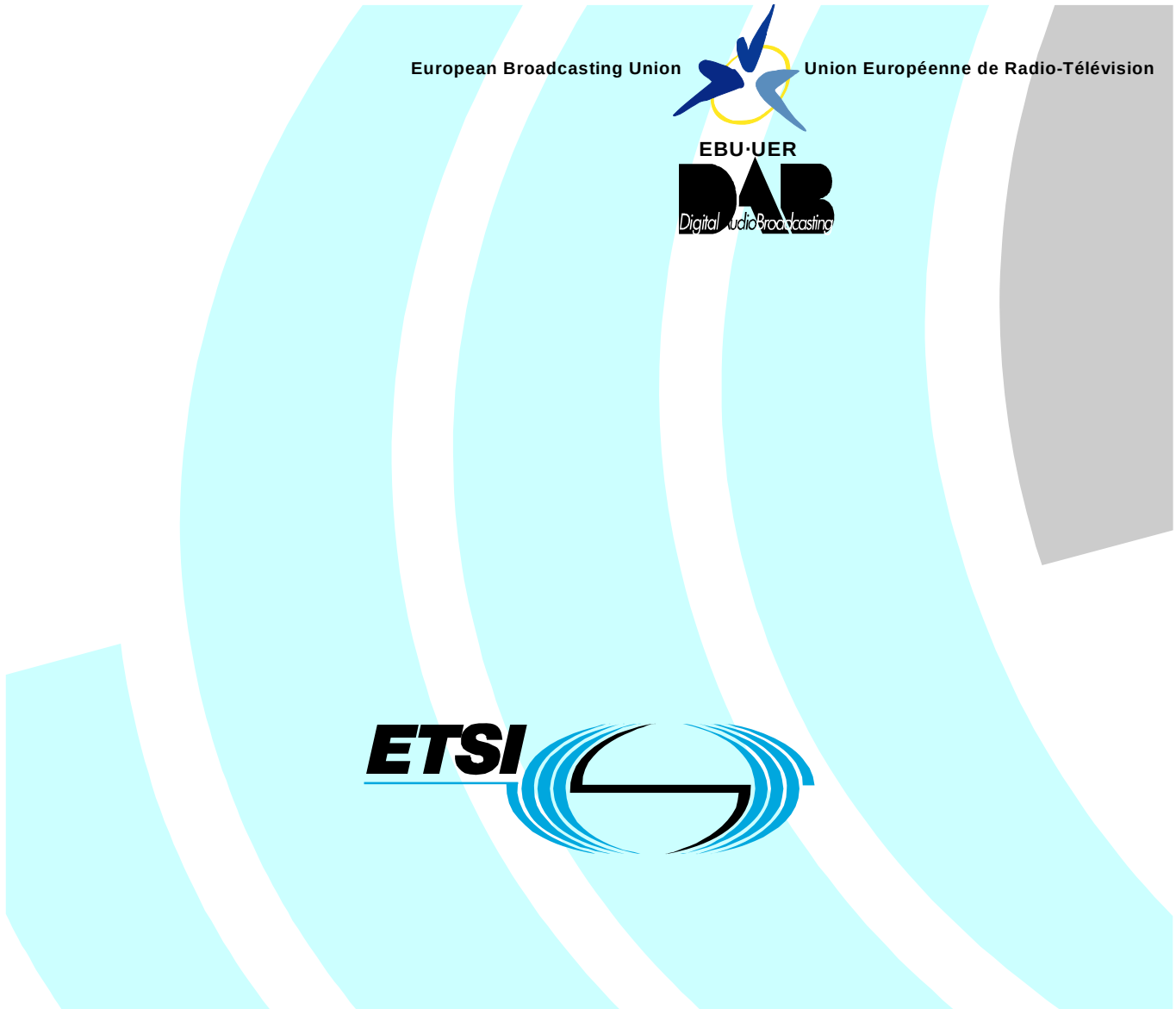
ETSI TS 101 499 V2.2.1 (2008-07)

Technical Specification

Digital Audio Broadcasting (DAB); MOT SlideShow; User Application Specification



Reference
RTS/JTC-DAB-57

Keywords
audio, broadcasting, DAB, digital, PAD

ETSI

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

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

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

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

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

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Contents

Intellectual Property Rights

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

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

Foreword

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

NOTE 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, EN 300 401 [1], for DAB (see note 2) which now has worldwide acceptance. The members of the Eureka Project 147 are drawn from broadcasting organizations and telecommunication providers together with companies from the professional and consumer electronics industry.

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

1 Scope

The present document describes the protocol required to create the DAB user application "SlideShow".

The "SlideShow" user application applies the DAB-MOT protocol (EN 301 234 [3]) and allows a service provider to deliver a sequence of slides which carry information in the form of images.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

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

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

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

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[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] ISO/IEC IS 15948: "Information technology - Computer graphics and image processing - Portable Network Graphics (PNG): Functional specification".

2.2 Informative references

The following referenced documents are not essential to the use of the present document but they assist the user with regard to a particular subject area**.** For non-specific references, the latest version of the referenced document (including any amendments) applies.

Not applicable.

3 Abbreviations

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

APNG	Animated Portable Network Graphics
CD	Compact Disc
DAB	Digital Audio Broadcasting
DLS	Dynamic Label Segment
FIDC	Fast Information Data Channel
JPEG	Joint Pictures Expert Group
MJD	Modified Julian Date
MOT	Multimedia Object Transfer
MSC	Main Service Channel
PAD	Programme Associated Data
PNG	Portable Network Graphics
UTC	Universal Time Coordinated
X-PAD	eXtended Programme Associated Data

4 Introduction

SlideShow provides the user with a sequence of slides which carry information in the form of images. The slides will be presented on an appropriate display.

The main use for this user application is to visualize an audio programme. Examples are:

- news programme items complemented by photos from the reported events;
- programme items with popular songs accompanied by photos of the favourite groups or the covers of the CD the songs are taken from;
- images for advertising, brand reinforcement or promotional purposes.

The user application may also be provided as a stand-alone data service.

SlideShow allows JPEG and PNG images to be transmitted, which may be formed by compositing graphics, photographs and text.

Once activated, the SlideShow is a service provider-driven user application, which does not require any interaction from the user. Each slide appears automatically on the display and is replaced under the control of the service provider according to the needs of his service.

The SlideShow can be transmitted in the following ways:

- in the PAD of an audio service component of a programme or data service;
- as a secondary service component of a programme or data service (in this case the SlideShow will be a packet mode service component);
- as the primary service component of a data service (in this case the SlideShow will be a packet mode service component).

SlideShow receivers shall be able to decode the application data from both PAD and packet mode service components.

The SlideShow user application is designed to allow different levels of sophistication for both the broadcast content and the receiver complexity. As such, two profiles are defined:

- **simple**; and
- **enhanced**.

The profile in use is not signalled to receivers explicitly; rather the user application is designed such that the **simple** receiver is not confused by the reception of an **enhanced** Slideshow, but a **simple** receiver will generally give a degraded user experience compared to an **enhanced** receiver.

The **simple** profile was specified in version 2.1.1 of the present document. It defines a simple receiver behaviour that receives, decodes, renders and displays a single image at a time. This profile is only suitable for receivers with very restricted resources.

The **enhanced** profile extends the **simple** profile by defining control mechanisms to greatly enhance the presentation to the user. This includes the ability to present the slides independently of the transmission order and to transmit a series of slides in advance of the display time, thus allowing the images to play faster than would be permitted by a **simple** profile receiver. Animation of PNG images is also enabled.

5 Operation of the SlideShow user application

The SlideShow user application simply conveys one slide at a time from the service provider to the terminal. After complete and error-free reception it is presented on the display under the control of the service provider and replaces the previous slide.

5.1 Transport

The SlideShow user application uses the MOT protocol: Each slide, together with its parameters, is taken as one MOT object: it is segmented and its segments are transferred according to the rules of the MOT protocol. MOT headers and bodies are used. MOT directories are not used.

A SlideShow may be transferred either in the PAD part of an MSC stream audio sub-channel or in a MSC packet mode data sub-channel.

Wireless broadcast channels like DAB 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 DAB 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).

In order to efficiently manage memory, the receiver may assume that receipt of a new MOT body infers that all prior MOT bodies have been completed, and may clear the Assembly Buffer of incompletely received MOT bodies and Header segments. The slides will experience different delays (according to different reception conditions in different places), until the complete object is assumed to be available in an error-free state at the majority of terminals within the intended coverage area. If the service provider wants to ensure a precise start of the slide presentation, he has to start the transmission of the MOT Body sufficiently in advance, either with the TriggerTime parameter set to a time in the future, or without a TriggerTime parameter and subsequently transmit a Header Update to amend the TriggerTime to a new time or to "Now".

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.

5.2 Storage and memory management

The user application requires the receiving terminal conceptually to control a number of buffers.

NOTE: Actual implementations may achieve identical results using fewer buffers than those shown here.

In the conceptual model, one buffer is used to re-assemble incoming MOT object Segments into a complete MOT object. Complete MOT objects are passed to the Holding buffer.

A **simple** profile receiver will immediately render the image into a rendering buffer. When the image is triggered, the already rendered image will be copied to the screen buffer,

An **enhanced** profile receiver is fast enough to render an image in real-time. Therefore it will await a valid TriggerTime before starting the rendering process. When triggered, images are rendered and immediately displayed.

The buffer sizes shall be determined by the maximum MOT object size and the rendered image size, as given in clause 8.2.

All buffers shall be cleared when the application is terminated, either by a change in the signalling of FIG0/13, or by the user ending the SlideShow application.

If a SlideShow is a packet mode service component of the currently selected service and the user selects another service with the same SlideShow as a service component, then the buffers shall be preserved and the SlideShow shall be continued. A SlideShow in PAD shall be continued if the user switches to another service using the same audio service component. In all other circumstances, all buffers shall be cleared.

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.

5.2.1 Simple profile

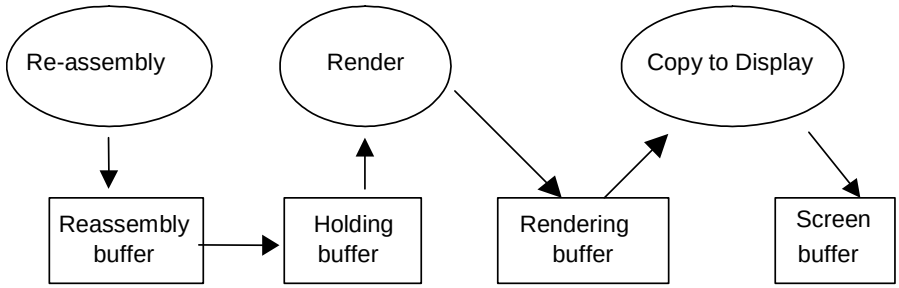


Figure 5.1: Simple profile Buffer management model for a MOT SlideShow decoder

The successful reception of an MOT object 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.

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

5.2.2 Enhanced profile

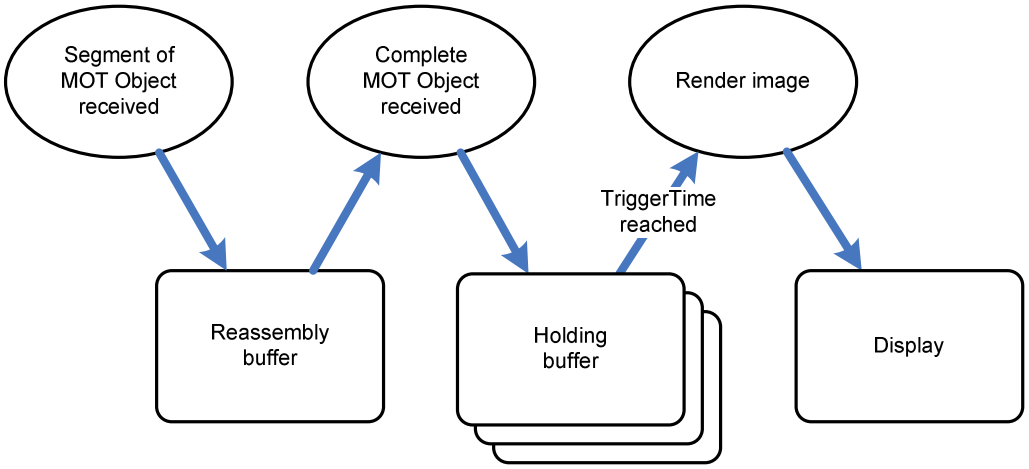


Figure 5.2: Enhanced profile Buffer management model for a MOT SlideShow decoder

The successful reception of a complete MOT 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 ContentName and TriggerTime parameters.

At TriggerTime the image is decompressed from the Holding Buffer to a bitmap format and then immediately copied to the Display.

If the Holding Buffer has insufficient space to store a newly reassembled image, the receiver should delete images one by one until sufficient space has been freed. Images with no TriggerTime should be removed first. If all images have a TriggerTime, the image with earliest historic TriggerTime should be deleted first. Images with a TriggerTime in the future should not be deleted.

5.3 Presentation

The SlideShow user application works with one display only: it can display only one slide at a time. Each image replaces the previous one, and remains on display until it is in turn replaced. There is no explicit way of removing a slide from the display. In the event that the SlideShow application is terminated, either by the service provider or by user action, the screen should return to a relevant user display, for example a full-screen Dynamic Label display.

5.4 Timing

The presentation time of each slide is solely controlled by the service provider.

The MOT Parameter "TriggerTime" (clause 6.2.2) defines the time at which the display shall be updated by rendering the image from the Holding Buffer to the Display (**enhanced** profile) or by copying the image from the rendering buffer to the Display (**simple** profile).

There are five cases for the value TriggerTime:

- TriggerTime greater than current UTC time and MJD date.
  The image is intended for display at the specified point in the future. The receiver shall hold the image in the Holding Buffer (or Rendering Buffer) until this time is reached.

- TriggerTime equal to the current UTC time and MJD date.
  The image shall be shown on the receiver's display. If the image is directly rendered from the Holding Buffer into the Display, then the rendering delay shall be taken into account when determining the start of the rendering process.

- TriggerTime less than the current UTC time and MJD date.
  If the reception of an MOT object completes later than the value of TriggerTime, the image shall be held in the Holding Buffer, but shall not be displayed until a new TriggerTime is received for this object, or the buffer space is required for a later image.

- TriggerTime "Now".
  The image shall be shown on the Display with minimal delay. "Now" is a specially signalled value (see clause 6.2.2) and only applies at the instant of reception - images in the Holding Buffer with a TriggerTime of "Now" should be displayed only once, unless a new TriggerTime of "Now" is subsequently received through the reception of a Header Update (see clause 5.5).

- No TriggerTime.
  The image shall be held in the Holding Buffer, but shall not be displayed until a TriggerTime is received for this object, or the buffer space is required for a later image.

If any service within an ensemble contains a SlideShow using TriggerTime with values other than "Now", it is mandatory to transmit FIG0/10. It is strongly recommended to use the long form of FIG0/10 (i.e. including seconds and milliseconds) transmitted every 10 seconds (repetition rate "C") or faster. If this is not possible and the short form of FIG0/10 is used, it should be transmitted every second (repetition rate "B") or faster, and the rate should be increased during the 5 seconds before and after the transition of the minute's edge.

In order to act upon TriggerTime, the receiver shall maintain an internal clock to a resolution of 1 second. It is recommended that this internal clock be continuously maintained through use of a battery backup or similar. The internal clock shall be synchronized to an accuracy of 1 second using FIG0/10 where it is transmitted. In the case of "short" form FIG0/10, the internal clock shall be set at the transition of a minute (i.e. when the value of "seconds" is 0), since seconds are not transmitted in the "short" form. In the event that the internal clock has never been synchronized to FIG0/10 transmitted from any ensemble, only objects with TriggerTime "Now" may be displayed. It shall not be possible for the user to amend the value of the internal clock; it shall only be set through synchronisation to FIG0/10.

Receiver manufacturers implementing functionality that allows audio to be delayed paused or time-shifted should refer to annex B for advisory information on handling SlideShow in these circumstances.

5.5 Header update

An MOT object may be transmitted without the TriggerTime parameter, and shall be held in the Holding Buffer. The TriggerTime may be subsequently transmitted as a Header update, using the ContentName to uniquely identify the target MOT object to update. The receiver shall then process the image, based on its actual TriggerTime, according to the cases above.

6 Interface to the Transport layer MOT

The SlideShow user application is implemented in the DAB system by transferring the slides including all necessary control information, as objects according to the Multimedia Object Transfer protocol used with the two Transport Mechanisms:

- "MSC stream audio" (PAD part).

- "MSC packet mode data".

The Transport Mechanisms "MSC stream data" and "FIDC" shall not be used for the SlideShow application.

NOTE: Forward Error Correcting schemes such as "FEC for MSC packet mode" (see EN 300 401 [1], clause 5.3.5), may be used.

MOT Headers and Bodies are used. MOT Directory shall not be used.

6.1 MOT ContentTypes and ContentSubTypes

According to the MOT protocol, each object has to be characterized by its ContentType and ContentSubType (see EN 301 234 [3] and TS 101 756 [2]). The user application requires this information in order to address the corresponding content decoders correctly. The following types are the only ones permitted for the use in the SlideShow user application.

6.1.1 ContentType "Image"

6.1.1.1 ContentSubType "JFIF" (JPEG)

For an `image/jpeg` content 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.

6.1.1.2 ContentSubType "PNG"

The ContentSubType PNG shall be used to indicate either a still or animated PNG image.

For an `image/png` content 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 IS 15948 [4]), 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.

6.1.2 ContentType "MOT transport"

6.1.2.1 ContentSubType "Header update"

In addition to the objects carrying the slides themselves, special objects with ContentType "MOT transport" and ContentSubType "Header update" can be used to transmit the TriggerTime for the presentation of an object previously broadcast.

6.2 MOT Parameters for the slide objects

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

Parameter	Parameter Id	Specified in	Mandatory for service provider	Mandatory for receiver	Occurrence
ContentName	0x0C	MOT EN 301 234 [3]	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)	Yes	Single

6.2.1 ContentName

According to the MOT protocol, the ContentName is needed for identifying and handling the object in the memory management. Therefore its use is mandatory within the SlideShow user application. The ContentName shall be changed with each new image.

The application provider may re-use a ContentName value after an appropriate interval. 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 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.

6.2.2 TriggerTime

This parameter specifies the time at which the presentation takes place. The TriggerTime activates the object according to its ContentType. The value of the parameter field is coded in the UTC plus MJD format (see EN 301 234 [3], clause 6.2.4.1).

If an object should be presented as soon as it is received, a TriggerTime of "Now" is used. This is indicated by setting the TriggerTime validity flag to 0.

The service provider controls the presentation of the object by setting this MOT parameter. If a SlideShow MOT Object is broadcast with no TriggerTime parameter, it can only be presented by the terminal if this slide object is subsequently followed by a "Header update" object with the TriggerTime for that object.

6.2.2.1 Simple profile

The TriggerTime may only be provided once. The receiver is only permitted to display the object once, then discard the object and ignore subsequent TriggerTime updates.

6.2.2.2 Enhanced profile

The TriggerTime may be provided multiple times through a series of Header update objects. The receiver is required to act upon all TriggerTime changes if the target object is still in the Holding Buffer.

6.3 MOT parameters for the Header update

The "Header update" object carries the parameters listed in the table below. All other parameters shall be ignored by the receiver.

Parameter	Parameter Id	Specified in	Mandatory for service provider	Mandatory for receiver	Occurrence
ContentName	0x0C	MOT EN 301 234 [3]	Yes	Yes	Single
TriggerTime	0x05	present document	Yes	Yes	Single

6.3.1 ContentName

This parameter is used to link the Header update to the slide object, the TriggerTime of which is to be updated.

Simple profile: The ContentName shall refer to the slide that was sent directly before the "Header update". It is not possible to send multiple slides in advance and trigger any one of those. If the "Header update" does not refer to the slide in the holding/rendering buffer then both the "Header update" and the slide shall be removed from their respective buffer in the receiver.

Enhanced profile: The ContentName shall refer to a previously transmitted MOT object (from this service). It is possible to send multiple MOT objects with different ContentName values, and provide subsequent Header Update objects using ContentName to identify which object in the Holding Buffer to update. Header Update objects that refer to an object not in the Holding Buffer may be ignored. Header Updates may cause previously displayed images to be displayed again.

6.3.2 TriggerTime

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

6.4 Optional MOT features

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. Clause 5.3.1 specifies memory management for caching content.

7 Application signalling

The use of the SlideShow user application within a DAB ensemble shall be signalled by the use of FIG 0/13. The UserApplicationType value is given in TS 101 756 [2].

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

NOTE: Although two profiles are defined, the profile is not explicitly signalled.

8 Other requirements

8.1 Display restrictions

Images may be broadcast at any size, shape and colour depth allowed by the image formats supported subject to the restrictions in clause 6.1.1. Pixels are always square.

It is strongly recommended that images intended for SlideShow applications accompanying audio services are authored at a resolution 320 × 240 pixels in landscape format, to prevent rescaling distortion in receivers. Deviation from these dimensions may create significantly sub-optimal display.

8.1.1 Simple profile

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 broadcast 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.

Content providers need to be aware that if they broadcast an image 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.

8.1.2 Enhanced profile

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.

8.2 Object and buffer sizes

The following restrictions apply to objects in the application. If an object exceeds the maximum size, or is signalled to do so, it may be ignored by the receiver.

The following requirements apply to the conceptual buffers in the receiver. Actual implementations may differ from this provided that the result is the same as the conceptual model would have produced.

8.2.1 Simple profile

All receivers 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.

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

8.2.2 Enhanced profile

All receivers shall be able to decode images up to an MOT object size (Body + Header) 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).

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

A.1 Introduction

APNG is an extension of the PNG [4] 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.

A.1.1 Terminology

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

An APNG stream is a normal PNG stream as defined in the PNG Specification [4], 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.2.1 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 illustrates 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.2.2 '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.2.3 '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' >= 0

'y_offset' >= 0

'width' > 0

'height' > 0

'x_offset' + 'width' <= 'IHDR' width

'y_offset' + 'height' <= 'IHDR' 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 1/100ths 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 [PNG-1.2]. 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.2.4 '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/.

An encoder (open source) is available in Mozilla versions newer than alpha 4.

An application (open source) using the Mozilla encoder to assemble APNGs available here: http://littlesvr.ca/apng/apngedit.html.

Annex B (informative): Implementing SlideShow with timeshifted audio services

This annex gives specific guidance to manufacturers implementing SlideShow on receiver devices that include functionality to Pause or Record audio services. 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 timeshifted - otherwise it would give an inconsistent and unsatisfactory experience for both users and service providers.

B.1 SlideShow in X-PAD

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

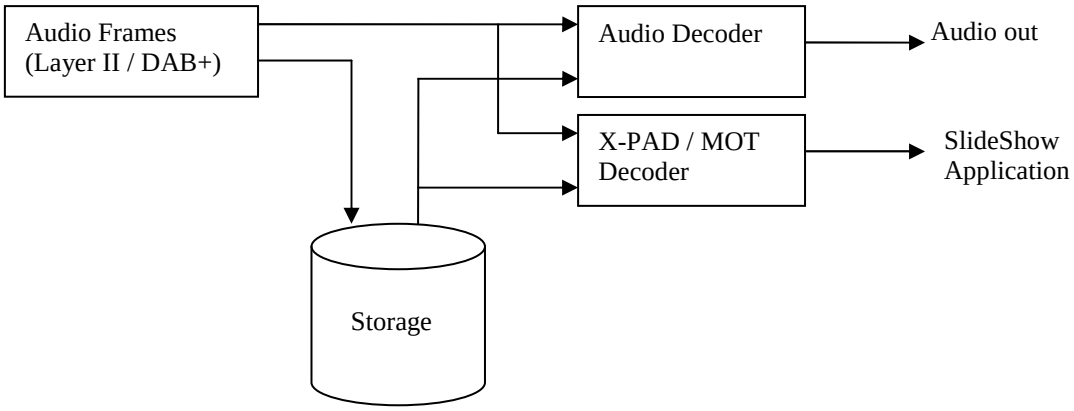


Figure B.1: Conceptual System for Timeshifting Slidehow in X-PAD

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 shall 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 1s at the moment the recording commences (as derived from FIG0/10).

B.2 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 byFIG0/10) with a resolution and accuracy of 1 s at the commencement of the recording.

B.3 Reference time and TriggerTime

The SlideShow application shall use an internal timer to reference time.

For a live broadcast, this shall be synchronized against FIG0/10 as described in clause 5.4. For a timeshifted broadcast, this timer may be initially synchronized against date and time at the commencement of the recording, and adjusted accordingly.

Specifically:

- The reference timer shall take 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") shall be made against the adjusted time.

History

Document history		
V1.1.1	July 2001	Publication
V2.1.1	January 2006	Publication
V2.2.1	July 2008	Publication