

ETSI TS 101 499 V3.2.1 (2023-07)



**Hybrid Digital Radio (DAB, DRM, RadioDNS);
SlideShow;
User Application Specification**



Reference

RTS/JTC-DAB-107

Keywordsaudio, broadcasting, DAB, digital, DNS, DRM,
radio**ETSI**650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

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

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important noticeThe present document can be downloaded from:
<https://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at <https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:
<https://portal.etsi.org/People/CommitteeSupportStaff.aspx>

If you find a security vulnerability in the present document, please report it through our Coordinated Vulnerability Disclosure Program:
<https://www.etsi.org/standards/coordinated-vulnerability-disclosure>

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.
The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2023.
© European Broadcasting Union 2023.
All rights reserved.

Contents

Intellectual Property Rights	6
Foreword.....	6
Modal verbs terminology.....	7
1 Scope	8
2 References	8
2.1 Normative references	8
2.2 Informative references.....	9
3 Definition of terms, symbols and abbreviations.....	9
3.1 Terms.....	9
3.2 Symbols.....	9
3.3 Abbreviations	9
4 Introduction	10
5 Application behaviour	10
5.1 Initialization	10
5.1.1 General.....	10
5.1.2 Application priority.....	11
5.1.3 Fall-back behaviour	11
5.2 Operation.....	11
5.3 Common Parameters	12
5.3.1 General.....	12
5.3.2 Trigger Time	13
5.3.3 ClickThroughURL.....	13
5.3.4 Expire time.....	13
5.3.5 Categorization.....	14
5.3.5.1 General	14
5.3.5.2 Category ID.....	14
5.3.5.3 SlideID	14
5.3.5.4 Category Title	14
5.4 Termination	14
6 Delivery using broadcast MOT	15
6.1 General	15
6.2 Parameters	15
6.2.1 General.....	15
6.2.2 ContentName	16
6.2.3 MOT ContentTypes and ContentSubTypes.....	16
6.2.4 TriggerTime.....	17
6.2.5 ExpireTime	17
6.2.6 CategoryID/SlideID	17
6.2.7 CategoryTitle	17
6.2.8 ClickThroughURL.....	17
6.2.9 AlternativeLocationURL	17
6.2.10 Alert.....	17
6.3 Updating parameters.....	18
6.4 Additional MOT Parameters	18
7 Delivery over IP	18
7.1 Application Discovery.....	18
7.1.1 General.....	18
7.1.2 Discovery requirements	19
7.2 Message structures	20
7.2.1 General.....	20
7.2.2 Text.....	20
7.2.3 Slide image	20

7.2.4	Metadata	20
7.3	Stomp transport	20
7.3.1	General.....	20
7.3.2	Subscribing to a destination.....	20
7.3.3	Receiving a message.....	21
7.3.3.1	General.....	21
7.3.3.2	destination.....	21
7.3.3.3	message-id.....	21
7.3.3.4	content-length.....	22
7.3.3.5	trigger-time.....	22
7.3.3.6	link	22
7.3.3.7	CategoryID.....	22
7.3.3.8	SlideID	22
7.3.3.9	CategoryTitle	22
7.3.4	Handling errors	22
7.4	HTTP transport.....	22
7.4.1	General.....	22
7.4.2	Request	22
7.4.3	Response.....	23
7.4.3.1	General.....	23
7.4.3.2	RadioVIS-Message-ID.....	24
7.4.3.3	RadioVIS-Destination.....	24
7.4.3.4	RadioVIS-Trigger-Time.....	24
7.4.3.5	RadioVIS-Link.....	24
7.4.3.6	RadioVIS-CategoryID.....	24
7.4.3.7	RadioVIS-SlideID	25
7.4.3.8	RadioVIS-CategoryTitle	25
7.4.4	Multiple Frame Response	25
7.4.5	JSONP Response	25
7.4.6	Implementation Requirements.....	25
7.4.7	Handling Responses.....	26
7.5	Void.....	26
7.6	Server-sent Events.....	26
7.6.1	General.....	26
7.6.2	Request	26
7.6.3	Response.....	26
7.6.3.1	General.....	26
7.6.3.2	scope	28
7.6.3.3	body.....	28
7.6.3.4	src.....	28
7.6.3.5	triggerTime.....	28
7.6.3.6	link	28
7.6.3.7	category.id.....	28
7.6.3.8	category.slideId	29
7.6.3.9	category.title.....	29
7.6.3.10	Category Keys.....	29
7.6.4	Heartbeat.....	29
7.6.5	Client identification	29
7.7	Topic Construction.....	29
7.7.1	General.....	29
7.7.1	Server-sent Events	30
7.7.2	Stomp and HTTP.....	30
8	Implementation guidelines for Service Providers	30
8.1	Images	30
8.2	TriggerTime	31
8.3	URL Handling.....	31
9	Device Profiles	31
9.1	Simple Profile.....	31
9.1.1	General.....	31
9.1.2	Storage and memory management.....	32

9.1.3	Display	32
9.2	Enhanced Profile	32
9.2.1	General.....	32
9.2.2	Storage and memory management.....	33
9.2.3	Display	34
9.3	Image Formats.....	34
9.3.1	Image/JFIF (JPEG)	34
9.3.2	Image/PNG	34
9.4	URL Handling.....	35
9.4.1	General.....	35
9.4.2	Click-Through URL.....	36
9.4.3	AlternativeLocationURL	36
Annex A (normative): APNG 1.0 Specification - Animated Portable Network Graphics.....		37
A.1	Introduction	37
A.1.1	Terminology	37
A.1.2	Error Handling.....	37
A.2	Structure	37
A.2.1	General	37
A.2.2	Chunk sequence numbers	38
A.2.3	'acTL': The Animation Control Chunk	38
A.2.4	'fcTL': The Frame Control Chunk	39
A.2.5	'fdAT': The Frame Data Chunk	40
A.3	Test encoder and sample images	41
Annex B (informative): Implementing SlideShow with time shifted audio services		42
B.1	Introduction	42
B.2	SlideShow in X-PAD	42
B.3	SlideShow in Packet Mode or X-PAD	43
B.4	Reference time and TriggerTime.....	43
Annex C (informative): Use Cases		44
C.1	AlternativeLocationURL.....	44
C.1.1	General	44
C.1.2	Faster image acquisition.....	44
C.1.3	Increased image relevance.....	44
C.1.4	Faster carousel rotation	45
C.2	ClickThroughURL	46
C.3	Categorized SlideShow	46
Annex D (normative): Methods for synchronizing SlideShow Reference Time.....		48
D.1	General	48
D.2	Delivery of SlideShow over DAB.....	48
D.3	Delivery of SlideShow over DRM	48
D.4	Delivery of SlideShow over IP.....	48
Annex E (normative): List of metadata properties		49
E.1	List of Server-sent Events properties	49
	History	51

Intellectual Property Rights

Essential patents

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

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

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

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

NOTE 1: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

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

The Eureka Project 147 was established in 1987, with funding from the European Commission, to develop a system for the broadcasting of audio and data to fixed, portable or mobile receivers. Their work resulted in the publication of European Standard, ETSI EN 300 401 [1], for DAB (see note 2) which now has worldwide acceptance.

NOTE 2: DAB is a registered trademark owned by one of the Eureka Project 147 partners.

The DAB family of standards is supported by World DMB, an organization with members drawn from broadcasting organizations and telecommunication providers together with companies from the professional and consumer electronics industry.

The RadioDNS Project was established in 2010 to standardize the combination of broadcast radio systems with additional applications, content and meta-data delivered over fixed or mobile IP networks. The project produced a specification using DNS to locate the broadcaster's Internet domain which is in use worldwide, and now standardized as ETSI TS 103 270 [6]. RadioDNS operates the authoritative name servers for the radiodns.org domain, and has members drawn from broadcasting organizations, manufacturers and service providers.

NOTE 3: "RadioDNS Hybrid Radio" and the RadioDNS Hybrid Radio logo are registered trademarks of RadioDNS Limited, a not-for-profit company owned by its members.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document describes an application that provides a visual accompaniment to a radio service.

In respect to previous versions of the present document, hybrid radio provisions have been added to allow a seamless experience for users when consuming radio services delivered by digital radio broadcasting systems (DAB, DRM) or IP or a combination of both. The use of the present document allows content to be created once by the service provider for delivery by both mechanisms and allows manufacturers to implement devices with many common elements.

The application can be delivered using broadcast or IP, or a combination of the two.

This revision adds a push mechanism for IP content, and adds DL Plus (ID3-like) capability to IP delivered text content.

2 References

2.1 Normative references

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

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

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

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

- [1] [ETSI EN 300 401](#): "Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers".
- [2] [ETSI TS 101 756](#): "Digital Audio Broadcasting (DAB); Registered Tables".
- [3] [ETSI EN 301 234](#): "Digital Audio Broadcasting (DAB); Multimedia Object Transfer (MOT) protocol".
- [4] [ETSI ES 201 980](#): "Digital Radio Mondiale (DRM); System specification".
- [5] [ETSI TS 101 968](#): "Digital Radio Mondiale (DRM); Data applications directory".
- [6] [ETSI TS 103 270](#): "RadioDNS Hybrid Radio; Hybrid lookup for radio services".
- [7] [ETSI TS 102 818](#): "Hybrid Digital Radio (DAB, DRM, RadioDNS); XML specification for Service and Programme Information (SPI)".
- [8] [ISO/IEC 15948](#): "Information technology -- Computer graphics and image processing -- Portable Network Graphics (PNG): Functional specification".
- [9] [ISO/IEC 10918 \(all parts\)](#): "Information technology -- Digital compression and coding of continuous-tone still images".
- [10] [ISO 8601:2004](#): "Data elements and interchange formats -- Information interchange -- Representation of dates and times".
- [11] [IETF RFC 2782](#): "A DNS RR for specifying the location of services (DNS SRV)".
- [12] [IETF RFC 4627 \(2006\)](#): "The application/json Media Type for Javascript Object Notation (JSON)".
- [13] [IETF RFC 4329 \(2006\)](#): "Scripting Media Types".

- [14] [IETF RFC 2616](#): "Hypertext Transfer Protocol -- HTTP/1.1".
- [15] [IETF RFC 2818](#): "HTTP Over TLS".
- [16] [IETF RFC 5246](#): "The Transport Layer Security (TLS) Protocol Version 1.2".
- [17] [Stomp Protocol Specification, Version 1.0](#).
- [18] [HTML Living Standard \(Section 9.2\)](#): "Server-sent events".
- [19] [ETSI TS 102 980](#): "Digital Audio Broadcasting (DAB); Dynamic Label Plus (DL Plus); Application specification".
- [20] [PNG \(Portable Network Graphics\) Specification, Version 1.2](#).

2.2 Informative references

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

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

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

- [i.1] ISO 3166-2:2007: "Codes for the representation of names of countries and their subdivisions -- Part 2: Country subdivision code".
- [i.2] [draft-daviel-http-geo-header-01.txt April 2000](#): "Geographic extensions for HTTP transactions".
- [i.3] IETF RFC 5909: "Network Time Protocol Version 4: Protocol and Algorithms Specification".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

device: hardware device or software client receiving the SlideShow application

image: PNG, JPG or APNG binary data

slide: image data, along with any associated parameters

SlideShow Reference Time: time held on the device, against which application time parameters are compared

3.2 Symbols

Void.

3.3 Abbreviations

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

APNG	Animated Portable Network Graphics
ASCII	American Standard Code for Information Interchange

DAB	Digital Audio Broadcasting
DL	Dynamic Label
DMB	Digital Multimedia Broadcasting
DNS	Domain Name Server
DRM	Digital Radio Mondiale
FIG	Fast Information Group
FQDN	Fully Qualified Domain Name
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
IBOC	In-Band On-Channel
ID	IDentifier
IP	Internet Protocol
ISO	International Standards Organization
JFIF	JPEG File Interchange Format
JPEG	Joint Pictures Expert Group
JSON	JavaScript Object Notation
JSONP	JavaScript Object Notation with Padding
MOT	Multimedia Object Transfer
MSC	Main Service Channel
NTP	Network Time Protocol
PNG	Portable Network Graphics
PPI	Pixels Per Inch
RDS	Radio Data System
SDC	Service Description Channel
SI	Service Information
SPI	Service and Programme Information
SRV	SeRVice (record)
TCP	Transmission Control Protocol
UI	User Interface
URL	Universal Resource Locator
UTC	Universal Time Coordinated
UTF	Unicode TransForm
VGA	Video Graphics Array
X-PAD	eXtended Programme Associated Data

4 Introduction

SlideShow is an application for devices that enables a Service Provider to provide a sequence of images for a Service. These may be used by the Service to visualize the audio being received, for example:

- A news programme complemented by photos from the events being reported.
- A music programme having each song accompanied by cover art of the current song.
- During an advertising break, or a promotional slot, showing images for advertising or promotional purposes.

5 Application behaviour

5.1 Initialization

5.1.1 General

The application should be automatically started when a SlideShow service is discovered for the current radio service through any of the following means:

- for DAB, reception of a SlideShow application definition in FIG0/13 [1];

- for DRM, reception of a SlideShow application definition in SDC data entity type 5 [4];
- for IP, a specific DNS SRV record.

When the application is started, or the radio service is changed, it is recommended that a device wait for up to 1 s in order to receive an image to show on the display. After that time, and until an image is received, the device may display a station logo at the most appropriate size, taken from the Service and Programme Information SI document [7].

The IP transport may also provide text information, which may be the only available transport of text for some services, whilst others may provide text via the broadcast channel (e.g. RDS Radio Text, DAB Dynamic Label, DRM text message).

5.1.2 Application priority

The Service Provider may provide zero or more DNS SRV records, indicating that the SlideShow service can be provided by more than one server. Each DNS SRV record has a Priority parameter, as an integer number, where a lower value indicates a higher priority.

If a broadcast application is being signalled, it shall be assumed to have a Priority = 100.

EXAMPLE: A Service Provider advertises the SlideShow in two ways:

- A DNS SRV record with Priority = 101
- A broadcast application - assumed Priority = 100

As the broadcast application has Priority 100, the preferred method of reception is from broadcast. IP cannot be used, unless the device is unable to receive the broadcast application.

5.1.3 Fall-back behaviour

If an error condition occurs, for instance a loss of IP connectivity or if the Service Provider indicates that a particular stream of content is not currently being provided over IP, then receiver should use broadcast to acquire the content. If the IP connectivity is subsequently restored, the device should attempt a reconnection.

EXAMPLE: A DAB Service Provider advertises the SlideShow on IP.
The Device subscribes to receive TEXT information.
The Service Provider returns an error indicating that TEXT is not currently provided over IP.
The Device shall fall back to using DAB Dynamic Label for text content for the remainder of the session.

5.2 Operation

The SlideShow user application works with one display only: it can display only one slide at a time. Two device profiles are defined (see also clause 9):

- **simple**
- **enhanced**

Two modes of operation are defined:

- **normal**
- **interactive**

The permitted combinations of profile and modes are summarized in table 1.

Table 1: Permitted Combinations of Profiles and Modes

Profile	Normal Mode	Interactive Mode
Simple	Yes	No
Enhanced	Yes	Yes

In the **normal mode**, images are presented at the TriggerTime specified on each image without any requirement for user interaction.

In the **interactive mode**, the device presents an overview of categories with titles received so far and allows the user to choose one category out of this list. Choosing a category allows the user to view all the slides received under this category.



Figure 1: Example of interactive mode UI for a touch screen device showing received categories on the left and the currently selected image (informative)

The following behaviour guidance is given:

- The user shall be allowed to switch to interactive mode.
- The device shall offer a menu view where the user shall be able to get an overview about the already received slide categories, which contain at least one displayable slide.
- When browsing through a category the user shall always know which slide (slide x of y) is currently present.
- Navigating to a certain slide may be possible via arrow keys, numbers, touchscreen (gestures), etc.
- The user shall be able to navigate into a certain category and back into the menu view.
- The user shall be allowed to leave the interactive mode (switch back to normal mode).

5.3 Common Parameters

5.3.1 General

Whilst the methods of delivery vary between broadcast and IP, there are a number of parameters common to both, as listed below:

- Trigger time
- Click-through URL
- Expire time
- Categorization

The function of these parameters are described in the following clauses. The implementation details of how these are signalled to a device are detailed in clause 6 for broadcast and clause 7 for IP.

5.3.2 Trigger Time

When the device is in **normal** mode, the presentation of each slide may be controlled by the Service Provider by using the **Trigger Time** parameter, given as a datetime or as the string value '**NOW**'.

Depending on the conditions, the device shall perform one of the five actions shown in table 2 for a slide sent by the Service Provider.

Table 2: TriggerTime Values and Behaviours

Value of Trigger Time	Behaviour
Greater than current SlideShow Reference Time	The image is intended for display at the specified point in the future. The device shall hold the image in the holding buffer until this time is reached. When the TriggerTime equals the SlideShow Reference Time, the image shall be displayed immediately.
Equal to the current SlideShow Reference Time	The image shall be shown immediately on the receiver's display.
Less than the current SlideShow Reference Time	The image shall be held in the holding buffer, but shall not be displayed.
Equals the string value: NOW	This value has special significance and indicates that the slide shall be shown immediately on the receiver's display. This value only applies at the instant of reception. An image in the holding buffer which gets updated with this value should be displayed only once unless a subsequent TriggerTime update of NOW is received.
No TriggerTime	The image shall be held in the holding buffer, but shall not be displayed. A subsequent TriggerTime update may be received for the slide, which may then apply one of the other conditions.

If the specified TriggerTime behaviour cannot be achieved (e.g. insufficient storage for the image), then the image shall not be displayed.

The TriggerTime has an accuracy of 1 s.

Methods for synchronizing the SlideShow Reference Time are given in annex D.

5.3.3 ClickThroughURL

This describes a URL that may be used by a device to respond to a user action (e.g. tapping the screen while the slide is displayed) to show a linked X(HTML) resource within a capable application on the device, e.g. an integrated web browser.

For example, a web page giving further information/content related to the slide.

The URL is specified as a string using UTF-8 encoding, up to a maximum of 512 bytes.

5.3.4 Expire time

This parameter specifies the datetime after which presentation of a slide is no longer valid. Once this is reached or passed, the device shall remove the slide from the display and any cache.

The Expire Time has an accuracy of 1 s.

A value for this parameter may only be provided once, and any subsequent updates shall be ignored.

5.3.5 Categorization

5.3.5.1 General

Images may be categorized so they can be browsed by the user in **interactive** mode.

When a slide is received containing a CategoryID/SlideID parameter value which matches that of any slide already in the Holding Buffer, the other slides with the same CategoryID/SlideID value shall be decategorized by setting their CategoryID/SlideID parameter value to 0x0000.

5.3.5.2 Category ID

An 8-bit number that uniquely identifies a Category. CategoryID shall not be 0x00, except to remove a previously delivered slide from a category.

5.3.5.3 SlideID

An 8-bit number in the range 0x01-0xFF that identifies the index of the slide within the Category. When browsed by the user, slides should be presented in the order specified by the SlideID.

The value of 0x00 shall not be used, except when decategorizing a slide.

5.3.5.4 Category Title

The CategoryTitle parameter is used to provide a user readable title for each CategoryID.

A CategoryTitle is updated by providing an object with an existing CategoryID and a new value for CategoryTitle.

Categories with CategoryID but with a null CategoryTitle shall not be shown to the user.

The CategoryTitle is a string using UTF-8 encoding, up to a maximum of 128 bytes.

5.4 Termination

The SlideShow application can be terminated either by the service provider or by user action. The screen should return to a relevant user display.

The Service Provider may signal Termination:

- For broadcast, by removal of the service/service component transporting the application.
- For IP, by expiration of the application's DNS SRV record.

NOTE: Changing services does not necessarily cause the application to terminate.

If the newly selected service has the same SlideShow as the previously selected service, the application should continue uninterrupted.

If the new selected service has a different or no SlideShow, images from the previously selected service should be removed from the display immediately.

6 Delivery using broadcast MOT

6.1 General

The SlideShow application uses the Multimedia Object Transfer (MOT) protocol [3].

The use of the SlideShow user application within a DAB ensemble shall be signalled by the use of FIG 0/13 as specified in ETSI EN 300 401 [1]. The UserApplicationType value is given in ETSI TS 101 756 [2]. SlideShow may be transported using an MSC packet mode data subchannel or MSC stream audio subchannel (X-PAD part).

The use of the SlideShow user application within a DRM channel shall be signalled by the use of SDC data entity type 5 as specified in ETSI ES 201 980 [4]. The application domain shall be set to "DAB" as given in ETSI TS 101 968 [5]. The UserApplicationType value is given in ETSI TS 101 756 [2]. SlideShow shall be transported using an MSC packet mode data stream.

No user application data bytes shall be conveyed. The receiver shall discard all user application data bytes.

Each image, together with its parameters is encoded as a single MOT object. The body data of the object contains the raw image data, and the required Header Parameters are attached to the object in order to signal parameters for the slide. This object is then segmented and transferred as per the broadcast system specification. This shall be done within MOT header mode only; MOT directories shall not be used.

Wireless broadcast channels may be disturbed and so bit errors may corrupt the objects. Therefore the objects should be repeated sufficiently, applying one of the repetition methods offered by the MOT protocol and/or the broadcast system itself.

The application provider shall transmit the segments of each MOT body contiguously, and shall not interleave segments of different MOT bodies. Header Updates may be interleaved between header segments and/or body segments, and the receiver shall correctly decode and process these updates which will usually be transmitted to update a TriggerTime parameter for a previously transmitted MOT object. The receiver shall also continue reassembly of the MOT object whose transmission was interrupted by the Header Update(s).

Annex D provides information on setting a suitable time reference.

Annex B provides important advisory information on timing issues for manufacturers when rendering SlideShow as part of an audio service, and particularly in the case where the audio may be significantly time-shifted through use of pause functionality, or recording and subsequent playback.

6.2 Parameters

6.2.1 General

Only the MOT parameters listed in table 3 may be used with the SlideShow application. All other parameters shall be ignored by the receiver.

Table 3: MOT Parameters

Parameter	Parameter Id	Specified in	Mandatory for service provider		Mandatory for receiver		Occurrence
			Normal mode	Interactive mode	Normal mode	Interactive mode	
ContentName	0x0C	MOT ETSI EN 301 234 [3]	Yes	Yes	Yes	Yes	Single
TriggerTime	0x05	The present document	No (if not present, the object shall be triggered by a "Header update" (see clause 6.3) or it will never be presented	No	Yes	Yes	Single
ExpireTime	0x04	The present document	No	No	No	Yes	Single
CategoryID/SlideID	0x25	The present document	No	Yes	No	Yes	Single
CategoryTitle	0x26	The present document	No	No (but has to be received at least once. see clause 5.3.5.4)	No	Yes	Single
ClickThroughURL	0x27	The present document	No	No	No	No	Single
AlternativeLocation URL	0x28	The present document	No	No	No	No	Single
Alert	0x29	The present document	No	No	No	Yes	Single

6.2.2 ContentName

This parameter is used for uniquely identifying the object for the purposes of cache management.

The ContentName shall be changed for each new slide, and the value shall never be reused in reference to different content.

On reception of an object with a ContentName that already exists in the Holding Buffer, the receiver shall overwrite the prior version with the newly received object. If this occurs and the object is also being displayed, the Display shall not be affected. If the object is an animated image, animation shall be stopped and the default image shall be displayed.

6.2.3 MOT ContentTypes and ContentSubTypes

Within the MOT protocol, it is mandatory for each object to be characterized by its ContentType and ContentSubType (see ETSI EN 301 234 [3] and ETSI TS 101 756 [2]). The following ContentType/ContentSubType pairs are the only ones permitted for the use in the SlideShow user application.

- **Image/JFIF (JPEG)**
- **Image/PNG**
- **MOT transport/Header update**
- **MOT transport/Header only**

Guidance on device implementation for decoding of JPEG and PNG is given in clause 9.3.

The MOT transport ContentSubType **Header update** is used to signal an update to parameters of a previously received object. This is explained in more detail in clause 6.3.

The MOT transport ContentSubType **Header only** is used to signal an object with no associated body data. It should thus only be relevant to IP-connected devices which can acquire the image data over IP.

6.2.4 TriggerTime

An optional parameter containing a value for the Trigger Time for the image as given in clause 5.3.2.

The value of the parameter field is coded in the form given in ETSI EN 301 234 [3], clause 6.2.4.1, Coding of time parameters. The milliseconds shall be set to zero. The TriggerTime shall always use UTC.

If an object should be presented as soon as it is received, the special TriggerTime value of "NOW" is used. This is indicated by setting the coded time parameter validity flag to 0.

6.2.5 ExpireTime

This parameter specifies the datetime after which presentation is no longer valid as given in clause 5.3.4.

The value of the parameter field is coded in the form given in ETSI EN 301 234 [3], clause 6.2.4.1, Coding of time parameters. The milliseconds shall be set to zero. The ExpireTime shall always use UTC.

This parameter may only be provided once and the receiver should ignore subsequent ExpireTime updates.

It is recommended to provide ExpireTime to allow for slides to be cached across the application lifetime. Slides without an ExpireTime shall be removed from the cache when the application is terminated.

6.2.6 CategoryID/SlideID

This parameter specifies the category and sequence order inside the category of the slide, as defined in clause 5.3.5. The CategoryID/SlideID is a 16-bit field divided into two 8-bit segments. The upper 8-bits represents the CategoryID, and the lower 8-bits represents the SlideID.

6.2.7 CategoryTitle

The CategoryTitle parameter contains a string value, as defined in clause 5.3.5.4.

6.2.8 ClickThroughURL

An optional parameter, containing a value for the Click-Through URL for the image as given in clause 5.3.3.

The length of this URL shall not exceed 512 characters.

6.2.9 AlternativeLocationURL

An optional parameter that allows devices with IP connectivity to acquire the image using an HTTP request to the URL specified in AlternativeLocationURL (see clause 9.4.3).

The URL is specified as a string using UTF-8 encoding, up to a maximum of 512 bytes.

6.2.10 Alert

The Alert parameter provides a means whereby the service provider may indicate an interruption to the interactive mode. It is encoded as an 8-bit integer. The following values are defined in table 4.

Table 4: Alert Values

Value	Meaning
0x00	Shall not be used
0x01	Emergency warning. Receiver shall switch back to the normal mode of presentation and may provide an appropriate user notification
0x02 to 0xff	Reserved for future use

NOTE: The presence of this parameter does not alter the Trigger Time requirements (see clause 5.3.2).

6.3 Updating parameters

The MOT protocol allows for an MOT object to be sent over the broadcast channel for the purposes of updating the Header Parameters of a previously sent object. The **ContentName** parameter value is used to match an update to a previously sent MOT object.

Only the TriggerTime and Category/SlideID parameters may be updated. All other parameters shall be ignored by the receiver.

The TriggerTime parameter carries the updated TriggerTime for the object specified by ContentName.

The CategoryID/SlideID parameter is used to change the category and/or ordering of a previously delivered slide or to decategorize the slide.

The slide being updated shall receive the updated CategoryID/SlideID value. Any other slide with the same CategoryID/SlideID value shall be decategorized.

If the update contains a CategoryID/SlideID with a value 0x0000, the slide shall be decategorized, effectively removing it from an interactive display mode.

6.4 Additional MOT Parameters

Because SlideShow is intended to be a simple application, the following optional MOT features shall not be used:

- Conditional Access on MOT level: scrambling on MOT level is limited to MOT directory mode. However, scrambling on subchannel or on datagroup level is permitted for the SlideShow.
- Compression on transport level (MOT level): images in JPEG and PNG formats are already compressed, so further compression on transport level would not be advantageous.
- MOT Directory: The MOT Directory shall not be transmitted, and therefore the caching functionality of Directory cannot be implemented. Clauses 9.1.2 and 9.2.2 specify memory management for caching content.

7 Delivery over IP

7.1 Application Discovery

7.1.1 General

For devices with Hybrid functionality, able to receive both broadcast and IP Services, as well as those able to receive only IP Services, methods may be made available by the Service Provider to receive programme related content (including the SlideShow application) over IP.

Three transport mechanisms are defined to deliver programme related content over IP. The recommended transport is Server-sent Events. The use of the Stomp and HTTP transports is deprecated but included in the present document to support legacy applications.

To support application discovery, a receiver shall be capable of resolving the authoritative FQDN for a service via the methodology defined in the RadioDNS Hybrid Radio specification [6].

A DNS SRV record request, for the TCP protocol, shall then be made against this authoritative FQDN using the service name specific to the relevant transport.

The present document defines three possible transports: **Server-sent Events**, **Stomp** and **HTTP**, using the SRV Record names shown table 5.

Table 5: SRV Record Names

Transport Used	SRV Record Service Name	Protocol
Server-sent Events	radiopush	tcp
Stomp (deprecated)	radiovis	tcp
HTTP (deprecated)	radiovis-http	tcp

The Server-sent Events transport is detailed in clause 7.6.

The Stomp transport is detailed in clause 7.3.

The HTTP transport is detailed in clause 7.4.

Each available transport shall be setup by the Service Provider as a separate DNS SRV record. For each transport, the port number within the SRV record returned defines the server port that a receiver should connect to for that transport.

If at least one SRV record is successfully returned for either transport, the service supports the delivery of content over IP, accessed using the associated transport on the host and port indicated in the relevant SRV record.

For example, for the Authoritative FQDN **rdns.example.org**, a DNS SRV record query is made to:

```
_radiopush._tcp.rdns.example.org
```

Using the `nslookup` tool, this would yield the following DNS SRV record:

```
service = 0 100 443 push.example.org.
```

This indicates that content delivered over IP can be accessed using the Server-sent Events transport on the host **push.example.org**, port **443**.

Note that more than one SRV record may be returned for a transport, with different values. This can potentially be used for load balancing purposes by providing different hosts/ports with different priorities/weightings. See the SRV record specification [11] for a more detailed explanation on handling DNS SRV records.

7.1.2 Discovery requirements

For Service Providers implementing delivery of programme related content over IP:

- Server-sent Events transport shall be implemented.
- Stomp transport may be implemented to support legacy devices.
- A transport may be implemented on more than one host and/or port.

For manufacturers and developers implementing reception of programme related content over IP:

- Server-sent Events transport shall be implemented.
- Stomp transport may be implemented to support legacy services.

NOTE: It is strongly recommended that transports are implemented on their standard ports.

7.2 Message structures

7.2.1 General

Each transport will carry a set of messages to a receiver over IP. These messages follow a specific format depending on whether the Service Provider is sending a text, a slide image or metadata payload.

7.2.2 Text

Provides a text message to be displayed on the receiver. The message shall not exceed 128 characters in length. A receiver receiving a text message longer than this may ignore the message.

A valid message shall be displayed immediately and shall replace any existing text message currently displayed on the receiver.

7.2.3 Slide image

Provides an HTTP URL referring to a slide image to be downloaded and displayed on the receiver.

The URL shall not exceed 512 characters in length. A receiver receiving a URL longer than this may ignore the URL.

7.2.4 Metadata

Provides structured metadata that the receiver may present in a format of its choosing.

The message is only available over the Server-sent Events transport.

7.3 Stomp transport

7.3.1 General

The Stomp transport is based upon the Stomp specification [17]. The version used may be negotiated between receiver and service provider as detailed in the specification, but both shall support and be backward compatible with version 1.0.

A receiver connects to the Stomp server and then subscribes to one or more destinations. Once the receiver subscribed, text-based frames are then received related to the chosen destination.

The destinations used in the Stomp transport are the topics as defined in clause 7.7, and are constructed from the broadcast parameters of the current service and desired content type.

Stomp messages are sent and received as frames. Each frame consists of a set of headers and a body. All frames should be encoded as UTF-8 with Unicode character encoding. All frames are terminated using a NULL ASCII character ^@ (control-@).

The following clauses define the essential frames used in RadioVIS. The Stomp specification [17] provides specific and detailed protocol definitions.

It is recommended the Service Provider allows receivers to receive Stomp messages without first having authenticated (anonymous access), and that the receiver does not provide any authentication parameters whilst connecting.

7.3.2 Subscribing to a destination

A device subscribes to a destination by using the SUBSCRIBE frame, as given in the Stomp specification. The header parameter **receipt** shall be added to this frame, given a value which may be used by the device to identify that destination.

```
SUBSCRIBE
destination: <topic>
receipt: <value>
```

```
^@
```

The device shall then wait for the service provider to respond with a receipt frame for this subscription request. If successful, the response frame will contain a header parameter **receipt-id** containing the same value as the parameter **receipt** in the original request frame.

```
RECEIPT
receipt-id: <value>
```

```
^@
```

Once the receipt frame is received, the device may continue subscribing to additional destinations.

If the frame returned from the service provider is not a valid receipt frame, a different frame (e.g. an error frame) or a receipt frame without a matching **receipt-id** for that destination, then the subscription shall be deemed to have been unsuccessful.

The device shall wait for a response up to a maximum of 10 s before assuming the subscription was unsuccessful.

If the service provider wishes to indicate to a device that a particular destination is not supported (e.g. if they support image acquisition over IP, but not text), a Stomp ERROR frame shall be returned.

If a device attempts to subscribe to a destination and is unsuccessful, it shall not make any additional attempts to subscribe to that destination in the current session. If an alternative means of acquiring the content represented by that destination over a broadcast bearer is available, this should be used.

7.3.3 Receiving a message

7.3.3.1 General

Once successfully subscribed to at least one destination, the receiver may receive message frames in the following format:

```
MESSAGE
destination: <topic>
message-id: <message-identifier>
content-length: <body byte length>
```

```
<body>^@
```

The **body** of this frame contains a message body, either one of:

- TEXT <text> - text as described in clause 7.2.2
- SHOW <url> - an image as described in clause 7.2.3

Each frame may contain headers, as detailed in the following clauses. A receiver may ignore any non-mandatory headers.

7.3.3.2 destination

A mandatory header that confirms from which topic the message has been received.

This may help differentiate when subscribed to multiple topics over the same connection, and shall be parsed by the receiver.

7.3.3.3 message-id

This reflects an ID allocation within the Service Provider. The allocation shall be unique for each individual message over all topics from the Service Provider for a period of at least 24 hours. A receiver shall not make any assumptions on the structure or sequence of this allocation.

7.3.3.4 content-length

An optional header, which may be used by the device as a byte count of the message body minus the terminator.

Parsing shall end when a null terminator occurs, regardless of the value of this header. If the content-length is missing then no length check should be performed.

7.3.3.5 trigger-time

An optional header sent with a **SHOW** message body, containing a value for the Trigger Time for the image as given in clause 5.3.2.

The datetime value for this header is given as an ISO 8601 [10] combined datetime.

7.3.3.6 link

An optional header sent with a **SHOW** message body, containing a value for the ClickThroughURL for the image as given in clause 5.3.3.

The length of this URL shall not exceed 512 characters.

7.3.3.7 CategoryID

An optional header, containing a value for CategoryID as specified in clause 5.3.5.2. If CategoryID is provided, it is mandatory to provide a SlideID (clause 5.3.5.3).

7.3.3.8 SlideID

An optional header, containing a value for SlideID as specified in clause 5.3.5.3. If SlideID is provided, it is mandatory to provide a CategoryID (clause 5.3.5.2).

7.3.3.9 CategoryTitle

An optional header, containing a value for CategoryTitle as specified in clause 5.3.5.4.

7.3.4 Handling errors

A device shall handle, but not show to the end-user, any errors and retain previously received text or image content.

If the device determines that it has lost its connection to the Stomp server, it shall attempt a reconnection, and subscribe to the destinations it previously successfully subscribed to, in the current session. The device shall not attempt to subscribe to any destinations that were unsuccessful.

7.4 HTTP transport

7.4.1 General

This transport is implemented as a HTTP request, which is opened and then held open until the server has a response to return. The server will send the response in a JSON/JSONP format and close the connection once sent.

The receiver should then send another request to the server to wait for the next response, unless the Service Provider does not respond with a message containing a valid Message ID, as detailed in clause 7.4.3.2.

7.4.2 Request

The HTTP request URL is constructed using both the FQDN of the server returned in the HTTP transport SRV record being used, and the following path structure:

```
/radiodns/vis/vis.json?topic=<topic>[&last_id=<last_id>][&callback=<callback>]
```

Table 6 defines the parameters used when constructing this URL.

Table 6: HTTP Request URL Parameters

Parameters	Description	Value	Status
topic	The topic the receiver wishes to subscribe to, as defined in clause 7.7.	string	Mandatory
last_id	Provide the message ID returned from the last HTTP transport response (see clause 7.4.3.2). This parameter may be omitted in the first request for the service, but shall sent in all subsequent requests using the last returned value.	string	mandatory for each request after the 1 initial response, otherwise omitted
callback	If given, this wraps the response in a JavaScript method style in order for it to be evaluated directly within JavaScript, as per the JSONP methodology (see clause 7.4.5).	string	optional

It is that a timeout period of at least 60 seconds be used on the request.

A receiver shall not treat a timeout as an error.

7.4.3 Response

7.4.3.1 General

The response format is a JSON representation of the message body, along with associated parameters, referred to as a **frame**.

An example frame is shown below:

```
{
  "headers": {
    "RadioVIS-Message-ID": "00192-c667a8",
    "RadioVIS-Destination": "/topic/fm/cel/c479/09580/image",
    "RadioVIS-Link": "http://www.capitalfm.com/onair",
    "RadioVIS-Trigger-Time": "NOW"
  },
  "body": "SHOW http://www.capitalfm.com/images/4abf.jpg",
}
```

The value of **body** within this frame contains a message body, either one of:

- TEXT <text> - text as described in clause 7.2.2
- SHOW <url> - an image as described in clause 7.2.3

The following headers in a frame are valid:

- RadioVIS-Message-ID
- RadioVIS-Destination
- RadioVIS-Trigger-Time
- RadioVIS-Link

- RadioVIS-CategoryID
- RadioVIS-SlideID
- RadioVIS-CategoryTitle

It is recommended that both receivers and Service Providers use the same case as specified above.

The Expire Time is not provided in the frame headers. The HTTP Expires header field of the slide is used to indicate the expiry time of the slide.

7.4.3.2 RadioVIS-Message-ID

An optional header sent back with every frame.

This reflects an ID allocation within the Service Provider, which may be used to identify whether the message within the frame is the latest message sent. The allocation shall be unique for each individual message over all topics from the Service Provider for a period of at least 24 hours. A receiver shall not make any assumptions on the structure or sequence of this allocation.

If the Service Provider includes this within any frame in the response, the receiver should immediately make another request to the Service Provider, with the value of this header included by the receiver as the **last_id** query string parameter.

If multiple frames are returned in a response, the value used by the receiver should be the *most recent* value, i.e. the value specified in the last processed frame.

If the Service Provider does not include this header in the response, the receiver shall not make another request to the service provider until the receiver stops receiving the service.

If the **last_id** parameter is included in the request to the server, and the server identifies that this does not identify the latest sent messages for the requested topics, it may respond with any intermediate messages in order to bring the receiver up-to-date, taking into account the upper limit on number of frames and total response size.

If no **last_id** parameter is included in the request to the server (i.e. for the initial request), or a value is given that the server does not recognize or cannot determine which message it corresponds to, the server shall send back the latest sent message(s) for the requested topic(s) to the receiver.

7.4.3.3 RadioVIS-Destination

A mandatory header sent back with every frame and confirms from which topic the message has been received.

This may help differentiate when subscribed to multiple topics over the same connection, and shall be parsed by the receiver.

7.4.3.4 RadioVIS-Trigger-Time

An optional header sent with a **SHOW** message body, containing a value for the Trigger Time for the image as given in clause 5.3.2.

The datetime value for this header is given as an ISO 8601 [10] combined datetime.

7.4.3.5 RadioVIS-Link

An optional header sent with a **SHOW** message body, containing a value for the Click-Through URL for the image as given in clause 5.3.3.

The length of this URL shall not exceed 512 characters.

7.4.3.6 RadioVIS-CategoryID

An optional header, containing a value for CategoryID as specified in clause 5.3.5.2. If RadioVIS-CategoryID is provided, it is mandatory to provide a RadioVIS-SlideID (clause 7.4.3.7).

7.4.3.7 RadioVIS-SlideID

An optional header, containing a value for SlideID as specified in clause 5.3.5.3. If RadioVIS-SlideID is provided, it is mandatory to provide a RadioVIS-CategoryID (clause 7.4.3.6).

7.4.3.8 RadioVIS-CategoryTitle

An optional header, containing a value for CategoryTitle as specified in clause 5.3.5.4.

7.4.4 Multiple Frame Response

A response may optionally be returned by the Service Provider as a set of frames within a JSON array.

The Service Provider may also use this to return a set of messages back to the receiver, e.g. to pre-load a set of slides with a trigger-time set in the future.

An example of such a response is shown below:

```
[
  {
    "headers": {
      "RadioVIS-Message-ID": "a46a8-bcd89",
      "RadioVIS-Destination": "/topic/fm/ce1/c479/09580/image",
      "RadioVIS-Link": "http://www.capitalfm.com/onair",
      "RadioVIS-Trigger-Time": "NOW"
    },
    "body": "SHOW http://www.capitalfm.com/images/4abf.jpg",
  },
  {
    "headers": {
      "RadioVIS-Message-ID": "ee789-de901",
      "RadioVIS-Destination": "/topic/fm/ce1/c479/09580/image",
      "RadioVIS-Link": "http://www.capitalfm.com/commercial",
      "RadioVIS-Trigger-Time": "2012-03-20T11:15:46.271Z"
    },
    "body": "SHOW http://www.capitalfm.com/images/commercial.jpg",
  }
]
```

A JSON Array is ordered, and the Service Provider shall return the response in a time-ordered way from oldest to most recent.

A receiver shall process the array in a time-ordered way such that it starts at the oldest frame and proceeds to the newest frame. The receiver is not obliged to process all the frames returned in the response.

7.4.5 JSONP Response

JSONP (JSON with Padding) is a means of providing cross-domain access to JSON data from a web service by wrapping that data within a javascript function call.

Should the receiver wish to be returned JSONP by making a request with the parameter **callback**, the server will return the JSON response given in clauses 7.4.3 and 7.4.4 wrapped within a function call with the same name as given in the **callback** parameter.

For example, for a request made with a callback method of **onCometResponse**, the HTTP response body would be of the form:

```
onCometResponse (<JSON>)
```

Where **JSON** is the JSON data given in clauses 7.4.3 and 7.4.4.

7.4.6 Implementation Requirements

For a JSON response, the Content Type parameter in the HTTP response shall be set by the Service Provider to **application/json** [12]. For a JSONP response, this shall be set to **application/javascript** [13].

Any response from the Service Provider shall not exceed 8 message frames, or a response body content length of 16 kB, whichever is first reached.

7.4.7 Handling Responses

The HTTP response may include any valid response within the HTTP specification, meaning a receiver should properly handle responses with common HTTP status code, including following any indicated redirects.

A device shall handle, but not show to the end-user, any errors and retain previously received text or image content.

If a device makes a request for a topic and receives the HTTP status code 404, it shall not make any additional attempts to make a request to that topic again in the current session. If an alternative means of acquiring the content represented by that topic over a broadcast bearer is available, this should be used.

7.5 Void

7.6 Server-sent Events

7.6.1 General

The Server-sent Events transport is defined in the Server-sent Events specification [18]. An HTTP request is performed and the connection held open.

Zero or more events will then follow over time as part of the same multi-part response.

All object keys in any Server-side Events JSON payload are case sensitive: clients and servers shall respect the case of all object keys.

7.6.2 Request

The Server-sent Events request URL is constructed using both the FQDN of the server returned in the Server-sent Events transport SRV record being used, and the following path structure:

```
/radiodns/push/3/<topic>
```

The value of **topic** is defined in clause 7.7.

7.6.3 Response

7.6.3.1 General

The structure of a Server-sent Events response consists of key: value pairs. An event is terminated by an empty line.

The initial response will include zero or more events.

The connection shall remain open and further messages will be received over the same connection when sent by the Service Provider.

The event fields **id**, **event** and **retry** are defined in the Server-sent Events specification [18].

The event field **data** is a JSON object.

An example event is shown below:

```
EXAMPLE 1:  id: <id>
            event: <event>
            retry: <retry>
            data: <data>
            ↵
```

where ↵ indicates an empty line.

The value of **event** is a valid content-type from one of the following:

- text;
- image;
- meta.

The client will only receive an event for the specified content-type in the original request (see clause 7.6.1). If no content-type is specified in the request, the client may receive responses containing any content-type.

The following properties of the **data** field JSON object are valid for all event types:

- scope.

The following properties of the **data** field JSON object are only valid for **event** type text:

- body.

An example text event is shown below:

```
EXAMPLE 2: id: 1
           event: text
           retry: 15000
           data: {
             "scope": ["dab:ce1.c123.c456.0"],
             "body": "You are listening to Example Radio"
           }
           ↵
```

The following properties of the **data** field JSON object are only valid for **event** type image:

- src
- triggerTime
- link
- category
 - category.id
 - category.slideId
 - category.title

An example image data field JSON object is shown below:

```
EXAMPLE 3: id: 2
           event: image
           retry: 15000
           data: {
             "scope": ["fm:ce1.c456.09890", "dab:ce1.c123.c456.0"],
             "src": "http://slides.example.org/image.jpg",
             "triggerTime": "NOW",
             "link": "http://example.org/",
             "category": {
               "id": 100,
               "slideId": 32,
               "title": "News"
             }
           }
           ↵
```

NOTE 1: `category.id`, `category.slideId` and `category.title` refer to Categorised Slideshow. They are nested properties of an object under the root level `category` property.

NOTE 2: `triggerTime` and `slideId` are examples of mixed case object keys: all object keys are case sensitive.

The following properties of the **data** field JSON object are valid for **event** type meta:

- Any category Key as defined in annex E.

An example meta data field JSON object is shown below:

```
EXAMPLE 4:  id: 3
            event: meta
            retry: 15000
            data: {
                "scope": [ "fm:cel.c456.09890", "dab:cel.c123.c456.0" ],
                "item": {
                    "artist": "Paolo Nutini",
                    "title": "Last Request",
                    "album": "These Streets",
                    "artwork": "http://example.org/artwork.jpg"
                }
            }
            ↵
```

7.6.3.2 scope

A mandatory data property sent back with every event. The value is an array that contains one or more bearers confirming which service(s) the event relates to.

NOTE: This is equivalent to the destination header in Stomp and HTTP transports.

7.6.3.3 body

For a text type event, this is a mandatory property containing Text, as defined in clause 7.2.2.

7.6.3.4 src

For an image type event, this is a mandatory property containing a Slide image as defined in clause 7.2.3.

7.6.3.5 triggerTime

An optional property sent with an image type event, containing a value for the Trigger Time for the image as given in clause 5.3.2.

The datetime value for this property is given as an ISO 8601 [10] combined datetime.

7.6.3.6 link

An optional property sent with an image type event, containing a value for the Click-Through URL for the image as given in clause 5.3.3.

The length of this URL shall not exceed 512 characters.

7.6.3.7 category.id

An optional property sent with either an image or text type event, containing a value for CategoryID as specified in clause 5.3.5.2. If `category.id` is provided, it is mandatory to provide `category.slideId` (clause 7.6.3.8).

7.6.3.8 category.slideId

An optional property, containing a value for SlideID as specified in clause 5.3.5.3. If category.slideId is provided, it is mandatory to provide categoryId (clause 7.6.3.7).

7.6.3.9 category.title

An optional property, containing a value for CategoryTitle as specified in clause 5.3.5.4.

7.6.3.10 Category Keys

Any value associated to a category key may be a JSON formatted number, string, array of numbers and/or strings, or null.

An array shall indicate multiple values are applicable to this category key.

Once a category key has been received, the contents may be persisted until a new matching category key is received or a null value is received.

7.6.4 Heartbeat

A message shall be sent by a Service Provider at least every 20 seconds. A receiver failing to receive any message within 30 seconds should attempt to reconnect. A service provider that experiences a socket error when attempting to send to a receiver shall end the connection.

When a Service Provider has no relevant message to send within 20 seconds, a heartbeat message may be sent. The heartbeat message shall consist of a single empty key: value pair, designated as a single colon character, as shown below:

```
:
↵
```

7.6.5 Client identification

The methods for client identification over HTTPS are given in ETSI TS 103 270 [6], clause 9. This is applicable to Server-sent Events.

7.7 Topic Construction

7.7.1 General

When constructing a topic for a transport, the following shall be implemented:

- The value of **ServiceIdentifier** is given by the RadioDNS Hybrid Radio [6] ServiceIdentifier, and specific to the means by which the service is being received. Where required, the <separator> element will be a forward slash character ('/').
- It shall be assumed that systems handling topics are case sensitive and therefore topics shall be entirely in lowercase.
- A Service Provider may also support a wildcard request, where the **ServiceIdentifier** is replaced with an asterisk "*" character, meaning the client requests to receive events for all services provided by the Service Provider. If a client receives an unknown resource response from a wildcard request, this means it is not supported by the Service Provider.
- Clients are strongly recommended not to make a wildcard request unless absolutely necessary. It is expected to be useful only in circumstances where the client is part of an aggregation platform. End user devices shall only request a specific **ServiceIdentifier**.

- Server-sent Events support Client Identification (clause 7.6.5). Client Identification is not possible using the Stomp or HTTP transport.

7.7.1 Server-sent Events

For Server-sent Events, the topic has the format:

```
<ServiceIdentifier>/<content-type>
```

The value of **content-type** for the Server-sent Events transport may be one of the following:

- image
- text
- meta

Alternatively, omitting the **content-type** shall result in all available content types being returned.

NOTE: A topic may have a variable number of path segments, reflecting that the **ServiceIdentifier** IBOC mId element, the **ServiceIdentifier** DAB UAtype element and **content-type** are optional.

EXAMPLE 1: An IBOC **ServiceIdentifier** without SPS and a **content-type** of image:
hd/292/07a26/image

EXAMPLE 2: An IBOC **ServiceIdentifier** with SPS value of 2 and no **content-type**:
hd/292/07a26/2

EXAMPLE 3: An IBOC **ServiceIdentifier** with SPS value of 2 and a **content-type** of meta:
hd/292/07a26/2/meta

7.7.2 Stomp and HTTP

For Stomp or HTTP transports, the topic has the format:

```
/topic/<ServiceIdentifier>/<content-type>
```

The value of **content-type** for Stomp and HTTP transports shall be one of the following:

- image
- text

8 Implementation guidelines for Service Providers

8.1 Images

Images may be made available at any size, shape and colour depth allowed by the image formats supported subject to the restrictions in clause 9.3. Pixels are always square.

The service provider should consider how images will be rendered by devices supporting both the simple and enhanced profile, taking into account the variations in maximum image data size, display colour depth, scaling and cropping, and caching.

Service providers are strongly recommended to implement customized sizing of images in response to the HTTP Header information provided by devices (see clauses 8.3 and 9.4).

8.2 TriggerTime

If the SlideShow application uses the TriggerTime parameter with values other than "NOW", it shall only be displayed on devices which have a synchronized SlideShow Reference Time as specified in annex D.

8.3 URL Handling

The following general guidance is given with regard to URL parameters:

- Service providers shall not refer to URLs with any scheme other than **http** [14]/**https** [15] and [16].
- Service providers are recommended to make use of the HTTP request header **User-Agent** sent by the device, in order to assist in returning a more appropriate resource for that device.
- Service providers are recommended to follow standard HTTP practices for caching in order to help minimize bandwidth usage, particularly by the use and handling of the following HTTP headers:
 - Expires
 - Last-Modified
 - If-Modified-Since
- At the Expires time, the device will request the image again. If the Service Provider returns a 404 response, the image will be immediately removed from the cache and the display.
- Service providers are recommended to make use of the HTTP headers shown in table 7 sent by the device, in order to assist in returning a more appropriate resource for that device.

Table 7: HTTP headers to assist in returning a more appropriate resource

Header Name	Description	Mandatory
Display-Height	Visible device screen height, in pixels	No (default is 240 pixels)
Display-Width	Visible device screen width, in pixels	No (default is 320 pixels)
Display-PPI	Visible device pixel density, in Pixels Per Inch (PPI)	No (default is 72 PPI)
geo.position	Single space-separated point of latitude and longitude in WGS84 coordinate reference system	No
geo.region	Geographical region, taken from the reserved list in ISO 3166-2 [i.1]	No

9 Device Profiles

9.1 Simple Profile

9.1.1 General

The **simple** profile is most suitable for devices with very restricted resources. It defines a simple device behaviour that receives, decodes, renders and displays a single PNG [8] or JPEG [9] image at a time.

9.1.2 Storage and memory management

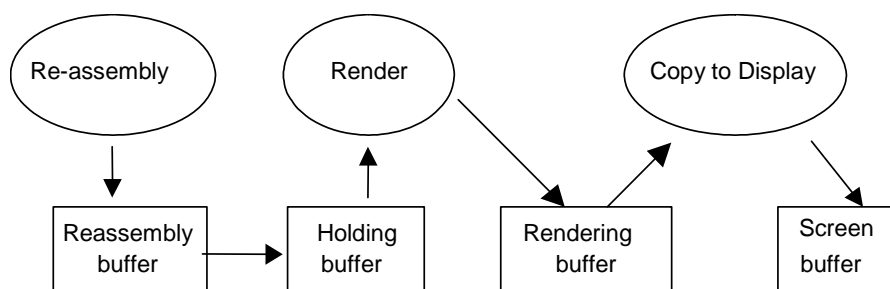


Figure 2: Simple profile Buffer management model for a SlideShow decoder

The successful reception of a slide causes the object to pass from the Reassembly Buffer to the Holding Buffer, overwriting any object previously stored there. This happens regardless of whether the rendering of the previous object has been completed or not.

The image is decompressed from the Holding Buffer to a bitmap format in the Rendering Buffer which is only large enough to hold a single image.

All buffers shall be cleared when the application is terminated.

At the TriggerTime, the bitmap is moved from the Rendering Buffer to the Screen Buffer/Display and is presented to the user.

A Header Update for a slide within the simple profile shall only apply to the current slide, and any other updates shall be ignored.

A receiver shall be able to decode images up to a file size (JPEG or PNG) of 50 kbytes (51 200 bytes). The Holding Buffer shall be large enough for one image. If an image exceeds this size, or is signalled to do so, it may be ignored by the receiver.

NOTE: This requirement implies a minimum Assembly Buffer size in excess of 50 kbytes (51 200 bytes).

9.1.3 Display

Receivers shall be able to display an image at a resolution of 320×240 pixels at a colour/grey scale depth of 8 bits per pixel (¼-VGA). If a receiver cannot display an image natively at this resolution, it is permitted to rescale it provided the aspect ratio is maintained and the image is fully visible.

If a slide is received which is smaller than 320×240 pixels, then it shall be displayed in the centre of the screen surrounded by a black background if needed.

If a slide is received which is larger than 320×240 pixels, it may be cropped by the receiver or not displayed at all. Receivers are only permitted to crop at the right hand side and at the bottom of the image.

9.2 Enhanced Profile

9.2.1 General

The **enhanced** profile extends the **simple** profile by defining control mechanisms to greatly enhance the presentation to the user. This includes giving a Service Provider the ability to present the slides independently of the transmission order and to transmit a series of slides to be cached by the device in advance of the display time, thus allowing the images to be more reliably and precisely timed. Animation of PNG images is also enabled.

9.2.2 Storage and memory management

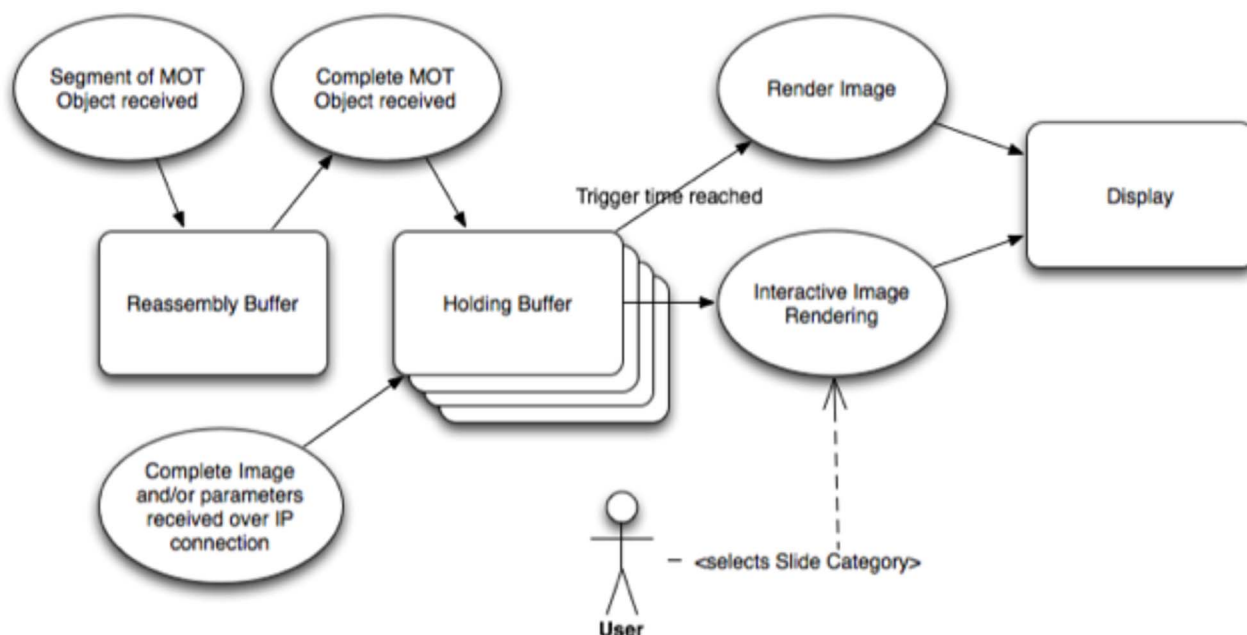


Figure 3: Enhanced profile Buffer management model for a SlideShow decoder including interactive mode

If the object is delivered using MOT, the successful reception of an object causes the object to pass from the Reassembly Buffer to the Holding Buffer which is large enough to hold multiple images along with their relevant parameters.

If the image and/or parameters is delivered using HTTP, the image/parameters are stored directly into the Holding Buffer.

A receiver shall be able to decode images up to a combined size of the image data and its associated parameters of 450 kbytes (460 800 bytes). The Holding Buffer shall be at least 450 kbytes (460 800 bytes) and be able to store between 1 and 64 images. When multiple images are stored, each image may be a different size and/or colour depth.

NOTE: This requirement implies a minimum Assembly Buffer size of 450 kbytes (460 800 bytes).

If the Holding Buffer has insufficient space to store a newly reassembled image, the receiver should delete images, with the following priorities, one by one until sufficient space has been freed:

- Slides with an ExpireTime prior to SlideShow Reference Time.
- Slides with no TriggerTime nor Category/Slide ID (or Category/Slide ID = 0).
- The slide with the earliest TriggerTime (in the past), with no Category/Slide ID (or Category/Slide ID = 0).
- The oldest received slide with a valid Category/Slide ID, with no TriggerTime or a TriggerTime prior to the SlideShow Reference Time.

In the **normal mode**, at TriggerTime the image is decompressed from the Holding Buffer to a bitmap format and then immediately copied to the Display.

In the **interactive mode**, the holding buffer makes slides available based on the selection of categories by the user; the required image is decompressed from the Holding Buffer to a bitmap format and then immediately copied to the Display.

Buffers shall not be cleared in the event of a temporary interruption to the service through poor reception; the most recently displayed image shall remain on the screen.

When the application is terminated, either by a change in the signalling over broadcast or IP, or by the user ending the SlideShow application (e.g. by tuning the device to a different service) the device may clear all image data in the buffers.

However, if the Holding Buffer contains slides with an `ExpireTime` parameter at a value greater than the SlideShow Reference Time, devices may preserve the image information of these slides in the Holding Buffer even when the device is tuned to another service. This allows implementations to provide a better user experience should the user tune back to the service at a later point, avoiding a "cold start" of the SlideShow with empty buffers.

9.2.3 Display

Receivers are strongly recommended to implement a display equal to or larger than 320×240 pixels, at a colour depth of at least 15 bits per pixel. Receivers shall not implement SlideShow on displays smaller than 160×120 pixels.

The SlideShow application display may be rotated to best fit the physical display aspect ratio (portrait or landscape), assuming that the majority of content will be formatted to fit a landscape display. However the orientation of the SlideShow application display shall be consistent across all services, and individual images received by the application shall not be rotated on a case-by-case basis.

The original aspect ratio of the image shall always be preserved.

Images may be scaled at factors of 150 % or greater in order to maximize the available physical display space.

It is mandatory to implement a scale factor of 50 %, and this is the only downscaling factor permitted.

The use of anti-aliasing and similar techniques is strongly recommended to optimize the quality of the scaled images.

9.3 Image Formats

9.3.1 Image/JFIF (JPEG)

For a JPEG format image, all receivers shall conform to the following restrictions:

- a receiver shall support baseline coding as a minimum;
- a receiver need not support progressive and/or multiscan coding;
- a receiver need not support arithmetic entropy coding;
- a receiver shall support JPEG files with up to 4 components (colour channels) at a resolution of up to 8 bits/component.

A receiver shall ignore any images it is unable to decode.

9.3.2 Image/PNG

For a PNG format image, all receivers shall display the default frame of the image.

A still PNG image shall conform to version 1.1 of the PNG specification (ISO/IEC 15948 [8]), and shall be supported by all receivers.

An animated PNG image shall conform to version 1.0 of the APNG specification, as defined in annex A. The minimum frame display time, determined by the parameters `delay_num` and `delay_den` in the Frame Control Chunk (`fCTL`) shall not be less than 100 ms. This enforces a maximum frame-rate for animation of 10 frames per second. If a receiver is unable to support this frame-rate, it shall display the default frame of this image and not attempt animation.

A receiver shall ignore any images it is unable to decode, and shall ignore any extension chunks within the image that it cannot decode.

9.4 URL Handling

9.4.1 General

The following general guidance is given with regard to device handling of URL parameters:

- Devices shall ignore URLs with any scheme other than **http [14]/https [15] and [16]**.
- Devices are recommended to include the HTTP request header **User-Agent** when acquiring an image or resource from a URL, to assist in returning a more appropriate resource for that device. The value of this header shall not contain any user-identifiable information.
- Devices shall handle HTTP status codes denoting errors or resource redirections in the expected way, and avoid unnecessary interruption to the user experience. It is recommended that the device follow standard best-practice when dealing with status codes other than 200 (OK), including limiting the number of redirections followed.
- Devices are recommended to follow standard HTTP best-practice for caching in order to help minimize bandwidth usage, particularly by the use and handling of the following HTTP headers:
 - Expires
 - Last-Modified
 - If-Modified-Since
- At the Expires time, the image shall be reacquired using HTTP. If a 404 response is received, the image shall be removed from both the cache and display immediately.

Devices are recommended to include the following HTTP headers in their request when acquiring images from a URL, indicating the display dimensions and resolution density of the device. This is to assist in returning a more appropriate resource for that device.

Table 8

Header Name	Description	Mandatory
Display-Height	Visible device screen height, in pixels	No (default is 240 pixels)
Display-Width	Visible device screen width, in pixels	No (default is 320 pixels)
Display-PPI	Visible device pixel density, in pixels per inch (PPI)	No (default is 72 PPI)

Once the image has been acquired and is to be displayed, the device shall examine the dimensions and content type of the image, as these may differ from that requested.

Devices are recommended to include their geographical location, if available, to enable the service provider to return a more appropriate resource. This should take the form of additional parameters in the HTTP request for the resource, as per [i.2].

Table 9

Header Name	Description	Mandatory
geo.position	Geographical coordinates as ordered Latitude and Longitude separated by a semicolon (";")	No
geo.region	Geographical region, taken from the reserved list in ISO 3166-2 [i.1]	No

9.4.2 Click-Through URL

A device without the means to display X(HTML) content shall ignore a Click-Through URL parameter on a slide and not indicate to the user that one is available, nor perform any actions on user interaction.

If the device has the means to display X(HTML) content, for example by using a web browser, then it should switch to this content on user interaction. The user shall then actively choose to switch back to the SlideShow interface in order to resume viewing slides.

The device is recommended to continue receiving the SlideShow service in the background, and may or may not decide to continue receiving the broadcast audio based on whether there is any audio associated with the content at the URL location. The device may choose to use user preference when making this decision.

9.4.3 AlternativeLocationURL

The AlternativeLocationURL provides a URL from which a more appropriate version of the image can be retrieved. The service provider may choose to send more appropriate content for the device from this URL - for example, based on the user agent and other headers of the HTTP request, or on the device's apparent location.

If the device has IP connectivity, it should immediately and then periodically attempt to retrieve the image from the AlternativeLocationURL until it is either successfully received (HTTP Status Code 200) or the device receives an error (HTTP Status Code 404) indicating that the image will never be provided. The MOT Body should continue to be assembled until it is completed, or until the image has been successfully retrieved from the AlternativeLocationURL.

If the MOT Object has a zero body size and the AlternativeLocationURL returns with HTTP Status Code 404, then the slide shall be discarded.

Annex A (normative): APNG 1.0 Specification - Animated Portable Network Graphics

A.1 Introduction

A.1.1 Terminology

APNG is an extension of the PNG [8] format, adding support for animated images.

APNG is backwards-compatible with PNG; any PNG decoder should be able to ignore the APNG-specific chunks and display a single image.

The "default image" is the image described by the standard 'IDAT' chunks, and is the image that is displayed by decoders that do not support APNG.

The "canvas" is the area on the output device on which the frames are to be displayed. The contents of the canvas are not necessarily available to the decoder. As per the PNG Specification, if a 'bKGD' chunk exists it may be used to fill the canvas if there is no preferable background.

The "output buffer" is a pixel array with dimensions specified by the width and height parameters of the PNG 'IHDR' chunk. Conceptually, each frame is constructed in the output buffer before being composited onto the canvas. The contents of the output buffer are available to the decoder. The corners of the output buffer are mapped to the corners of the canvas.

"Fully transparent black" means red, green, blue and alpha components are all set to zero.

For purposes of chunk descriptions, an "unsigned int" shall be a 32-bit unsigned integer in network byte order limited to the range 0 to $(2^{31})-1$; an "unsigned short" shall be a 16-bit unsigned integer in network byte order with the range 0 to $(2^{16})-1$; a "byte" shall be an 8-bit unsigned integer with the range 0 to $(2^8)-1$.

A.1.2 Error Handling

APNG is designed to allow incremental display of frames before the entire image has been read. This implies that some errors may not be detected until partway through the animation. It is strongly recommended that when any error is encountered decoders should discard all subsequent frames, stop the animation, and revert to displaying the default image. A decoder which detects an error before the animation has started should display the DEFAULT image. An error message may be displayed to the user if appropriate.

A.2 Structure

A.2.1 General

An APNG stream is a normal PNG stream as defined in the PNG Specification [8], with three additional chunk types describing the animation and providing additional frame data.

To be recognized as an APNG, an 'acTL' chunk shall appear in the stream before any 'IDAT' chunks. The 'acTL' structure is described below.

Conceptually, at the beginning of each play the output buffer shall be completely initialized to a fully transparent black rectangle, with width and height dimensions from the 'IHDR' chunk.

The default image may be included as the first frame of the animation by the presence of a single 'fcTL' chunk before 'IDAT'. Otherwise, the default image is not part of the animation.

Subsequent frames are encoded in 'fdAT' chunks, which have the same structure as 'IDAT' chunks, except preceded by a sequence number. Information for each frame about placement and rendering is stored in 'fcTL' chunks. The full layout of 'fdAT' and 'fcTL' chunks is described below.

The boundaries of the entire animation are specified by the width and height parameters of the PNG 'IHDR' chunk, regardless of whether the default image is part of the animation. The default image should be appropriately padded with fully transparent pixels if extra space will be needed for later frames.

Each frame is identical for each play, therefore it is safe for applications to cache the frames.

A.2.2 Chunk sequence numbers

The 'fcTL' and 'fdAT' chunks have a 4 byte sequence number. Both chunk types share the sequence. The purpose of this number is to detect (and optionally correct) sequence errors in an Animated PNG, since the PNG specification does not impose ordering restrictions on ancillary chunks.

The first 'fcTL' chunk shall contain sequence number 0, and the sequence numbers in the remaining 'fcTL' and 'fdAT' chunks shall be in order, with no gaps or duplicates.

The tables below illustrate the use of sequence numbers for images with more than one frame and more than one 'fdAT' chunk.

If the default image is the first frame:

Sequence number	Chunk
(none)	'acTL'
0	'fcTL' first frame
(none)	'IDAT' first frame/default image
1	'fcTL' second frame
2	first 'fdAT' for second frame
3	second 'fdAT' for second frame

If the default image is not part of the animation:

Sequence number	Chunk
(none)	'acTL'
(none)	'IDAT' default image
0	'fcTL' first frame
1	first 'fdAT' for first frame
2	second 'fdAT' for first frame

Decoders shall treat out-of-order APNG chunks as an error. APNG-aware PNG editors should restore them to correct order using the sequence numbers.

A.2.3 'acTL': The Animation Control Chunk

The 'acTL' chunk is an ancillary chunk as defined in the PNG Specification. It shall appear before the first 'IDAT' chunk within a valid PNG stream.

The 'acTL' chunk contains:

Byte	Name	Description	Notes
0	num_frames (unsigned int)	Number of frames	
4	num_plays (unsigned int)	Number of times to loop this APNG	0 indicates infinite looping

'num_frames' indicates the total number of frames in the animation. This shall equal the number of 'fcTL' chunks. 0 is not a valid value. 1 is a valid value for a single-frame APNG. If this value does not equal the actual number of frames it should be treated as an error.

'num_plays' indicates the number of times that this animation should play; if it is 0, the animation should play indefinitely. If non-zero, the animation should come to rest on the final frame at the end of the last play.

A.2.4 'fcTL': The Frame Control Chunk

The 'fcTL' chunk is an ancillary chunk as defined in the PNG Specification. It shall appear before the 'IDAT' or 'fdAT' chunks of the frame to which it applies, specifically:

For the default image, if a 'fcTL' chunk is present it shall appear before the first 'IDAT' chunk. Position relative to the 'acTL' chunk is not specified.

For the first frame excluding the default image (which may be either the first or second frame), the 'fcTL' chunk shall appear after all 'IDAT' chunks and before the 'fdAT' chunks for the frame.

For all subsequent frames, the 'fcTL' chunk for frame N shall appear after the 'fdAT' chunks from frame N-1 and before the 'fdAT' chunks for frame N.

Other ancillary chunks are allowed to appear among the APNG chunks, including between 'fdAT' chunks.

Exactly one 'fcTL' chunk is required for each frame.

Format:

Byte	Name	Description	Notes
0	sequence_number (unsigned int)	Sequence number of the animation chunk, starting from 0	
4	width (unsigned int)	Width of the following frame	
8	height (unsigned int)	Height of the following frame	
12	x_offset (unsigned int)	X position at which to render the following frame	
16	y_offset (unsigned int)	Y position at which to render the following frame	
20	delay_num (unsigned int)	Frame delay fraction numerator	
22	delay_den (unsigned short)	Frame delay fraction denominator	
24	dispose_op (unsigned short)	Type of frame area disposal to be done after rendering this frame	
25	blend_op (byte)	Type of frame area rendering for this frame	

The frame shall be rendered within the region defined by 'x_offset', 'y_offset', 'width', and 'height'. The offsets shall be non-negative, the dimensions shall be positive, and the region may not fall outside of the default image.

Constraints on frame regions:

$$'x_offset' \geq 0$$

$$'y_offset' \geq 0$$

$$'width' > 0$$

$$'height' > 0$$

$$'x_offset' + 'width' \leq 'IHDR' \text{ width}$$

$$'y_offset' + 'height' \leq 'IHDR' \text{ height}$$

The 'delay_num' and 'delay_den' parameters together specify a fraction indicating the time to display the current frame, in seconds. If the denominator is 0, it is to be treated as if it were 100 (that is, 'delay_num' then specifies one-hundredths of a second). If the value of the numerator is 0 the decoder should render the next frame as quickly as possible, though viewers may impose a reasonable lower bound.

Frame timings should be independent of the time required for decoding and display of each frame, so that animations will run at the same rate regardless of the performance of the decoder implementation.

'dispose_op' specifies how the output buffer should be changed at the end of the delay (before rendering the next frame).

Valid values for 'dispose_op' are:

- 0 APNG_DISPOSE_OP_NONE
- 1 APNG_DISPOSE_OP_BACKGROUND
- 2 APNG_DISPOSE_OP_PREVIOUS

APNG_DISPOSE_OP_NONE: no disposal is done on this frame before rendering the next; the contents of the output buffer are left as is.

APNG_DISPOSE_OP_BACKGROUND: the frame's region of the output buffer is to be cleared to fully transparent black before rendering the next frame.

APNG_DISPOSE_OP_PREVIOUS: the frame's region of the output buffer is to be reverted to the previous contents before rendering the next frame.

If the first 'fcTL' chunk uses a 'dispose_op' of APNG_DISPOSE_OP_PREVIOUS it should be treated as APNG_DISPOSE_OP_BACKGROUND.

'blend_op' specifies whether the frame is to be alpha blended into the current output buffer content, or whether it should completely replace its region in the output buffer.

Valid values for 'blend_op' are:

- 0 APNG_BLEND_OP_SOURCE
- 1 APNG_BLEND_OP_OVER

If 'blend_op' is APNG_BLEND_OP_SOURCE all colour components of the frame, including alpha, overwrite the current contents of the frame's output buffer region. If 'blend_op' is APNG_BLEND_OP_OVER the frame should be composited onto the output buffer based on its alpha, using a simple OVER operation as described in the "Alpha Channel Processing" section of the PNG specification [20]. Note that the second variation of the sample code is applicable.

Note that for the first frame the two blend modes are functionally equivalent due to the clearing of the output buffer at the beginning of each play.

The 'fcTL' chunk corresponding to the default image, if it exists, has these restrictions:

- The 'x_offset' and 'y_offset' fields shall be 0.
- The 'width' and 'height' fields shall equal the corresponding fields from the 'IHDR' chunk.

As noted earlier, the output buffer shall be completely initialized to fully transparent black at the beginning of each play. This is to ensure that each play of the animation will be identical. Decoders are free to avoid an explicit clear step as long as the result is guaranteed to be identical. For example, if the default image is included in the animation, and uses a 'blend_op' of APNG_BLEND_OP_SOURCE, clearing is not necessary because the entire output buffer will be overwritten.

A.2.5 'fdAT': The Frame Data Chunk

The 'fdAT' chunk has the same purpose as an 'IDAT' chunk. It has the same structure as an 'IDAT' chunk, except preceded by a sequence number.

At least one 'fdAT' chunk is required for each frame. The compressed datastream is then the concatenation of the contents of the data fields of all the 'fdAT' chunks within a frame. When decompressed, the datastream is the complete pixel data of a PNG image, including the filter byte at the beginning of each scanline, similar to the uncompressed data of all the 'IDAT' chunks. It utilizes the same bit depth, colour type, compression method, filter method, interlace method, and palette (if any) as the default image.

Format:

Byte	Name	Description	Notes
0	sequence_number (unsigned int)	Sequence number of the animation chunk, starting from 0	
4	frame_data (X bytes)	Frame data for this frame	

Each frame inherits every property specified by any critical or ancillary chunks before the first 'IDAT' in the file, except the width and height, which come from the 'fcTL' chunk.

If the PNG 'pHYs' chunk is present, the APNG images and their 'x_offset' and 'y_offset' values shall be scaled in the same way as the main image. Conceptually, such scaling occurs while mapping the output buffer onto the canvas.

A.3 Test encoder and sample images

Sample images are available from the APNG implementation page at <http://littlesvr.ca/apng/>.

Open source encoders and applications to assemble APNGs are available; for example see <http://littlesvr.ca/apng/apngedit.html>.

Annex B (informative): Implementing SlideShow with time shifted audio services

B.1 Introduction

This annex gives specific guidance to manufacturers implementing SlideShow on receiver devices that include functionality to Pause or Record audio services from broadcast. The recording of audio services may be through direct user intervention, or through the use of background recording enabled through an Electronic Programme Guide or other timer functionality.

It is mandatory that such receivers adjust their implementation of SlideShow to account for the difference between the time of original broadcast and the time of replay, which will recreate the original timing relationship between audio and SlideShow images.

If the receiver is unable to implement adjusted timings, then the SlideShow application should be stopped whenever audio is time shifted - otherwise it would give an inconsistent and unsatisfactory experience for both users and service providers.

B.2 SlideShow in X-PAD

Figure B.1 illustrates a conceptual system for handling time shifted audio when the SlideShow component (along with DL and other applications) is transported in X-PAD.

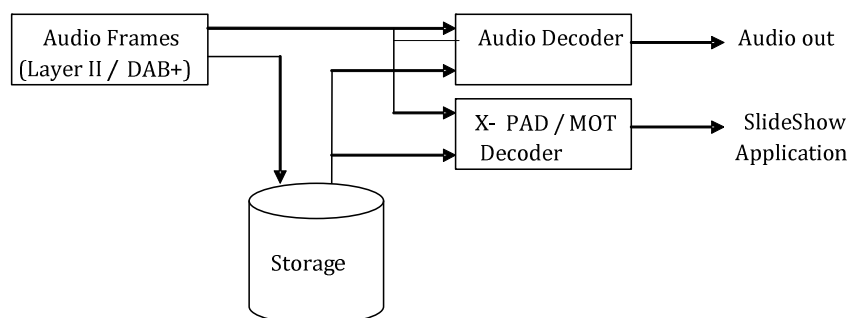


Figure B.1: Conceptual System for Time shifting SlideShow in X-PAD

The receiver should record the entire audio stream, including the X-PAD content. At playback, the entire audio stream is passed to both the audio decoder and the X-PAD decoder, which will recreate the original transmission of both audio and data.

The receiver should also record two additional pieces of information for each recording:

- The original FIG 0/13 signalling at the time of recording, in order to determine if a SlideShow is present and all the parameters needed to decode the SlideShow application (e.g. the used X-PAD Application Types).
- The date and time with a resolution and accuracy of 1 s at the moment the recording commences (as derived from FIG0/10).

B.3 SlideShow in Packet Mode or X-PAD

An alternative approach could be equally applicable to SlideShow when transmitted in either Packet Mode or X-PAD, but is potentially more complex.

Whilst recording, the audio SlideShow images could be decoded into individual image files (along with the relevant ContentName and TriggerTime parameters), which would be stored with the recorded audio.

It would also be necessary to record date and time (provided by FIG0/10) with a resolution and accuracy of 1 s at the commencement of the recording.

B.4 Reference time and TriggerTime

For a time shifted broadcast, the SlideShow Reference Time needs to be adjusted accordingly to reflect the original transmission time of the broadcast.

Specifically:

- The SlideShow Reference Time takes into account any position change in the playback stream, such as rewinding, pausing, fast-forwarding or skipping.
- The comparison of TriggerTime (for values other than "Now") is made against the adjusted SlideShow Reference Time.

Annex C (informative): Use Cases

C.1 AlternativeLocationURL

C.1.1 General

The bandwidth of an IP channel is assumed to be greater than that available via broadcast, and although not unconstrained, it is typically orders of magnitude greater. This may then be used by Service provider and Device in a number of novel ways.

A device which is not IP-connected will ignore the AlternativeLocationURL and continue to use the slide image as received over broadcast.

C.1.2 Faster image acquisition

The acquisition time of a slide over broadcast is a consequence of several factors, with the size of the file usually being the most significant.

For example, consider a SlideShow service with a bandwidth of 12 kbps. A rough estimate for the acquisition time for a slide of 15 kB (ignoring any protocol overheads) would be 10 seconds. Given an average mobile IP-connection bandwidth of 1,5 Mbps, the acquisition time would be 0,08 seconds.

Thus, the AlternativeLocationURL can be used to acquire the image much faster than over broadcast. It may also be used to serve a higher quality, less compressed version of the image, which would be of a larger file size.

C.1.3 Increased image relevance

If a slide is broadcast with an AlternativeLocationURL, IP-connected devices may send additional information as part of the HTTP request used to acquire the slide. Some of these are part of the core HTTP specification, such as the *User Agent*.

This identifies the class and characteristics of the requesting device, and is used in *Content Negotiation*, with the response being tailored to the most suitable content for the requesting device:

- Image of larger dimensions, to fit higher dimension displays, by using Display Dimension information in the HTTP request to acquire the image.
- Content tailored to the specific device model or range of devices, by using User Agent information in the HTTP request to acquire the image.
- Content tailored to the location of the user, by using a Geo-IP lookup for the device IP address.

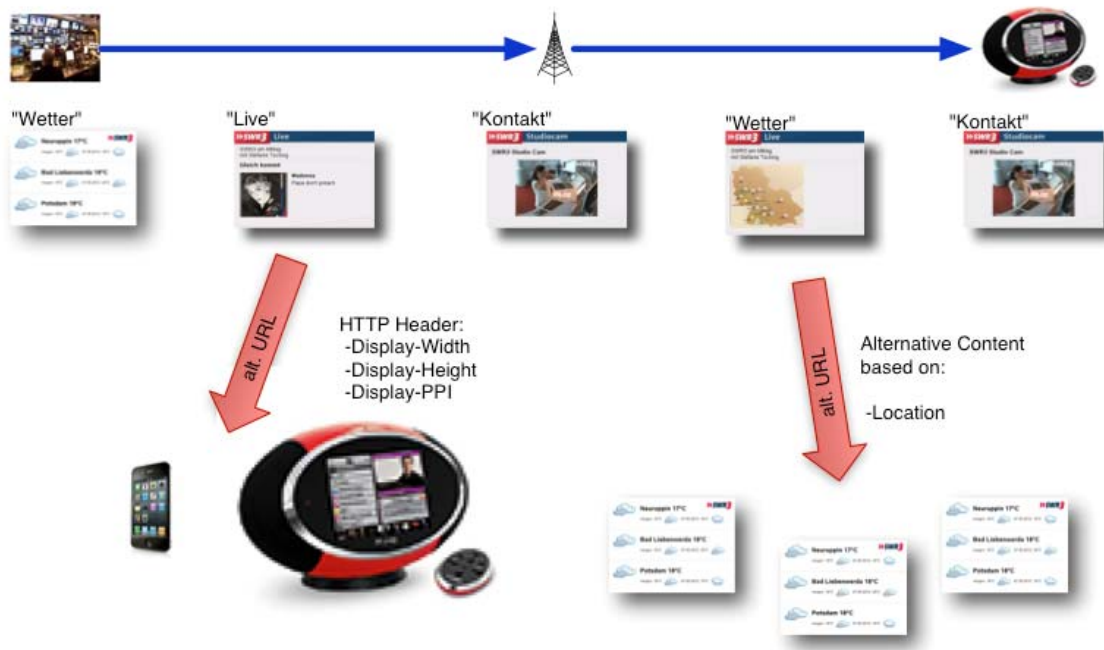


Figure C.1: Example AlternativeLocationURL use

C.1.4 Faster carousel rotation

Due to the increased bandwidth available to IP-connected devices, a service provider may use the AlternativeLocationURL to give the impression of faster carousel rotation.

For example, assume an MOT carousel rotation of one slide every 20 seconds. Within this time, the service provider may send out MOT Objects with an AlternativeLocationURL and no body content. For IP-connected devices supporting this additional functionality, this would cause a new image to be acquired and displayed. Devices not supporting this additional functionality would ignore this, and continue to display the last acquired image.

C.2 ClickThroughURL



Figure C.2: Example ClickThroughURL use

C.3 Categorized SlideShow

Categorized SlideShow allows the service provider to add categorization information to the slides in order to provide a thematical grouping for the slide contents. Usually these categories will be topics like "News", "Sport", "Traffic", or "Weather". A legacy device, which does not implement the additional MOT parameters used for categorization will not be affected and will present the slides in the same linear sequence as ever. However a receiver which implements the categorization feature will allow the end user to switch into an interactive presentation mode, which provides the functionality to access the previously received and categorized slides in a non-linear and interactive way.

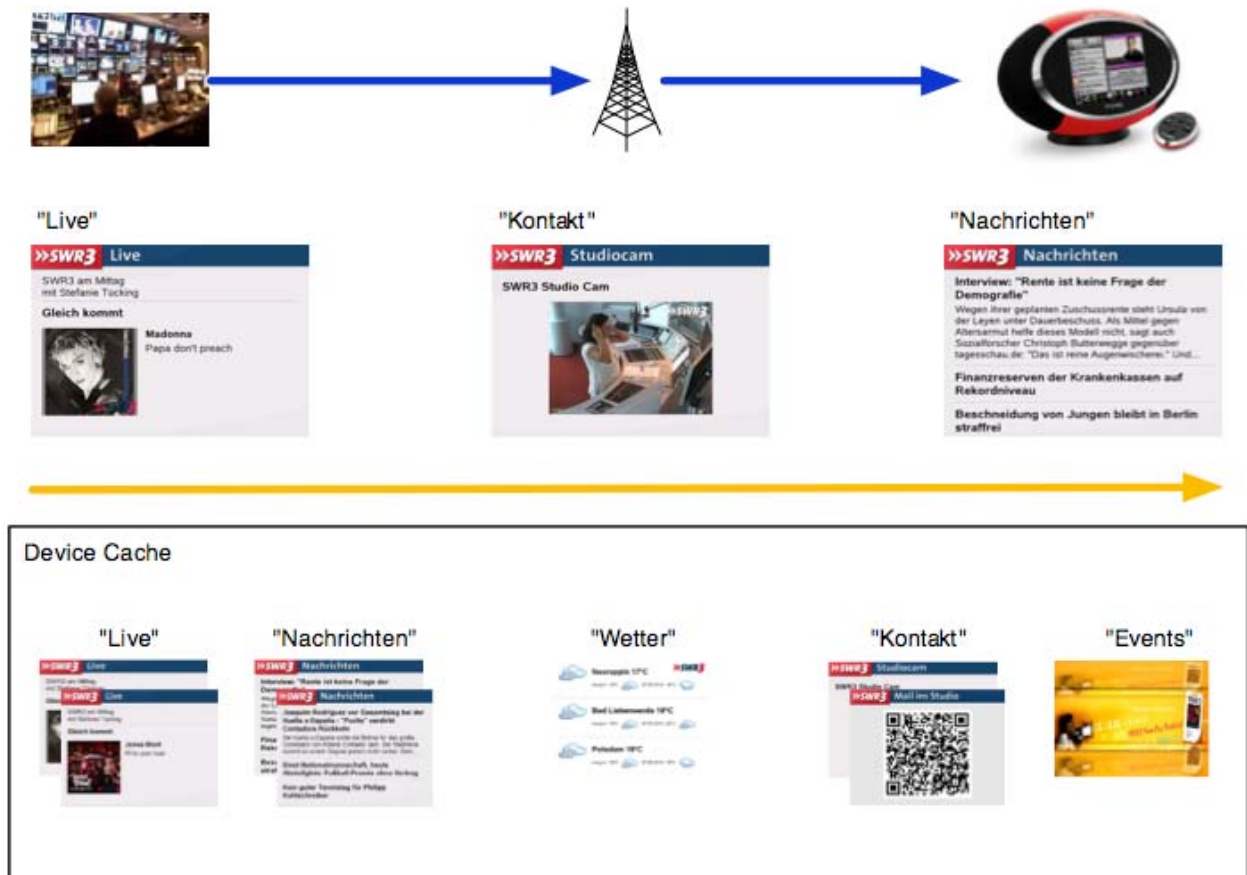


Figure C.3: Example of linear and interactive presentation modes

Figure C.3 shows a snapshot from a listening session on a device which supports the categorized SlideShow. During the last few minutes, the service provider has delivered a slide with information about the upcoming music title (Madonna - "Papa don't preach"), a slide with a web cam image from the studio and a slide with actual news (depicted in the middle row). Additionally with the image content, categorization information was sent and the device builds up a slide cache (holding buffer). If the end user switches from the linear presentation mode to the interactive mode, he/she will be able to browse and view the slide cache based on his/her preferred category.

The cache structure and the presented slide content is always under the control of the service provider.

The following use cases are possible:

Linear Slide only: If no categorization parameters are given (CategoryID/SlideID, CategoryTitle), the slide will be presented in the linear mode only.

Categorized Slide only: If no trigger time is set, but the categorization parameters are given, the slide is presented in the interactive mode only. However if a later TriggerTime update for this particular slide is received, the device will also present this slide out of the slide cache in the linear mode.

Remove Slide from interactive mode: A previously categorized slide can be removed from the presentation in the interactive mode, simply by sending a MOT update for the CategoryID/SlideID parameter with value 0x0000.

Replace a slide in the categorized view: If an existing slide in the categorized view should be replaced by another/new slide. Simply the new slide is sent over the MOT with the same categorization parameters as the slide to replace.

Annex D (normative): Methods for synchronizing SlideShow Reference Time

D.1 General

In order to act upon TriggerTime and ExpireTime, the receiver shall maintain an internal time reference, the SlideShow Reference Time, to a resolution of 1 second. It is recommended that this be continuously maintained through use of a battery backup or similar.

It shall not be possible for the user to amend the value of the SlideShow Reference Time. It shall only be set through synchronization as described here.

D.2 Delivery of SlideShow over DAB

It is mandatory to synchronize the SlideShow Reference Time to UTC using the values in FIG0/10.

If FIG0/10 is being transmitted in the 'short' form, then care should be taken to synchronize the SlideShow Reference Time at the transition from one minute to the next.

D.3 Delivery of SlideShow over DRM

It is mandatory to synchronize the SlideShow Reference Time to UTC using the values in SDC Data Entity 8.

Care should be taken to synchronize the SlideShow Reference Time at the transition from one minute to the next.

D.4 Delivery of SlideShow over IP

It is mandatory to synchronize the SlideShow Reference Time to UTC using Network Time Protocol (NTP) [i.3].

Annex E (normative): List of metadata properties

E.1 List of Server-sent Events properties

The category keys for Server-sent Events are defined in table E.1. They are derived from DL Plus content types, see ETSI TS 102 980 [19], annex A. Only the category keys defined in table E.1 shall be used: other DL Plus content types exist but do not have a direct equivalence, for example, because the functionality is better provided by use of Service and Programme Information, ETSI TS 102 818 [7]. Equivalent ID3v2 tags are shown where they relate directly to the same information.

The **key** value is in simple JSON dot notation format. For example, `item.title` implies that the response contains a JSON object with the property `item`, which has the value of another object with the property `title`, which has the value of the title of current item on air.

```
EXAMPLE:  {
           "item": {
             "title": "The Item Title"
           }
         }
```

Table E.1: List of keys

Key	Corresponding DL Plus content type	ID3v2 tag	Description
<code>item.title</code>	ITEM.TITLE	TIT2	Title of the current item on air; e.g. title of track that belongs to an album
<code>item.album</code>	ITEM.ALBUM	TALB	The collection name to which this item belongs
<code>item.trackNumber</code>	ITEM.TRACKNUMBER	TRCK	The track number of the item on the album on which it was originally released
<code>item.artist</code>	ITEM.ARTIST	TPE1	A person or band/collective generally considered responsible for the work
<code>item.composition</code>	ITEM.COMPOSITION	TIT1	A complete composition (classical music broadcasters should use this item to identify the composition)
<code>item.movement</code>	ITEM.MOVEMENT	TIT3	A movement is a large division of a composition or musical form (classical music broadcasters should use this item to identify the movement)
<code>item.conductor</code>	ITEM.CONDUCTOR	TPE3	The artist(s) who performed the work. In classical music this would be the conductor
<code>item.composer</code>	ITEM.COMPOSER	TCOM	Name of the original composer/author
<code>item.band</code>	ITEM.BAND	TPE2	Band/orchestra/accompaniment/musician
<code>item.comment</code>	ITEM.COMMENT	COMM	Any comment related to the content
<code>item.genre</code>	ITEM.GENRE	TCON	The main genre of the audio, e.g. "classical", "hip-hop", "jazz", "oldies", "drama", etc.
<code>item.artwork</code>	-		URL of artwork for the audio.
<code>info.news.headline</code>	INFO.NEWS		News headline (see note 1)
<code>info.news.local</code>	INFO.NEWS.LOCAL		Local news (see note 1)
<code>info.stockmarket</code>	INFO.STOCKMARKET		Quote information; may be built from several distinct parts, e.g. "name __latest value __change __high __low __volume" (see note 1)
<code>info.sport</code>	INFO.SPORT		Result of a game; may be built from several distinct parts, e.g. "match __result" (see note 1)
<code>info.lottery</code>	INFO.LOTTERY		Raffle/lottery (see note 1)
<code>info.horoscope</code>	INFO.HOROSCOPE		Horoscope; key word for a table, e.g. "sign of the zodiac" (see note 1)
<code>info.dailyDiversion</code>	INFO.DAILY_DIVERSION		Daily tip/diversion/joke/etc. (see note 1)
<code>info.health</code>	INFO.HEALTH		Information about health (see note 1)
<code>info.event</code>	INFO.EVENT		Info about an event (see note 1)

Key	Corresponding DL Plus content type	ID3v2 tag	Description
info.scene	INFO.SCENE		Information for the target audience of the programme service; for a youth programme info for the community and about its scene (hot locations to be) (see note 1)
info.cinema	INFO.CINEMA		Information about movies in cinema (see note 1)
info.tv	INFO.TV		Information about TV-movies (see note 1)
info.weather	INFO.WEATHER		Information about weather (see note 1)
info.traffic	INFO.TRAFFIC		Information about traffic. This shall alert users in case of exceptional traffic news (see note 1)
info.alarm	INFO.ALARM		Alarm information and other urgent information; if implemented by a receiver, it should be presented immediately to the user. The receiver configuration should allow de-selection of this feature
info.advertisement	INFO.ADVERTISEMENT		Advertisement (e.g. book review, commercial, etc.), may be additional information to an advertisement currently sent out in the audio channel of that programme (see note 1)
info.url	INFO.URL		URL link to interesting information sources; <Keyword>_<URL> (see note 1)
info.other	INFO.OTHER		Other information, topic defined by service provider; different topics may be specified by keywords; <Keyword>_<information> (see note 2)
programme.name	PROGRAMME.NOW		Name of the current programme (useful if SPI service is not available)
programme.description	PROGRAMME.NOW		Description of the current programme (useful if SPI service is not available)
programme.artwork	-		URL of the programme artwork
chat.content	CHAT		Chat content is delivered by the radio station, may be a chat contribution from a listener
chat.center	CHAT.CENTER		Address (may be URL or SMS), where a listener should send his replies as chat contribution to
vote.question	VOTE.QUESTION		A question sent concerning a vote; could be binary, i.e. with two valid answers ("yes" or "no" or "1" or "2"), could also be a multiple choice question; could also be used with riddles, 10€-Questions, etc.
vote.centre	VOTE.CENTRE		URL or SMS number to send the answer to: the range of values for an expected answer will be indicated by VOTE.QUESTIONS (preferred characters are decimal numbers, alphabet characters not excluded)
descriptor.place	DESCRIPTOR.PLACE		Adds info about a location, address, etc.; important use case: in conjunction with INFO.EVENT (see note 2)
descriptor.appointment	DESCRIPTOR.APPOINTMENT		Adds in general info about date and time for something in future; important use case: appointment in conjunction with INFO.EVENT (see note 2)
descriptor.identifier	DESCRIPTOR.IDENTIFIER	TSRC	For music it is the International Standard Recording Code (http://www.ifpi.org/isrc/) (see note 2)
descriptor.purchase	DESCRIPTOR.PURCHASE	WPAY	Address (URL or SMS number) where item can be ordered or purchased; includes items free of charge (see note 2)
descriptor.data	DESCRIPTOR.GET_DATA		SMS number or URL as source to retrieve more data about that is linked to the corresponding DL Plus object (see note 2)
NOTE 1: Multiple Server-sent Events category keys can be combined in a single payload to provide more structure to a message.			
NOTE 2: A Descriptor will always provide more information about another category key that is included in the same payload.			

History

Document history		
V1.1.1	July 2001	Publication
V2.1.1	January 2006	Publication
V2.2.1	July 2008	Publication
V2.3.1	May 2013	Publication
V3.1.1	January 2015	Publication
V3.2.1	July 2023	Publication