Errata 1 to TS 102 796 V1.4.1

2017-06-01

Table of Contents

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 obselete.

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

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

2 Conventions

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

3 Summary

The following table summarises the changes included in this document.

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
4913	Clarify resource usage by HTML5 media elements	Ambiguity	9.6.2, A.2.1	4.5.2, 4.9.3
5307	Clarifying media synchroniser error codes	Ambiguity	8.2.3.2	4.4.2
5324	Clarify URL scheme and MPD anchor rules	Ambiguity	9.2, E.4.5	4.5.1, 4.10.1
5864	text in wrong location in annex A - referring to MetadataSearch	Editorial	A.1	4.9.1
5910	UPnP version to be used with DIAL	Implementation	2.1, 14.7	4.1.2, 4.8.1
6049	Avoid MediaSynchroniser API leaking "secret" stream URLs	Other	8.2.3.2.1, 13.6.2, 13.8.2.2	4.4.1, 4.7.1
6544	missing XML declaration in example XML AIT	Error	7.2.3.2	4.3.3
6550	Clarification on XML AIT example	Error	7.2.3.2	4.3.2, 4.8.2
6560	Channel change key behaviour with broadcast independent applications	Ambiguity	6.2.2.2	4.2.1

Issue #	Short Description	Category	Clauses Impacted	Clause in this document
6591	"channels" should be "audioChannels"	Editorial	A.2.4.6	4.9.2
6610	Update reference to TS 102 809	Other	2.1, 7.2, 11.1	4.1.1, 4.3.1, 4.6.1

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 2 – References

4.1.1 Update reference to TS 102 809

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

4.1.2 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 Clause 6 – Service and application model

4.2.1 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).

4.3 Clause 7 – Formats and protocols

4.3.1 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.

4.3.2 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.

4.3.3 Missing XML declaration in example XML AIT

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

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

4.4 Clause 8 – Browser application environment

4.4.1 Avoid MediaSynchroniser API leaking "secret" stream URLs

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

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

4.4.2 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4.5 Clause 9 – System integration

4.5.1 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).

4.5.2 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.

4.6 Clause 11 – Security

4.6.1 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.

4.7 Clause 13 - Media synchronization

4.7.1 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:
>
> - 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 tThe `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.

4.8 Clause 14 – Companion screens

4.8.1 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].
>
> 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 respond~~s~~es 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
> ```

4.8.2 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>
```

4.9 Annex A – OIPF DAE Specification Profile

4.9.1 Text referring to MetadataSearch in wrong location in annex A

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

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

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

4.9.2 "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.

4.9.3 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)".

4.10 Annex E - Profiles of MPEG DASH

4.10.1 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