ETSI TS 102 635-2 V1.1.1 (2009-08)

Technical Specification

Digital Audio Broadcasting (DAB);
Middleware;
Part 2: DAB

European Broadcasting Union   Union Européenne de Radio-Télévision

EBU·UER

Reference

DTS/JTC-DAB-54-2

Keywords

broadcasting, DAB, digital

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 [9], 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 specifies the additional definitions to apply MATE middleware to Eureka-147 Digital Audio Broadcasting (DAB) (EN 300 401 [9]). Within the present document the term "DAB" is used to refer to the Eureka-147 Digital Audio Broadcasting standard.

This present document defines only the terms that shall be applied to DAB. It includes the definitions of basic data formats, protocols to deliver data, to signal downloadable applications and to download them, ways to denote resources on DAB, and detailed interfaces among receiver platform, broadcast and communication networks, and the applications.

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.

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] ISO/IEC 13818-1 (2000): "Information technology - Generic coding of moving pictures and associated audio - Part 1: Systems".

[2] ISO/IEC 11171-3 (1993): "Information technology - Coding of moving pictures and associated audio for digital storage media at up to 1,5 Mbit/s - Part 3: Audio".

[3] IETF RFC 2396 "Uniform Resource Identifiers (URI): Generic Syntax".

[4] ISO/IEC 14496-12 (2005): "Information technology - Coding of audio-visual objects - Part 12: ISO base media file format".

[5] ISO/IEC 14496-15 (2004): "Information technology - Coding of audio-visual objects - Part 15: Advanced Video Coding (AVC) file format".

[6] ETSI EN 301 234: "Digital Audio Broadcasting (DAB); Multimedia Object Transfer (MOT) protocol".

[7] ETSI TS 101 759: "Digital Audio Broadcasting (DAB); Data Broadcasting - Transparent Data Channel (TDC)".

[8] IETF RFC 2616: "Hypertext Transfer Protocol -- HTTP/1.1".

[9] ETSI EN 300 401: "Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers".

[10] ETSI TS 102 428: "Digital Audio Broadcasting (DAB); DMB video service; User Application Specification".

[11] ETSI TS 102 563: "Digital Audio Broadcasting (DAB); Transport of Advanced Audio Coding (AAC) audio".

[12] ETSI TS 102 818: "Digital Audio Broadcasting (DAB); Digital Radio Mondial (DRM); XML Specification for Electronic Programme Guide (EPG)".

[13] ETSI TS 102 371: "Digital Audio Broadcasting (DAB); Digital Radio Mondiale (DRM); Transportation and Binary Encoding Specification for Electronic Programme Guide (EPG)".

[14] ETSI TS 102 635-1: "Digital Audio Broadcasting (DAB); Middleware; Part 1: System aspects".

[15] ETSI TS 101 756: " Digital Audio Broadcasting (DAB); Registered tables".

[16] TTAS.KO-07.0026/R1 (2006-12): "Specification of the video services for Terrestrial Digital Multimedia Broadcasting (DMB) to mobile, portable and fixed receivers".

[17] IETF RFC 3066: "Tags for the Identification of Languages".

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 Definitions, abbreviations and conventions

3.1 Definitions

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

device driver: system software responsible for basic operation of hardware units

event: group of one or more media with specified start and end times

EXAMPLE: An event can be a soccer half time, a news flash, and so on.

platform standard: standard that is based on MATE system aspects, and designates media-specifics, where media means either terrestrial or satellite DMB

program: group of one or more events being transmitted under a single broadcaster's control

EXAMPLE: A program can be news or entertainment.

service: series of programs being transmitted under a single broadcaster's control

service binding: binding of applications with services

NOTE: An application bound to a service is executed automatically upon user's selection of the service. If the user stops the service, the application is also destroyed.

3.2 Abbreviations

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

API	Application Programming Interface
BNF	Backus-Naur Form
CA	Certificate Authority or Conditional Access
CTS	Composition Time Stamp
DMB	Digital Multimedia Broadcasting
HTTP	Hyper Text Transfer Protocol
JVM	Java Virtual Machine
MATE	Multimedia Application Terminal Environment
MOT	Multimedia Object Transfer
SI	Service Information
STC	System Time Clock
TDC	Transparent Data Channel
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

3.3 Conventions

3.3.1 Syntax of binary messages

The symbols, the abbreviations, and the methods for the description of syntaxes of binary messages in the present document shall follow those defined in clauses 2.2 and 2.3 of ISO/IEC 13818-1 [1].

3.3.2 BNF

Unless otherwise specified, the BNF notation in the present document shall follow the definitions of clause 2.1 of RFC 2616 [8].

4 Introduction

MATE provides a definition of a platform-independent environment, where executable applications can be signalled, transferred to receivers, and executed in the DMB environment. Together with the MATE system aspects [14], the present document specifies only the terms that shall be applied to DAB.

5 System architecture

The system architecture is specified in clause 5 of MATE system aspects [14].

6 Basic data formats

The present document requires support for the following file formats as well as those defined in clause 6 of MATE system aspects [14].

6.1 Audio file format

A receiver complying with this standard shall understand and playback all the audio file formats corresponding to its capabilities as designated in table 1.

Table 1: Mapping between receiver capability and supported audio coding and format

Receiver Capability	Audio Coding	File Format
All DAB Receivers	MPEG-1 Layer II	As defined in ISO/IEC 11171-3 [2]
DMB Profile 1 [9]	MPEG-4 ER BSAC	As defined in ISO/IEC 14496-12 [4]
DMB Profile 2 [9]	MPEG-4 HE AAC V2	As defined in ISO/IEC 14496-12 [4]
DAB AAC (DAB+) [10]	MPEG-4 HE AAC V2	As defined in ISO/IEC 14496-12 [4]

6.2 Video file format

All the DAB receivers supporting DMB application shall understand and playback video files the format of which is defined in clause 15 of ISO/IEC 14496-15 [5]. But the video coding shall be confined to H.264 as specified in [10].

6.3 Font file format

The present document does not support downloadable fonts. Therefore, the format is not specified.

7 Transport protocols

The present document requires support for the protocols specified in this clause, as well as those defined in clause 7 of MATE system aspects [14].

7.1 Broadcast channel protocols

The present document requires full compliance with clause 7.1 of MATE system aspects [14], and additionally designates specific protocols that map to abstract protocol models defined in the same clause of MATE system aspects.

Any compliant implementation of MATE for DAB shall support the file transport protocol, the packet transport protocol, and the trigger protocol that synchronizes application behaviours with AV, as defined in this clause

7.1.1 File transport protocol

The MOT protocol defined in EN 301 234 [6] shall be supported, and it is further constrained as specified in the following clauses.

7.1.1.1 MOT transport mode

Between MOT header mode and directory mode defined in clause 7 of EN 301 234 [6], only directory mode shall be supported.

7.1.1.2 Header extension

Among MOT header extensions, those listed in table 2 shall be supported as designated in the table.

Table 2: MOT header extensions that shall be supported

Name	Mandatory or Optional	Notes
ContentName	Mandatory	Only UTF-8(1111) shall be supported for the character set used.
CompressionType	Mandatory	Only GZIP(0x01) shall be supported
CAInfo	Mandatory	
Expiration	Optional	If caching is implemented, this extension shall be considered as mandatory.
UniqueBodyVersion	Mandatory	
Priority	Optional	
PermitOutdatedVersion	Optional	
RetransmissionDistance	Optional	

7.1.2 Packet transport protocol

Only "TDC in a packet mode service component with data groups" defined in clause 4.1.2 and "TDC in X-PAD with data groups" defined in clause 4.3.2 of TS 101 759 [7] shall be supported. A javax.microedition.Datagram in Java API and a data group in TDC are in one-to-one correspondence.

7.1.3 Trigger protocol

A trigger protocol is for synchronizing application behaviour with other media such as DMB video service, where DMB video service refers to the AV service defined in TS 102 428 [10].

To build a data service synchronized with a DMB video service, trigger time information, at which a specific event should be performed by an application, should be transmitted to the receiver before the trigger time. The packets used for the transport of triggers are called trigger packets. A trigger packet contains trigger time information and data that is used by applications.

7.1.3.1 Format of trigger packet

A trigger packet should be scheduled in the head-end, in order to be transmitted before the designated trigger time. The receiver processes a received trigger packet in 2 steps, which are preparation and execution steps. In the preparation step, a trigger packet is passed to an application to let it prepare for an event to be performed. In the execution step, the same trigger packet is again passed to the application to let it perform an event at the time designated by the trigger.

The format of a trigger packet is as follows:

Table 3: Format of trigger packet

Syntax	No. of Bits	Mnemonic
trigger_packet(){		
Triggered	16	uimsbf
Timeout	8	uimsbf
Unused	7	"0000000"
triggerTime	33	uimsbf
for(i=0;i<N;i++){		
privateDataByte	8	uimsbf
}		
}		

triggerID: An identifier that identifies a trigger among those transmitted in the same trigger stream.

timeout: A trigger packet should be ignored if it is not executed for the duration specified in this field from its reception in the receiver. The unit is 1/10 seconds. This timeout mechanism is devised to cope with the discontinuities and/or temporary drifts within AV clock system. If the time specified by a trigger packet has passed without being traced due to bad reception condition, the invalid trigger can be removed as soon as possible. And the receiver can avoid the problem of activating wrong triggers when the trigger times happen to coincide with some other AV CTSs due to several failures in recognizing discontinuities in AV clock system.

triggerTime: The time point at which an event must be triggered. For instance, for synchronization of application behaviour with video, this field should indicate the video CTS that corresponds to the scene, at which an event designated by the trigger should be performed. The same can be achieved with audio or BIFS in a similar way.

privateDataByte: The data passed to an application at the time designated by the trigger.

7.1.3.2 Transport of trigger packets

"TDC in a packet mode service component with data groups" designated in clause 4.1.2 of TS 101 759 [7] shall be supported. A dmb.io.Trigger in Java API and a data group in TDC are in one-to-one correspondence.

7.1.3.3 Coping with discontinuities in media time

Even in cases of discontinuities in the media time (MPEG-2 STC) in a media such as an AV stream that is to be synchronized with an application, it is possible to transmit triggers, before a media time discontinuity, that refer to the times after the discontinuity. In this way, a trigger time after a discontinuity can be specified if the discontinuity is known in advance.

7.1.4 Dynamic label

In addition to concrete mappings to the abstract protocol models defined in MATE system aspects [14], this standard supports dynamic label segments defined in clause 7.4.5.2 of TS 300 401 [9]. The data conveyed via this protocol is accessed with APIs defined in dmb.dl package in Java environment.

7.2 Communication channel protocols

This specification requires full compliance with clause 7.2 of MATE system aspects [14] for the supported communication channel protocols.

8 DAB locator

This clause defines the syntax of locators for designating ensembles, services, and so on in terrestrial DMB. The general concept and the relationship between locators and Java APIs are as specified in clause 8 of MATE system aspects [14].

8.1 Syntax

This specification defines locators for designating ensembles, services, service components, packet streams, broadcast file systems, trigger streams, ESes within terrestrial DMB service components, and so on. The BNF (as used in RFC 2396 [3]) for the locators is as follows:

```
dab_url = dab_ensemble | dab_service | dab_component | dab_packet | dab_bfs | dab_dl
        | tdmb_trigger | tdmb_component
dab_ensemble = dab_ensemble_scheme "://" ecc eid
dab_ensemble_scheme = "dab.ensemble"
ecc = 2hex
eid = 4hex
dab_service = dab_service_scheme "://" dab_service_address
dab_service_scheme = "dab.service"
dab_service_address = dab_program_service_address | dab_data_service_address
dab_program_service_address = ecc eid "." ecc program_sid
program_sid = 4hex
dab_data_service_address = ecc eid "." data_sid
data_sid = 8hex
dab_component = dab_component_scheme "://" dab_component_address
dab_component_scheme = "dab.component"
dab_component_address = dab_service_address "." scids
scids = hex
dab_packet = dab_packet_scheme "://" dab_component_address
dab_packet_scheme = "dab.packet"
dab_bfs = dab_bfs_scheme "://" dab_component_address "/" path_segments
dab_bfs_scheme = "dab.bfs"
tdmb_trigger = tdmb_trigger_scheme "://" dab_component_address
```

```
tdmb_trigger_scheme = "tdmb.trigger"
tdmb_component = tdmb_component_scheme "://" dab_component_address "." esid
tdmb_component_scheme = "tdmb.component"
esid = 4hex
dab_dl = dab_dl_scheme "://" dab_component_address
dab_dl_scheme = "dab.dl"
hex = digit | "a" | "b" | "c" | "d" | "e" | "f"
digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
(path_segments is defined in IETF RFC 2396[5].)
```

8.2 Ensemble Locator

The formal syntax of the locator designating an ensemble is defined as dab_ensemble in clause 8.1. The syntax is as follows:

- dab.ensemble://<ECC><EId>

where <ECC> and <EId> are ECC(Extended Country Code) and EId(Ensemble Identifier) defined in EN 300 401 [9] respectively, and represented in hexadecimal.

8.3 Service locator

The formal syntax of the locator designating a service is defined as dab_service in clause 8.1. The syntax varies depending on whether the corresponding service is a program or a data service. The syntax in each case is as follows:

- **Program service:** dab.service://<ECC><EId>.<ECC><Program_SId>
- **Data service:** dab.service://<ECC><EId>.<Data_SId>

where <ECC> and <EId> are ECC (Extended Country Code) and EId (Ensemble Identifier) defined in EN 300 401 [9] respectively, and represented in hexadecimal. <Program_SId> and <Data_SId> are SIds (Service Identifiers) for program and data services respectively, as defined in EN 300 401 [9], and are represented in hexadecimal. <Program_SId> is 4 hexadecimal digits, and <Data_SId> is 8 hexadecimal digits

8.4 Service component locator

The formal syntax of the locator designating a service is defined as dab_component in clause 8.1. The syntax varies depending on whether the corresponding service is a program or a data service. The address for a service is defined in clause 8.3, and to further identify a specific component within a service, <SCIdS> is used. It is SCIdS(Service Component Identifier within Service) defined in EN 300 401 [9].

- **Component within a program service:** dab.component://<ECC><EId>.<ECC><Program_SId>.<SCIdS>
- **Component within a data service**: dab.component://<ECC><EId>.<Data_SId>.<SCIdS>

8.5 Packet stream locator

The formal syntax of the locator designating a service is defined as dab_packet in clause 8.1. Compared to a service component locator, only scheme part differs in that dab.packet is used in this case.

- **Packet stream in a program service:** dab.packet://<ECC><EId>.<ECC><Program_SId>.<SCIdS>
- **Packet stream in a data stream:** dab.packet://<ECC><EId>.<Data_SId>.<SCIdS>

8.6 Broadcast file system locator

A broadcast file system means a file transport component in compliance with clause 7.1.1. The formal syntax of the locator designating a broadcast file system is defined as dab_bfs in clause 8.1.

- **Broadcast file system:** dab.bfs://<ECC><EId>.<Data_SId>.<SCIdS>/<path_segments>

where the syntax of path_segments is as defined in RFC 2396 [3], and it corresponds to the value of ContentName header extension for an MOT object within the MOT designated by the locator. That is, a broadcast file system locator designates an MOT object with the value of its ContentName header extension identical to path_segments of the locator.

8.7 Dynamic Label locator

The formal syntax of the locator designating a dynamic label is defined as dab_dl in clause 7.1.1. Compared to a program service component locator, only scheme part differs in that dab.packet is used in this case.

- dab.dl://<ECC><EId>.<ECC><Program_SId>.<SCIdS>

8.8 Trigger stream locator

A trigger stream means a trigger stream component in compliance with clause 7.1.3. The formal syntax of the locator designating a trigger stream is defined as tdmb_trigger in clause 8.1. Compared to a service component locator, only scheme part differs in that dab.trigger is used in this case.

- **Trigger stream within a program service:** tdmb.trigger://<ECC><EId>.<ECC><Program_SId>.<SCIdS>
- **Trigger stream within a data service:** tdmb.trigger://<ECC><EId>.<Data_SId>.<SCIdS>

8.9 Locator for an ES within a terrestrial DMB service component

ES designates that defined in TTAS.KO-07.0026/R1 [16]. The formal syntax of the locator designating an ES is defined as tdmb_component in clause 8.1. The syntax is as follows, and the format is identical to the locator for a service component defined in clause 8.4, except for the additional <ESId>.

- **ES within a program service:** tdmb.component://<ECC><EId>.<ECC><Program_SId>.<SCIdS>.<ESId>
- **ES within a data service:** tdmb.component://<ECC><EId>.<Data_SId>.<SCIdS>.<ESId>

where ESId is a 16-bit identifier as defined in TTAS.KO-07.0026/R1 [16].

9 Security model

This specification requires full compliance with clause 9 of MATE system aspects [14] in terms of security model. But support for the digital signing of applications is optional.

10 Graphic system model

This specification requires full compliance with clause 10 of MATE system aspects [14] in terms of graphic system model.

11 Application model

This specification requires full compliance with clause 11 of MATE system aspects [14] in terms of application model.

12 Application signalling and delivery

This specification requires full compliance with clause 12 of MATE system aspects [14] in terms of application signalling and delivery, and further specifies additional requirements in the following clauses.

12.1 Application signalling

12.1.1 Overview

Clause 12.3 of MATE system aspects defines 5 messages for conveying information on applications to the receiver. In this clause, the mechanism used to transport application information message, module information message, service binding message, and certificate message, are defined. A user application information in FIG 0/13 is used to mark the location where those messages can be found, and the remaining application control message is inserted as the user application data in the same user application information. Also the format of service_locator(), that is for locating a service within service binding message, is defined.

12.1.2 Message transport

With an exception of application control message, all other signalling messages shall be transmitted within an MOT as specified in clause 7.1.1. A service component containing an MOT conveying signalling messages is identified via a user application information in FIG 0/13 with User Application Type of "Middleware" (see TS 101 756 [15]).

Application control message is put in the user application data field of the user application information in FIG 0/13 identifying the MOT carrying the other corresponding signalling messages. Signalling messages other than application control message are identified within an MOT by their names, as specified in table 4.

Table 4: Names of signalling messages

Signalling Message	Name
Application Information Message	.AIM
Module Information Message	.MIM
Service Binding Message	.SBM
Certificate Message	.CM

12.1.3 Signalling message monitoring

The receiver must monitor signalling messages within the currently selected service.

Additionally, if there is a service the primary service component of which is an MOT carrying signalling messages within the ensemble that is currently tuned to, the messages must also be monitored, regardless of the current selection of service.

12.2 Transport of application modules

Application modules are transported via the MOT protocol as designated in clause 7.1.1. Each module corresponds to an MOT object.

12.3 Format of service binding message

In the case of transmitting application signalling messages within services, service binding message can specify the bindings for services other than that carrying the message as well as the bindings for the service carrying the messages. To designate a service within service binding messages, clause 12.5.8 of MATE system aspects uses an opaque service_locator() without specifying its format. In the context of terrestrial DMB, the format of service_locator() is defined as follows:

Table 5: service_locator format

Syntax	No. of Bits	Mnemonic
service_locator(){		
programOrData	1	uimsbf
Reserved	7	"1111111"
ensembleECC	8	uimsbf
ensembleID	16	uimsbf
if (programOrData == "0") {		
programServiceECC	8	uimsbf
programServiceID	16	uimsbf
} else {		
dataServiceID	32	uimsbf
}		

programOrData: If "0", the locator designates a program service, and if "1", a data service.

ensembleECC: Extended Country Code for an ensemble.

ensembleID: Ensemble ID.

programServiceECC: Extended Country Code for a program service.

programServiceID: Program service ID.

dataServiceID: Data service ID.

12.4 Format of application control message

Application control message is inserted to user application data of FIG 0/13 and its format is defined in table 6.

Table 6: User data format

Syntax	No. of Bits	Mnemonic
user_data(){		
service_binding_message_version	32	Uimsbf
for(i=0;i<N;i++){		
binding_tag	8	Uimsbf
}		
}		

service_binding_message_version: The version of a service binding message corresponding to the application control message conveyed in this user data. Only when the version of the corresponding service binding message stored in the receiver is equal to or greater than this value, binding_tag is considered valid.

binding_tag: One of tags that represent bindings between a service and an application in the service binding message corresponding to the application control message conveyed in this user data.

13 Java environment

13.1 Overview

The present document provides an execution environment, for Java applications, consisting of JVM and APIs. JVM and APIs must comply with requirements set in clause 13 of MATE system aspects [14], as well as additional requirements specified in this clause.

13.2 Service information API

In terrestrial DMB, service information is available via 2 different mechanisms, that is, DAB SI defined in EN 300 401 [9] and XML EPG defined in TS 102 818 [12] and conveyed in an MOT. The schema for XML EPG is defined in TS 102 818 [12], and a binary encoding of documents complying with the schema and a transport mechanism via MOT are defined in TS 102 371 [13].

This specification does not require the availability of XML EPG. Therefore, in this clause, it is defined that what data shall be available via SI API in the absence of XML EPG. Also the full set of data to be available via SI API when XML EPG is present.

13.2.1 Notation for designating XML elements and attributes

In the definition of the schema supported by SI API, the following notation is used to designate XML elements or attributes defined in TS 102 818 [12].

Element: Specifies its name. When there is a namespace associated with the element, the prefix assigned to it in the schema file is specified as it is. And if it is required to specify the enclosing element together with an element to locate the element unambiguously, the name of the enclosing element comes first, a ".", and the name of the element follow in that order. For example, a service element within an ensemble element is represented as ensemble.service.

Attribute: Specifies name of the containing element, a "#", and name of the attribute concatenated in that order. For example, ID for an ensemble is represented as ensemble#id.

13.2.2 Service information based on DAB SI

In the absence of XML EPG-based data, SIObjects specified in table 7 are provided via SI API.

Table 7: Information provided based on DAB SI (SIObject)

Objec Type	Attribute	Corresponding XML EPG element or attribute	Notes
ENSEMBLE	ID	ensemble#id	FIG 0/0
	NAME	ensemble.mediumName	FIG 1/0
	FREQUENCY	ensemble.frequency#kHz	FIG 0/21 in kHz
SERVICE	ID	ensemble.service.serviceID#id	FIG 0/2
	NAME	ensemble.service.serviceID.mediumName	FIG 1/1, FIG 1/5

13.2.3 Service information based on XML EPG

The information is based on DAB XML EPG defined in TS 102 818 [12], but supports only the case where XML EPG is transmitted via the mechanism defined in TS 102 371 [13].

13.2.4 Service information objects

SI Objects provided via SI API are categorized into ENSEMBLE, SERVICE, PROGRAM, PROGRAM_EVENT, LOCATION, and GROUP. The constants for the object types are defined in tdmb.si.Types interface.

13.2.4.1 Ensemble

It represents the "ensemble" defined in EN 300 401 [9], and includes one or more services within it. In SI API, SIObjects representing an ensemble have the tdmb.si.Types.ENSEMBLE constant as the value of its type attribute. This type of SI objects corresponds to ensemble element in XML EPG.

13.2.4.2 Service

It represents the "service" defined in EN 300 401 [9]. A service is a series of programs in terms of time, and at a specific moment, physically consists of one or more service components. In SI API, SIObjects representing a service have the tdmb.si.Types.SERVICE constant as the value of its type attribute. This type of SI objects corresponds to service element in XML EPG.

13.2.4.3 Program

It represents the "program(programme)" defined in EN 300 401 [9]. But in this specification, services, and start and end times for a program are separated out as a location (refer to clause 13.2.4.5) so that, for example, the same movie being broadcast in multiple services at different times can be recognized. This is also consistent with the design of XML EPG [11]. In SI API, SIObjects representing a program have the tdmb.si.Types.PROGRAM constant as the value of its type attribute. This type of SI objects corresponds to programme element in XML EPG.

13.2.4.4 Program Event

It represents the "program event(programme event)" defined in XML EPG [11], and designates a meaningful time interval within a program. In SI API, SIObjects representing a program event have the tdmb.si.Types.PROGRAM_EVENT constant as the value of its type attribute. This type of SI objects corresponds to programmeEvent element in XML EPG.

13.2.4.5 Location

It collectively represents a service, and a start time and an end time, respectively where and when a program is broadcast. In cases where a program is broadcast on multiple services at multiple times, the SI object corresponding to the program has the same number of location objects as that of its broadcasts. In SI API, SIObjects representing a location have the tdmb.si.Types.LOCATION constant as the value of its type attribute. This type of SI objects corresponds to location element in XML EPG.

13.2.4.6 Group

It represents the "group" defined in XML EPG [11], and is a collection of programs, program events, or groups. Among programs (or program events) sent out on multiple services at multiple times, those with the same subject may be bound as a group. In addition to that, different groups may be put into a parent group. In SI API, SIObjects representing a group have the tdmb.si.Types.GROUP constant as the value of its type attribute. This type of SI objects corresponds to programmeGroup element in XML EPG.

13.2.5 Data encoding

When determining types of attribute values in Java API from received XML EPG data, in general, the following rules are applied.

Enumerated data values: 1-byte constants defined in TS 102 371 [13] shall be remapped to the original strings defined in TS 102 818 [12], and the strings shall be returned. That is, such attribute values are of java.lang.String type. Note that the returned Strings are all literal strings, so may be compared with other literal strings or String.intern()"ed strings using "==" operator.

Date and time: encoded as a java.util.Date object.

Duration: encoded as an int value representing a duration in seconds.

Content ID: encoded as a java.lang.String that is a locator locating the same entity as the original ID.

Bit rate: encoded as an int value, that is the very representation specified in TS 102 371 [13] for representing bit rates.

kHz: encoded as an int value representing the frequency in kHz.

Token: tokens are represented as strings as in the original schema. The type of such attributes is java.lang.String.

When a datum is designated as optional, and absent in a specific instance of data, then value of the corresponding attribute is null if the attribute is of a reference type, and -1 if the attribute is of a numeric type and its values are confined to non-negative ones.

13.2.6 Attributes of SI objects

In this clause, each attribute of each SI object is described in detail, presenting its name, description, type (Java type), and its relationship with XML EPG.

13.2.6.1 Ensemble

An ensemble object has the following attributes. In the "correspondence to XML EPG" column of table 8, ensemble element is omitted for brevity. Therefore every name in the column should be interpreted relative to ensemble element.

Table 8: Attributes of ensemble object

Attribute	Description	Type	Correspondence to XML EPG
ID	Ensemble locator.	String	#id
SHORT_NAME	Short name.	String	shortName
NAME	Medium name.	String	mediumName
LONG_NAME	Long name.	String	longName
FREQUENCY_TYPE	List of frequency types. This corresponds to frequency#type, and each entry in this list has one-to-one relationship with an entry at the same index in FREQUENCY_KHZ.	String[]	frequency#type
FREQUENCY_KHZ	List of frequencies in kHz. This corresponds to frequency#kHz, and each entry in this list has one-to-one relationship with an entry at the same index in FREQUENCY_TYPE.	int[]	frequency#kHz
SHORT_DESCRIPTION	Short description.	String	mediaDescription.shortDescription
LONG_DESCRIPTION	Long description.	String	mediaDescription.longDescription
MULTIMEDIA_URL	List of URLs locating relevant multimedia data. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#url
MULTIMEDIA_MIME	List of MIME types of relevant multimedia data. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#mimeValue
MULTIMEDIA_TYPE	List of multimedia types relevant to this ensemble. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#type
MULTIMEDIA_WIDTH	List of width of multimedia data relevant to this ensemble. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	int[] (non-negative integers)	mediaDescription.multimedia#width
MULTIMEDIA_HEIGHT	List of height of multimedia data relevant to this ensemble. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	int[] (non-negative integers)	mediaDescription.multimedia#height
CA	CA value.	String	CA
KEYWORDS	List of keywords.	String[]	keywords
LINK_URL	List of URLs locating relevant contents. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	link#url
LINK_MIME	List of MIME types. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	link#mimeValue
LINK_DESCRIPTION	List of descriptions on links. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	link#description
LINK_EXPIRY_TIME	List of expiry times for links relevant to this ensemble. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	Date[]	link#expiryTime
SERVICES	An SIView containing service objects belonging to this ensemble.	SIView	service

13.2.6.2 Service

A service object has the following attributes. In the "correspondence to XML EPG" column of table 9, service element is omitted for brevity. Therefore every name in the column should be interpreted relative to service element.

Table 9: Attributes of service object

Attribute	Description	Type	Correspondence to XML EPG
ID	Service ID (locator).	String	No direct correspondence to XML EPG
COMPONENT_LIST	List of service components (locators). Primary component is placed at index 0.	String[]	serviceID#id
SIMULCAST_SYSTEM	List of network systems carrying the same service. An entry in this list corresponds to an entry at the same index in the value of SIMULCAST_ID attribute.	String[]	simulcast#system
SIMULCAST_ID	List of IDs carrying the same content with this service. An entry in this list corresponds to an entry at the same index in the value of SIMULCAST_ID attribute.	String[]	simulcast#id
FORMAT	Service format.	String	#format
BITRATE	Bit rate.	int	#bitrate
SHORT_NAME	Short name.	String	shortName
NAME	Medium name.	String	mediumName
LONG_NAME	Long name.	String	longName
SHORT_DESCRIPTION	Short description.	String	mediaDescription.shortDescription
LONG_DESCRIPTION	Long description.	String	mediaDescription.longDescription
MULTIMEDIA_URL	List of URLs locating relevant multimedia data. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#url
MULTIMEDIA_MIME	List of MIME types of relevant multimedia data. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#mimeValue
MULTIMEDIA_TYPE	List of multimedia types relevant to this service. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#type
MULTIMEDIA_WIDTH	List of width of multimedia data relevant to this service. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	int[] (non-negative integers)	mediaDescription.multimedia#width
MULTIMEDIA_HEIGHT	List of height of multimedia data relevant to this service. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	int[] (non-negative integers)	mediaDescription.multimedia#height
GENRE	List of genre descriptions, where each description is formatted as follows: Each of CS (classification scheme), level 1, level 2, and level 3 classifications is encoded as a 3-digit decimal representation, and concatenated in that order separated by "."s. If lower level classifications are not required, they can be left out (for example, if classification scheme and level 1 classification are specified, it can be encoded like "005.007")	String[]	genre

Attribute	Description	Type	Correspondence to XML EPG
LANGUAGE	List of languages of a service. Represented with language codes defined in RFC 3066 [17]	String[]	epgLanguage
CA	CA value	String	CA
KEYWORDS	List of keywords	String[]	keywords
LINK_URL	List of URLs locating relevant contents. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	link#url
LINK_MIME	List of MIME types. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	link#mimeValue
LINK_DESCRIPTION	List of descriptions on links. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	link#description
LINK_EXPIRY_TIME	List of expiry times for links relevant to this service. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	Date[]	link#expiryTime
ENSEMBLE	An ensemble containing this service (an SIView containing only one ensemble object).	SIView	No direct correspondence to XML EPG

13.2.6.3 Program

A program object has the following attributes. In the "correspondence to XML EPG" column of table 10, programme element is omitted for brevity. Therefore every name in the column should be interpreted relative to programme element.

Table 10: Attributes of program object

Attribute	Description	Type	Correspondence to XML EPG
ID	CRID for a program.	String	#id
SHORT_ID	Short CRID for a program.	int	#shortId
RECOMMENDATION	Whether a program is a recommended one or not.	boolean	#recommendation
ON_AIR	Whether a program is on air or not. If it is on-air, the value of this attribute is true, otherwise, false.	boolean	#broadcast
BITRATE	Bit rate.	int	#bitrate
SHORT_NAME	Short name.	String	shortName
NAME	Medium name.	String	mediumName
LONG_NAME	Long name.	String	longName
SHORT_DESCRIPTION	Short description.	String	mediaDescription.shortDescription
LONG_DESCRIPTION	Long description.	String	mediaDescription.longDescription
MULTIMEDIA_URL	List of URLs locating relevant multimedia data. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#url
MULTIMEDIA_MIME	List of MIME types of relevant multimedia data. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#mimeValue

Attribute	Description	Type	Correspondence to XML EPG
MULTIMEDIA_TYPE	List of multimedia types relevant to this program. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#type
MULTIMEDIA_WIDTH	List of width of multimedia data relevant to this program. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	int[] (non-negative integers)	mediaDescription.multimedia#width
MULTIMEDIA_HEIGHT	List of height of multimedia data relevant to this program. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	int[] (non-negative integers)	mediaDescription.multimedia#height
GENRE	List of genre descriptions, where each description is formatted as follows: Each of CS (classification scheme), level 1, level 2, and level 3 classifications is encoded as a 3-digit decimal representation, and concatenated in that order separated by "."s. If lower level classifications are not required, they can be left out (for example, if classification scheme and level 1 classification are specified, it can be encoded like "005.007").	String[]	genre
CA	CA value.	String	CA
KEYWORDS	List of keywords.	String[]	keywords
LINK_URL	List of URLs locating relevant contents. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	link#url
LINK_MIME	List of MIME types. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	link#mimeValue
LINK_DESCRIPTION	List of descriptions on links. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	link#description
LINK_EXPIRY_TIME	List of expiry times for links relevant to this program. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	Date[]	link#expiryTime
PROGRAM_EVENTS	List of program events in this program (contains program event objects)	SIView	programmeEvent
GROUPS	List of groups this program belongs to (contains group objects)	SIView	memberOf
LOCATIONS	List of locations where this program is available (contains location objects)	SIView	Location

13.2.6.4 Program event

A program event object has the following attributes. In the "correspondence to XML EPG" column of table 11, programmeEvent element is omitted for brevity. Therefore every name in the column should be interpreted relative to programmeEvent element.

Table 11: Attributes of program event object

Attribute	Description	Type	Correspondence to XML EPG
ID	CRID for a program event.	String	#id
SHORT_ID	Short CRID for a program event.	int	#shortId
RECOMMENDATION	Whether a program event is a recommended one or not.	boolean	#recommendation
ON_AIR	Whether a program event is on air or not. If it is on-air, the value of this attribute is true, otherwise, false.	boolean	#broadcast
SHORT_NAME	Short name.	String	shortName
NAME	Medium name.	String	mediumName
LONG_NAME	Long name.	String	longName
SHORT_DESCRIPTION	Short description.	String	mediaDescription.shortDescription
LONG_DESCRIPTION	Long description.	String	mediaDescription.longDescription
MULTIMEDIA_URL	List of URLs locating relevant multimedia data. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#url
MULTIMEDIA_MIME	List of MIME types of relevant multimedia data. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#mimeValue
MULTIMEDIA_TYPE	List of multimedia types relevant to this program event. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#type
MULTIMEDIA_WIDTH	List of width of multimedia data relevant to this program event. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	int[] (non-negative integers)	mediaDescription.multimedia#width
MULTIMEDIA_HEIGHT	List of height of multimedia data relevant to this program event. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	int[] (non-negative integers)	mediaDescription.multimedia#height
GENRE	List of genre descriptions, where each description is formatted as follows: Each of CS (classification scheme), level 1, level 2, and level 3 classifications is encoded as a 3-digit decimal representation, and concatenated in that order separated by "."s. If lower level classifications are not required, they can be left out (for example, if classification scheme and level 1 classification are specified, it can be encoded like "005.007").	String[]	genre
CA	CA value.	String	CA
KEYWORDS	List of keywords.	String[]	keywords
LINK_URL	List of URLs locating relevant contents. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	link#url
LINK_MIME	List of MIME types. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	link#mimeValue
LINK_DESCRIPTION	List of descriptions on links. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	link#description

Attribute	Description	Type	Correspondence to XML EPG
LINK_EXPIRY_TIME	List of expiry times for links relevant to this program event. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	Date[]	link#expiryTime
PROGRAM	The program this program event belongs to (contains a program object).	SIView	No direct correspondence to XML EPG
GROUPS	The group this program event belongs to (contains group objects).	SIView	memberOf
LOCATIONS	The locations where this program event is available (contains location objects).	SIView	location

13.2.6.5 Location

A location object has the following attributes. In the "correspondence to XML EPG" column of table 12, location element is omitted for brevity. Therefore every name in the column should be interpreted relative to location element.

Table 12: Attributes of location object

Attribute	Description	Type	Correspondence to XML EPG
TIME	Start time.	Date	time#time
RELATIVE_TIME	Relative start time (valid only for program events).	int	relativeTime#time
DURATION	Duration.	int	time#duration
ACTUAL_TIME	Actual start time.	Date	time#actualTime
RELATIVE_ACTUAL_TIME	Relative actual start time (valid only for program events)	int	relativeTime#actualTime
ACTUAL_DURATION	Actual duration.	int	time#actualDuration
BEARER_ID	List of IDs that locate the location where the corresponding program or program event is available. Each entry in this list corresponds one-to-one to an entry at the same index in the value of BEARER_TRIGGER attribute.	String	bearer#id
BEARER_TRIGGER	List of triggers. Each entry in this list corresponds one-to-one to an entry at the same index in the value of BEARER_ID attribute. Each trigger is represented as an 8-digit hexadecimal number of the value.	String	bearer#trigger
PROGRAM	The program associated with this location object if this is for a program (contains one program object). Otherwise, the value is null.	SIView	No direct correspondence to XML EPG
PROGRAM_EVENT	The program event associated with this location object if this is for a program event (contains one program event object). Otherwise, the value is null.	SIView	No direct correspondence to XML EPG

13.2.6.6 Group

A group object has the following attributes. In the "correspondence to XML EPG" column of table 13, programmeGroup element is omitted for brevity. Therefore every name in the column should be interpreted relative to programmeGroup element.

Table 13: Attributes of group object

Attribute	Description	Type	Correspondence to XML EPG
ID	CRID for a group.	String	#id
SHORT_ID	Short CRID for a group.	int	#shortId
GROUP_TYPE	Group type.	String	#type
SHORT_NAME	Short name.	String	shortName
NAME	Medium name.	String	mediumName
LONG_NAME	Long name.	String	longName
SHORT_DESCRIPTION	Short description.	String	mediaDescription.shortDescription
LONG_DESCRIPTION	Long description.	String	mediaDescription.longDescription
MULTIMEDIA_URL	List of URLs locating relevant multimedia data. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#url
MULTIMEDIA_MIME	List of MIME types of relevant multimedia data. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#mimeValue
MULTIMEDIA_TYPE	List of multimedia types relevant to this group. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	String[]	mediaDescription.multimedia#type
MULTIMEDIA_WIDTH	List of width of multimedia data relevant to this group. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	int[] (non-negative integers)	mediaDescription.multimedia#width
MULTIMEDIA_HEIGHT	List of height of multimedia data relevant to this group. Entries at the same index in each attribute whose name begins with MULTIMEDIA form a description on a multimedia data.	int[] (non-negative integers)	mediaDescription.multimedia#height
GENRE	List of genre descriptions, where each description is formatted as follows: Each of CS (classification scheme), level 1, level 2, and level 3 classifications is encoded as a 3-digit decimal representation, and concatenated in that order separated by "."s. If lower level classifications are not required, they can be left out (for example, if classification scheme and level 1 classification are specified, it can be encoded like "005.007").	String[]	genre
KEYWORDS	List of keywords.	String[]	keywords
LINK_URL	List of URLs locating relevant contents. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	Link#url
LINK_MIME	List of MIME types. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	Link#mimeValue
LINK_DESCRIPTION	List of descriptions on links. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	String[]	Link#description
LINK_EXPIRY_TIME	List of expiry times for links relevant to this group. Entries at the same index in each attribute whose name begins with LINK form a description on a link.	Date[]	Link#expiryTime

Attribute	Description	Type	Correspondence to XML EPG
PARENT_GROUP	The parent group of this group (contains a group object). If this group has no parent, the value is null.	SIView	memberOf
GROUPS	List of groups contained in this group (contains group objects).	SIView	No direct correspondence to XML EPG
PROGRAMS	List of groups contained in this group (contains program objects).	SIView	No direct correspondence to XML EPG
PROGRAM_EVENTS	List of program events (contains program events).	SIView	No direct correspondence to XML EPG

Annex A (informative): API specification

Package Summary		***Page***
dmb.epg	Defines attributes for SI objects representing EPG data used for DAB/T-DMB.	*30*

Package dmb.epg

Defines attributes for SI objects representing EPG data used for DAB/T-DMB.

See:

Description

Interface Summary		Page
Common	Defines the attributes common to all kinds of SI objects, including ensemble, service, program, program event, location, and group.	30
Ensemble	Defines the attribute keys specific to `dmb.si.SIObjects` representing ensembles and an object that is a possible value of `SIAttribute.TYPE` attribute and identifies ensemble objects (`TYPE`).	34
Group	Defines the attribute keys specific to `dmb.si.SIObjects` representing groups and an object that is a possible value of `SIAttribute.TYPE` attribute and identifies group objects (`TYPE`).	36
Location	Defines the attribute keys specific to `dmb.si.SIObjects` representing locations and an object that is a possible value of `SIAttribute.TYPE` attribute and identifies location objects (`TYPE`).	38
Program	Defines the attribute keys specific to `dmb.si.SIObjects` representing programs and an object that is a possible value of `SIAttribute.TYPE` attribute and identifies program objects (`TYPE`).	41
ProgramEvent	Defines the attribute keys specific to `dmb.si.SIObjects` representing program events and an object that is a possible value of `SIAttribute.TYPE` attribute and identifies program event objects (`TYPE`).	44
Service	Defines the attribute keys specific to `dmb.si.SIObjects` representing services and an object that is a possible value of `SIAttribute.TYPE` attribute and identifies service objects (`TYPE`).	46

Package dmb.epg Description

Defines attributes for SI objects representing EPG data used for DAB/T-DMB.

Interface Common

dmb.epg

All Known Subinterfaces:

Ensemble, Group, Program, ProgramEvent, Service

---

public interface **Common**

Defines the attributes common to all kinds of SI objects, including ensemble, service, program, program event, location, and group. Each constant defined here is an attribute key for `dmb.si.SIObjects`.

Field Summary		*Page*
String	ID The ID of an dmb.si.SIObject.	*32*
String	KEYWORDS List of keywords for an dmb.si.SIObject.	*33*
String	LINK_DESCRIPTION List of descriptions for contents associated with an dmb.si.SIObject.	*34*
String	LINK_EXPIRY_TIME List of expiry times for contents associated with an dmb.si.SIObject.	*34*
String	LINK_MIME List of MIME types for contents associated with an dmb.si.SIObject.	*34*
String	LINK_URL List of URLs locating contents associated with an dmb.si.SIObject.	*34*
String	LONG_DESCRIPTION The long description for an dmb.si.SIObject.	*32*
String	LONG_NAME The long name of an dmb.si.SIObject.	*32*
String	MULTIMEDIA_HEIGHT List of the heights of multimedia resources associated with an entity represented by an dmb.si.SIObject.	*33*
String	MULTIMEDIA_MIME List of MIME types of multimedia resources associated an entity represented by an dmb.si.SIObject.	*33*
String	MULTIMEDIA_TYPE List of the types of multimedia resources associated an entity represented by an dmb.si.SIObject.	*33*
String	MULTIMEDIA_URL List of URLs pointing to multimedia resources associated an entity represented by an dmb.si.SIObject.	*33*
String	MULTIMEDIA_WIDTH List of the widths of multimedia resources associated with an entity represented by an dmb.si.SIObject.	*33*
String	NAME The name of an dmb.si.SIObject.	*32*
String	SHORT_DESCRIPTION The short description for an dmb.si.SIObject.	*32*
String	SHORT_NAME The short name of an dmb.si.SIObject.	*32*

Field Detail

ID

```
public static final String ID = "id"
```

The ID of an `dmb.si.SIObject`. The value is a locator or CRID string depending on the type of each `dmb.si.SIObject`, and represented as a `String`.

SHORT_NAME

```
public static final String SHORT_NAME = "shortName"
```

The short name of an `dmb.si.SIObject`. The value is a `String`.

NAME

```
public static final String NAME = "name"
```

The name of an `dmb.si.SIObject`. The value is a `String`.

LONG_NAME

```
public static final String LONG_NAME = "longName"
```

The long name of an `dmb.si.SIObject`. The value is a `String`.

SHORT_DESCRIPTION

```
public static final String SHORT_DESCRIPTION = "shortDescription"
```

The short description for an `dmb.si.SIObject`. The value is a `String`.

LONG_DESCRIPTION

```
public static final String LONG_DESCRIPTION = "longDescription"
```

The long description for an `dmb.si.SIObject`. The value is a `String`.

MULTIMEDIA_URL

```
public static final String MULTIMEDIA_URL = "multimediaURL"
```

List of URLs pointing to multimedia resources associated an entity represented by an `dmb.si.SIObject`. The value of this attribute is an array of `String`.

MULTIMEDIA_MIME

```
public static final String MULTIMEDIA_MIME = "multimediaMIME"
```

List of MIME types of multimedia resources associated an entity represented by an `dmb.si.SIObject`. The value of this attribute is an array of `String`.

MULTIMEDIA_TYPE

```
public static final String MULTIMEDIA_TYPE = "multimediaType"
```

List of the types of multimedia resources associated an entity represented by an `dmb.si.SIObject`. The value of this attribute is an array of `String`.

MULTIMEDIA_WIDTH

```
public static final String MULTIMEDIA_WIDTH = "multimediaWidth"
```

List of the widths of multimedia resources associated with an entity represented by an `dmb.si.SIObject`. The value of this attribute is an array of `int`.

MULTIMEDIA_HEIGHT

```
public static final String MULTIMEDIA_HEIGHT = "multimediaHeight"
```

List of the heights of multimedia resources associated with an entity represented by an `dmb.si.SIObject`. The value of this attribute is an array of `int`.

KEYWORDS

```
public static final String KEYWORDS = "keywords"
```

List of keywords for an `dmb.si.SIObject`. The value of this attribute is an array of `String`s.

LINK_URL

```
public static final String LINK_URL = "linkURL"
```

List of URLs locating contents associated with an `dmb.si.SIObject`. The value of this attribute is an array of `String`s.

LINK_MIME

```
public static final String LINK_MIME = "linkMIME"
```

List of MIME types for contents associated with an `dmb.si.SIObject`. The value of this attribute is an array of `String`s.

LINK_DESCRIPTION

```
public static final String LINK_DESCRIPTION = "linkDescription"
```

List of descriptions for contents associated with an `dmb.si.SIObject`. The value of this attribute is an array of `String`s.

LINK_EXPIRY_TIME

```
public static final String LINK_EXPIRY_TIME = "linkExpiryTime"
```

List of expiry times for contents associated with an `dmb.si.SIObject`. The value of this attribute is an array of `Date`s.

Interface Ensemble

dmb.epg

All Superinterfaces:

Common

public interface **Ensemble**

extends Common

Defines the attribute keys specific to `dmb.si.SIObjects` representing ensembles and an object that is a possible value of `SIAttribute.TYPE` attribute and identifies ensemble objects (`TYPE`).

Field Summary		*Page*
String	**CA** The CA identifier for an dmb.si.SIObject.	36
String	**FREQUENCY_KHZ** List of frequencies, where an ensemble is present.	35
String	**FREQUENCY_TYPE** List of the types of frequencies, where an ensemble is present.	35
String	**SERVICES** List of services belonging to an ensemble.	36
Object	**TYPE** The value of the SIAttribute.TYPE attribute representing that an dmb.si.SIObject is an ensemble.	35

Fields inherited from interface dmb.epg.Common
ID, KEYWORDS, LINK_DESCRIPTION, LINK_EXPIRY_TIME, LINK_MIME, LINK_URL, LONG_DESCRIPTION, LONG_NAME, MULTIMEDIA_HEIGHT, MULTIMEDIA_MIME, MULTIMEDIA_TYPE, MULTIMEDIA_URL, MULTIMEDIA_WIDTH, NAME, SHORT_DESCRIPTION, SHORT_NAME

Field Detail

TYPE

```
public static final Object TYPE
```

The value of the SIAttribute.TYPE attribute representing that an dmb.si.SIObject is an ensemble.

FREQUENCY_TYPE

```
public static final String FREQUENCY_TYPE = "frequencyType"
```

List of the types of frequencies, where an ensemble is present. The value of this attribute is an array of Strings.

FREQUENCY_KHZ

```
public static final String FREQUENCY_KHZ = "frequencyKHz"
```

List of frequencies, where an ensemble is present. The value of this attribute is an array of int values.

CA

```
public static final String CA = "ca"
```

The CA identifier for an `dmb.si.SIObject`. The value of this attribute is a `String`.

SERVICES

```
public static final String SERVICES = "services"
```

List of services belonging to an ensemble. The value of this attribute is an `dmb.si.SIView` containing `dmb.si.SIObjects` of `Service.TYPE` type.

Interface Group

dmb.epg

All Superinterfaces:

Common

public interface **Group**

extends Common

Defines the attribute keys specific to `dmb.si.SIObjects` representing groups and an object that is a possible value of `SIAttribute.TYPE` attribute and identifies group objects (`TYPE`).

Field Summary		***Page***
`String`	**GENRE** List of genres for a group.	*37*
`String`	**GROUP_TYPE** The type of a group.	*37*
`String`	**GROUPS** List of groups contained in a group.	*38*
`String`	**PARENT_GROUP** The supergroup containing this group.	*37*
`String`	**PROGRAM_EVENTS** List of program events.	*38*
`String`	**PROGRAMS** List of programs contained in a group.	*38*

String	**SHORT_ID** The short CRID for a group.	*37*
Object	**TYPE** The value of the `SIAttribute.TYPE` attribute representing that an `dmb.si.SIObject` is a group.	*37*

Fields inherited from interface dmb.epg.Common
`ID, KEYWORDS, LINK_DESCRIPTION, LINK_EXPIRY_TIME, LINK_MIME, LINK_URL, LONG_DESCRIPTION, LONG_NAME, MULTIMEDIA_HEIGHT, MULTIMEDIA_MIME, MULTIMEDIA_TYPE, MULTIMEDIA_URL, MULTIMEDIA_WIDTH, NAME, SHORT_DESCRIPTION, SHORT_NAME`

Field Detail

TYPE

```
public static final Object TYPE
```

The value of the `SIAttribute.TYPE` attribute representing that an `dmb.si.SIObject` is a group.

SHORT_ID

```
public static final String SHORT_ID = "shortID"
```

The short CRID for a group. The value of this attribute is an `int`.

GROUP_TYPE

```
public static final String GROUP_TYPE = "groupType"
```

The type of a group. The value of this attribute is a `String`.

GENRE

```
public static final String GENRE = "genre"
```

List of genres for a group. The value of this attribute is an array of `String`s.

PARENT_GROUP

```
public static final String PARENT_GROUP = "parentGroup"
```

The supergroup containing this group. The value of this attribute is an `dmb.si.SIView` containing only one `dmb.si.SIObject` of `TYPE` type.

GROUPS

```
public static final String GROUPS = "groups"
```

List of groups contained in a group. The value of this attribute is an `dmb.si.SIView` containing `dmb.si.SIObjects` of `TYPE`.

PROGRAMS

```
public static final String PROGRAMS = "programs"
```

List of programs contained in a group. The value of this attribute is an `dmb.si.SIView` containing `dmb.si.SIObjects` of `Program.TYPE` type.

PROGRAM_EVENTS

```
public static final String PROGRAM_EVENTS = "programEvents"
```

List of program events. The value of this attribute is an `dmb.si.SIView` containing `dmb.si.SIObjects` of `ProgramEvent.TYPE` type.

Interface Location

dmb.epg

public interface **Location**

Defines the attribute keys specific to `dmb.si.SIObjects` representing locations and an object that is a possible value of `SIAttribute.TYPE` attribute and identifies location objects (`TYPE`).

Field Summary		*Page*
`String`	**`ACTUAL_DURATION`** The actual duration of a location.	*40*
`String`	**`ACTUAL_TIME`** The actual start time of a location.	*40*

Field Summary		*Page*
String	**BEARER_ID** The locator pointing to the location on a network where the target program or program event is presented.	*40*
String	**BEARER_TRIGGER** The trigger bound in the broadcast system that indicates when a program is being broadcast.	*41*
String	**DURATION** The duration of a location.	*40*
String	**PROGRAM** A program a location of which is described by a location object.	*41*
String	**PROGRAM_EVENT** A program event a location of which is described by a location object.	*41*
String	**RELATIVE_ACTUAL_TIME** The actual start time of a location relative to a program where the program event a location of which is described with the location object.	*40*
String	**RELATIVE_TIME** The start time of a location relative to a program containing the program event, a location of which is described by the location object.	*40*
String	**TIME** The start time of a location.	*39*
Object	**TYPE** The value of the `SIAttribute.TYPE` attribute representing that an `dmb.si.SIObject` is a location.	*39*

Field Detail

TYPE

```
public static final Object TYPE
```

The value of the `SIAttribute.TYPE` attribute representing that an `dmb.si.SIObject` is a location.

TIME

```
public static final String TIME = "time"
```

The start time of a location. The value of this attribute is a `Date` object.

RELATIVE_TIME

```
public static final String RELATIVE_TIME = "relativeTime"
```

The start time of a location relative to a program containing the program event, a location of which is described by the location object. The value of this attribute is an `int` value.

DURATION

```
public static final String DURATION = "duration"
```

The duration of a location. The value of this attribute is an `int` value.

ACTUAL_TIME

```
public static final String ACTUAL_TIME = "actualTime"
```

The actual start time of a location. The value of this attribute is a `Date` object.

RELATIVE_ACTUAL_TIME

```
public static final String RELATIVE_ACTUAL_TIME = "relativeActualTime"
```

The actual start time of a location relative to a program where the program event a location of which is described with the location object. The value of this attribute is an `int` value.

ACTUAL_DURATION

```
public static final String ACTUAL_DURATION = "actualDuration"
```

The actual duration of a location. The value of this attribute is an `int` value.

BEARER_ID

```
public static final String BEARER_ID = "bearerID"
```

The locator pointing to the location on a network where the target program or program event is presented. The value of this attribute is a `String`.

BEARER_TRIGGER

```
public static final String BEARER_TRIGGER = "bearerTrigger"
```

The trigger bound in the broadcast system that indicates when a program is being broadcast. The type of this attribute is a `String`.

PROGRAM

```
public static final String PROGRAM = "program"
```

A program a location of which is described by a location object. The value of this attribute is an `dmb.si.SIView` containing an `dmb.si.SIObject` of `Program.TYPE` type.

PROGRAM_EVENT

```
public static final String PROGRAM_EVENT = "programEvent"
```

A program event a location of which is described by a location object. The value of this attribute is an `dmb.si.SIView` containing an `dmb.si.SIObject` of `ProgramEvent.TYPE` type.

Interface Program

dmb.epg

All Superinterfaces:

Common

---

public interface **Program**

extends Common

Defines the attribute keys specific to `dmb.si.SIObjects` representing programs and an object that is a possible value of `SIAttribute.TYPE` attribute and identifies program objects (`TYPE`).

Field Summary		*Page*
`String`	**`BITRATE`** The bitrate of a program.	*43*
`String`	**`CA`** The CA identifier for an `dmb.si.SIObject`.	*43*

Field Summary		***Page***
`String`	**GENRE** List of genres of a program.	*43*
`String`	**GROUPS** List of groups of which a program is a member.	*44*
`String`	**LOCATIONS** List of locations where a program is presented.	*44*
`String`	**ON_AIR** Indicates whether a program is on-air or not.	*43*
`String`	**PROGRAM_EVENTS** List of program events belonging to a program.	*43*
`String`	**RECOMMENDATION** Indicates whether a program is recommended or not.	*43*
`String`	**SHORT_ID** The short CRID for a program.	*42*
`Object`	**TYPE** The value of the `SIAttribute.TYPE` attribute representing that an `dmb.si.SIObject` is a program.	*42*

Fields inherited from interface dmb.epg.Common
`ID, KEYWORDS, LINK_DESCRIPTION, LINK_EXPIRY_TIME, LINK_MIME, LINK_URL, LONG_DESCRIPTION, LONG_NAME, MULTIMEDIA_HEIGHT, MULTIMEDIA_MIME, MULTIMEDIA_TYPE, MULTIMEDIA_URL, MULTIMEDIA_WIDTH, NAME, SHORT_DESCRIPTION, SHORT_NAME`

Field Detail

TYPE

```
public static final Object TYPE
```

The value of the `SIAttribute.TYPE` attribute representing that an `dmb.si.SIObject` is a program.

SHORT_ID

```
public static final String SHORT_ID = "shortID"
```

The short CRID for a program. The value of this attribute is an `int`.

RECOMMENDATION

```
public static final String RECOMMENDATION = "recommendation"
```

Indicates whether a program is recommended or not. The value of this attribute is a `boolean` value.

ON_AIR

```
public static final String ON_AIR = "onAir"
```

Indicates whether a program is on-air or not. The value of this attribute is a `boolean` value.

BITRATE

```
public static final String BITRATE = "bitrate"
```

The bitrate of a program. The value of this attribute is an `int` value.

GENRE

```
public static final String GENRE = "genre"
```

List of genres of a program. The value of this attribute is an array of `String`s.

CA

```
public static final String CA = "ca"
```

The CA identifier for an `dmb.si.SIObject`. The value of this attribute is a `String`.

PROGRAM_EVENTS

```
public static final String PROGRAM_EVENTS = "programEvents"
```

List of program events belonging to a program. The value of this attribute is an `dmb.si.SIView` containing `dmb.si.SIObject`s of `ProgramEvent.TYPE` type.

GROUPS

`public static final String **GROUPS** = "groups"`

List of groups of which a program is a member. The value of this attribute is an `dmb.si.SIView` containing `dmb.si.SIObjects` of `Group.TYPE` type.

LOCATIONS

`public static final String **LOCATIONS** = "locations"`

List of locations where a program is presented. The value of this attribute is an `dmb.si.SIView` containing `dmb.si.SIObjects` of `Location.TYPE` type.

Interface ProgramEvent

dmb.epg

All Superinterfaces:

Common

---

public interface **ProgramEvent**

extends Common

Defines the attribute keys specific to `dmb.si.SIObjects` representing program events and an object that is a possible value of `SIAttribute.TYPE` attribute and identifies program event objects (`TYPE`).

Field Summary		*Page*
String	**CA** The CA identifier for an `dmb.si.SIObject`.	*46*
String	**GENRE** List of genres of a program event.	*46*
String	**GROUPS** List of groups of which a program event is a member.	*46*
String	**LOCATIONS** List of locations where a program event is presented.	*46*
String	**ON_AIR** Indicates whether a program event is on-air or not.	*45*

Field Summary		Page
String	**PROGRAM** A program where a program event belongs.	*46*
String	**RECOMMENDATION** Indicates whether a program event is recommended or not.	*45*
String	**SHORT_ID** The short CRID for a program event.	*45*
Object	**TYPE** The value of the `SIAttribute.TYPE` attribute representing that an `dmb.si.SIObject` is a program event.	*45*

Fields inherited from interface dmb.epg.Common
ID, KEYWORDS, LINK_DESCRIPTION, LINK_EXPIRY_TIME, LINK_MIME, LINK_URL, LONG_DESCRIPTION, LONG_NAME, MULTIMEDIA_HEIGHT, MULTIMEDIA_MIME, MULTIMEDIA_TYPE, MULTIMEDIA_URL, MULTIMEDIA_WIDTH, NAME, SHORT_DESCRIPTION, SHORT_NAME

Field Detail

TYPE

```
public static final Object TYPE
```

The value of the `SIAttribute.TYPE` attribute representing that an `dmb.si.SIObject` is a program event.

SHORT_ID

```
public static final String SHORT_ID = "shortID"
```

The short CRID for a program event. The value of this attribute is an `int`.

RECOMMENDATION

```
public static final String RECOMMENDATION = "recommendation"
```

Indicates whether a program event is recommended or not. The value of this attribute is a `boolean` value.

ON_AIR

```
public static final String ON_AIR = "onAir"
```

Indicates whether a program event is on-air or not. The value of this attribute is a `boolean` value.

GENRE

`public static final String **GENRE** = "genre"`

List of genres of a program event. The value of this attribute is an array of `String`s.

CA

`public static final String **CA** = "ca"`

The CA identifier for an `dmb.si.SIObject`. The value of this attribute is a `String`.

PROGRAM

`public static final String **PROGRAM** = "program"`

A program where a program event belongs. The value of this attribute is an `dmb.si.SIView` containing an `dmb.si.SIObject` of `Program.TYPE` type.

GROUPS

`public static final String **GROUPS** = "groups"`

List of groups of which a program event is a member. The value of this attribute is an `dmb.si.SIView` containing `dmb.si.SIObject`s of `Group.TYPE` type.

LOCATIONS

`public static final String **LOCATIONS** = "locations"`

List of locations where a program event is presented. The value of this attribute is an `dmb.si.SIView` containing `dmb.si.SIObject`s of `Location.TYPE` type.

Interface Service

dmb.epg

All Superinterfaces:

Common

public interface **Service**

extends Common

Defines the attribute keys specific to `dmb.si.SIObjects` representing services and an object that is a possible value of `SIAttribute.TYPE` attribute and identifies service objects (`TYPE`).

Field Summary		*Page*
`String`	**`BITRATE`** The bitrate allocated for a service.	*48*
`String`	**`CA`** The CA identifier for an `dmb.si.SIObject`.	*49*
`String`	**`COMPONENT_LIST`** List of service components in a service (list of locator strings).	*48*
`String`	**`ENSEMBLE`** An ensemble where a service belongs.	*49*
`String`	**`FORMAT`** The format of a service.	*48*
`String`	**`GENRE`** List of genres of a service.	*48*
`String`	**`LANGUAGE`** List of languages of a service.	*49*
`String`	**`SIMULCAST_ID`** List of the service IDs representing the services simulcasting a service.	*48*
`String`	**`SIMULCAST_SYSTEM`** List of network systems where a service is being simulcast.	*48*
`Object`	**`TYPE`** The value of the `SIAttribute.TYPE` attribute representing that an `dmb.si.SIObject` is a service.	*47*

Fields inherited from interface dmb.epg.Common
`ID`, `KEYWORDS`, `LINK_DESCRIPTION`, `LINK_EXPIRY_TIME`, `LINK_MIME`, `LINK_URL`, `LONG_DESCRIPTION`, `LONG_NAME`, `MULTIMEDIA_HEIGHT`, `MULTIMEDIA_MIME`, `MULTIMEDIA_TYPE`, `MULTIMEDIA_URL`, `MULTIMEDIA_WIDTH`, `NAME`, `SHORT_DESCRIPTION`, `SHORT_NAME`

Field Detail

TYPE

```
public static final Object TYPE
```

The value of the `SIAttribute.TYPE` attribute representing that an `dmb.si.SIObject` is a service.

COMPONENT_LIST

`public static final String **COMPONENT_LIST** = "componentList"`

List of service components in a service (list of locator strings). Note that the primary component is located at index 0. The value of this attribute is an array of `String`s.

SIMULCAST_SYSTEM

`public static final String **SIMULCAST_SYSTEM** = "simulcastSystem"`

List of network systems where a service is being simulcast. The value of this attribute is an array of `String`s.

SIMULCAST_ID

`public static final String **SIMULCAST_ID** = "simulcastID"`

List of the service IDs representing the services simulcasting a service. The value of this attribute is an array of `String`s.

FORMAT

`public static final String **FORMAT** = "format"`

The format of a service. The value of this attribute is a `String`.

BITRATE

`public static final String **BITRATE** = "bitrate"`

The bitrate allocated for a service. The value of this attribute is an `int` value.

GENRE

`public static final String **GENRE** = "genre"`

List of genres of a service. The value of this attribute is an array of `String`s.

CA

```
public static final String CA = "ca"
```

The CA identifier for an `dmb.si.SIObject`. The value of this attribute is a `String`.

LANGUAGE

```
public static final String LANGUAGE = "language"
```

List of languages of a service. The value of this attribute is an array of `String`s, each of which is a language code defined in RFC 3066 [17].

ENSEMBLE

```
public static final String ENSEMBLE = "ensemble"
```

An ensemble where a service belongs. The value of this attribute is an `dmb.si.SIView` containing an `dmb.si.SIObject` of `Ensemble.TYPE` type.

History

Document history		
V1.1.1	August 2009	Publication