported by the terminal for use with media synchronization and MPEG TEMI, i.e. audio, video and subtitles.
- Any component with stream_type 6 (private PES) and stream_id 1011 1101 ("private_stream_1") in the PES packet header, including, but not limited to, components where the PES packet payloads are empty.

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

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

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

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

4.12 Clause 14 – Companion screens

4.12.1 UPnP version to be used with DIAL

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

HbbTV® is a DIAL [50] application registered at the DIAL registry [i.8]. The registered name for HbbTV® applications is 'HbbTV'. For terminal and service endpoint discovery, the terminal shall support DIAL [50] except that the response to an M-SEARCH request, as specified by section 5.2 of DIAL [50], may be compliant with section 1.2.2 of UPnP Device Architecture 1.0 [67] instead of section 1.3.3 of UPnP Device Architecture 1.1 [68].

Confidential

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

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

Discovery Response

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

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

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

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

4.12.2 Clarification on XML AIT example

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

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

4.12.3 URL used in in DIAL example should be changed

In clauses 14.7.3.1 and 14.7.3.2, the references to

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

are replaced with

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

4.13 Annex A – OIPF DAE Specification Profile

4.13.1 Text referring to MetadataSearch in wrong location in annex A

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

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

It is moved from the row for the row for the “Basics” of the Programme class to the row for “The MetadataSearch class”.

4.13.2 "channels" should be "audioChannels"

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

4.13.3 Clarify resource usage by HTML5 media elements

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

If the resources that would be needed by an A/V Control object or a video/broadcast object are allocated to ~~in use by~~ an HTML5 media element (see clause 9.6.2), and the media element requiring the resource and the current media element owning the resource have not been added to the same media synchronizer object, then the request to present media through the object shall fail. For an A/V control object, the object shall go to playState 6 with the error property being 3, "insufficient resources". For a video/broadcast object, this shall be reported by an onChannelChangeError with errorState 11, "insufficient resources are available to present the given channel (e.g. a lack of available codec resources)".

4.13.4 VK_RECORD key is not in any set of keys

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

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

4.13.5 XML Parsing

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

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

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

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

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

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

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

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

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

A.2.1 Resource management

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

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

- If the resources needed for the request on the second object (suitable video decoder, suitable audio decoder and, suitable tuner if the second object is a video/broadcast object, suitable tuner) are not available then the request on the second object shall fail as defined by the API for the type of object concerned.
- If the resources needed for the request on the second object are available (e.g. the terminal has multiple audio and video decoders available to the HbbTV implementation) then the resources shall be allocated to the second object and the request shall not fail due to lack of resources (although it may fail for an another unrelated reason).
- If the request on the second object succeeds then the terminal shall present both objects at the same time without synchronisation. If applications wish to have multiple objects present media with synchronisation then the objects need to be added to a MediaSynchroniser object.

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

Confidential

to presenting a data service (see clause 7.2.6 of the present document) should not release the video or audio decoder.

NOTE 2: The policy for managing hardware resources defined here that applies to the A/V Control object and video/broadcast objects (first-come, first-served) is intentionally the exact opposite of the policy defined for the HTML 5 media element in clause 9.6.2 of the present document.

If a broadcast-related application that either

- does not include a video/broadcast object at all or
- includes a video/broadcast object that is in the unrealized state

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

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

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

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

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

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

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

~~then this binds the video/broadcast object to the channel whose audio is being presented.~~

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

Clause A.2.4.5 is extended with the following text;

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

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

4.13.7 Key “Label” should be “label”

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

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

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

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

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

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

If the current channel ~~currently being presented~~ is requested to be changed due to an action outside the application (for example, the user pressing the CH+ key on the remote) then any video/broadcast object ~~presenting~~ bound to that channel (i.e. in the connecting, presenting or

stopped states as the result of a call to `bindToCurrentChannel()` SHALL perform the same state transitions and dispatch the same events as if the channel change operation was initiated by the application using the `setChannel()` method.

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

- In clause 7.13.7.1, the definition of the property `currentChannel` is changed as shown;
The channel currently ~~being presented by~~bound to this embedded object (i.e. the object is in the connecting, presenting or stopped states as the result of a call to `bindToCurrentChannel()` if the user has given permission to share this information, possibly through a mechanism outside the scope of this specification. If no channel is ~~being presented bound to this embedded object~~, or if this information is not visible to the caller, the value of this property SHALL be null.

A new clause A.2.26 is added a shown.

A 2.26 Correction to the ApplicationPrivateData class

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

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

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

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

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

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

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

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

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

4.13.11 PVR API issues

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

The following changes are made in clause A.2.4.7.3:

1. The first instance of the property “ScheduledRecordingCollection getInProgressRecordings()” has the signature replaced by “function onPlayPositionChanged(Integer position)”. The definition of the property remains unchanged.
2. The second instance of the property “ScheduledRecordingCollection getInProgressRecordings()” is deleted.
3. The type of the playSpeed property is changed from Integer to Number.
4. The property onplaySpeedsArrayChangedonPlaySpeedsArrayChanged() is renamed onPlaySpeedsArrayChangedonPlaySpeedsArrayChanged() (i.e. “play”→”Play”).
5. The following text is added at the end of the clause - “In addition, the properties recordingState and onRecordingEvent defined in A.2.4.8.2 shall be supported.”

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

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

The following changes are made in clause A.2.4.8.2.

1. The definition of the “error” argument of the “onRecordingEvent” property is changed from “The current state of the recording. One of:” to “If the state of the recording has changed due to an error, this field contains an error code detailing the type of error. One of:”.
2. In the table listing error codes for the error argument of the onRecordingEvent property, the rows for values 5 and 6 (“Value not used” and “Recording has successfully completed”) are deleted.

4.13.12 Scope of AVAudio and AVVideoComponents

The following changes are made in clause A.2.4.6.

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

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

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

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

4.13.13 Allow applications to prioritise DRM to use

A new clause A.2.27 is added as follows:

A.2.27 Extensions to the application/oipfDrmAgent embedded object

This object shall be extended with the following additional method.

Boolean setActiveDRM(String DRMSystemID)			
Description	<p>Sets the DRM system, specified by <code>DRMSystemID</code>, that the terminal shall use with any new requests for playing protected broadband content. Any other DRM systems present in the terminal shall not be used with new requests until this method is called again even if this means playback of content fails.</p> <p>If the method is called with the <code>DRMSystemID</code> set to null, the algorithm used by the terminal to determine which DRM to use is outside the scope of the present document. The value <code>true</code> shall always be returned in this case. This shall be the default state if no calls to this method have been made.</p> <p>If for any reason the terminal is unable to set the specified DRM system as requested, the method shall return <code>false</code>, otherwise it shall return <code>true</code>.</p>		
Arguments	<table border="1"> <tr> <td style="vertical-align: top;"><i>DRM System ID</i></td> <td>The DRM system as defined in clause 9.3.10 of the OIPF DAE specification [1] and in Table 9 ("DRMControlInformation Type Semantics") of the OIPF Metadata specification [18].</td> </tr> </table>	<i>DRM System ID</i>	The DRM system as defined in clause 9.3.10 of the OIPF DAE specification [1] and in Table 9 ("DRMControlInformation Type Semantics") of the OIPF Metadata specification [18].
<i>DRM System ID</i>	The DRM system as defined in clause 9.3.10 of the OIPF DAE specification [1] and in Table 9 ("DRMControlInformation Type Semantics") of the OIPF Metadata specification [18].		

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

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

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

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

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

to:

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

4.13.15 Clarification to parental rating

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

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

is replaced with

For further information, see clause A.2.28.

A new clause A.2.28 is added as follows:

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

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

Table A.6: Encoding of parental rating

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

1

4.13.16 Reference to NI clause 10.1.1 of the DAE specification

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

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

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

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

4.13.17 Incoherent AVSubtitleComponent language with TTML tracks

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

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

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

DAE specification [1] as modified in the present document.

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

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

4.14 Annex E - Profiles of MPEG DASH

4.14.1 Clarify URL scheme and MPD anchor rules

The first paragraph of clause E.4.5 is extended with the text shown underlined.

When the URL of an MPD is referred to by an HbbTV Application, the URL may include MPD Anchors. Terminals shall support MPD Anchors using the 't' key of the URI fragment part as defined in clause C.4 of the MPEG DASH specification ISO/IEC 23009-1 [29] as profiled in clause 10.9.2 of the DVB DASH specification [45]. Support for other MPD Anchor keys is not required

4.14.2 Frame rate support with UHD content via DASH

A new clause E.4.6 is added as shown.

E.4.6 DASH specific aspects for video

Terminals shall support those frame rates in the 25Hz/50Hz family required by DVB-DASH [45]. There is no requirement to support frame rates in the 30Hz/60Hz family.

4.15 Annex G - Implementer guidelines for media synchronization

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

In clause G.1, the reference to DVB Bluebook A167-2 is replaced by a reference to ETSI TS 103 286-2.

4.16 New Annex L

4.16.1 Removing support for CSS3 navigation

A new annex L is added as shown.

Annex L (normative): Deprecated features

D.1 Introduction

Some of the features that have been required by previous versions of the present document have been deprecated. These features are listed in this Annex. Application authors should not use these features and terminals will not be required to support them in a future version of the present document.

D.2 CSS3 directional focus navigation

Confidential

The requirement for terminals to support CSS3 directional focus navigation (the nav-up, nav-right, nav-down and nav-left CSS properties) in clause 10.2.2.1 has been deprecated.