ETSI TS 102 371 V1.3.1 (2008-07)

Technical Specification

Digital Audio Broadcasting (DAB); Digital Radio Mondiale (DRM); Transportation and Binary Encoding Specification for Electronic Programme Guide (EPG)

European Broadcasting Union — Union Européenne de Radio-Télévision

EBU·UER

DAB Digital Audio Broadcasting

ETSI

Reference
RTS/JTC-DAB-58

Keywords
audio, broadcasting, DAB, DRM, digital, EPG

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 [3], 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.

Introduction

The present document defines how the EPG data will be transported, compressed and profiled such that a good user experience can be achieved using limited broadcast capacity. Using a combination of EPG profiles it is possible that a range of features could be supported in receivers, including:

- The display of schedules at varying levels of detail for programmes from a range of services.
- The display of schedules, with programmes and events ordered into particular groups.
- Navigation and selection of services and programmes.
- Searching through current and future programme listings.
- Timed recording of individual programmes, or of groups of programmes and themed or similar programming.
- Accurate timed recording of programmes and events using PNum signalling.

1 Scope

The present document defines how the XML schema data model for an Electronic Programme Guide (EPG) (TS 102 818 [1]) should be compressed, profiled and broadcast. Within the present document the term "DAB" is used to refer to the Eureka-147 Digital Audio Broadcasting standard (EN 300 401 [3]) and "DRM" is used to refer to the Digital Radio Mondiale standard (ES 201 980 [6]).

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

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

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

[4] ISO/IEC 10646: "Information technology - Universal Multiple-Octet Coded Character Set (UCS)".

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

[6] ETSI ES 201 980: "Digital Radio Mondiale (DRM); System specification".

[7] ETSI TS 101 968: "Digital Radio Mondiale (DRM); Data applications directory".

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 and abbreviations

3.1 Definitions

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

Conditional Access (CA): mechanism by which the user access to service components can be restricted

data service: service which comprises a non-audio primary service component and optionally secondary service components

entity reference: a group of characters used in text strings as a substitute for a single specific character, e.g. &amp

ensemble: transmitted signal, comprising a set of regularly and closely-spaced orthogonal carriers

NOTE: The ensemble is the entity that is received and processed. In general, it contains audio and data services.

Ensemble Identifier (EId): unique 16-bit code, allocated to an ensemble and intended to allow unambiguous worldwide identification of that ensemble

Extended Programme Associated Data (X-PAD): extended part of the PAD carried towards the end of the DAB audio frame, immediately before the Scale Factor Cyclic Redundancy Check (CRC), its length is variable

Programme Associated Data (PAD): information that is related to the audio data in terms of contents and synchronization

NOTE: The PAD field is located at the end of the DAB audio frame.

secondary service component: in the case where a service contains more than the primary service component, the additional service components are secondary service components

service: in TS 102 371 the term "service" is used to refer to a "radio station" such as BBC Radio 4 or Oneword. In strict DAB terms this is actually a service component of a service

service component: part of a service which carries either audio (including PAD) or data

NOTE: The service components of a given service are linked together by the Multiplex Configuration Information. Each service component is carried either in a sub-channel or in the Fast Information Data Channel.

Service Identifier (SId): 16-bit, 24-bit or 32-bit code used to identify a particular service

3.2 Abbreviations

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

CA	Conditional Access
CRC	Cyclic Redundancy Check
CRID	Content Reference ID
DAB	Digital Audio Broadcasting
DRM	Digital Radio Mondiale
ECC	Extended Country Code
EId	Ensemble Identifier

EPG	Electronic Programme Guide
FAC	Fast Access Channel (DRM Multiplex Component)
GI	Group Information
GZIP	GnuZIP
ISO	International Organization for Standardization
LTO	Local Time Offset
MIME	Multipurpose Internet Mail Extensions
MJD	Modified Julian Date
MOT	Multimedia Object Transfer
PAD	Programme Associated Data
PI	Programme Information
PNum	Programme Number
Rfa	Reserved for future addition
Rfu	Reserved for future use
SCIdS	Service Component Identifier within the Service
SI	Service Information
SId	Service Identifier
URL	Uniform Resource Location
UTC	Co-ordinated Universal Time
XML	eXtensible Markup Language
X-PAD	eXtended - Programme Associated Data

4 Encoding

There will be no transmission of the raw EPG XML data (TS 102 818 [1]). The XML specification should still be used to construct the valid information; this binary specification is then used to encode this information in a compact binary form.

The binary encoding described here uses a tag-length-value encoding. Each element or attribute is encoded using a unique tag value, a length value (indicating the length of the data contained within this element or attribute) and the actual data value(s). This enables receivers to easily skip elements that are not wanted or were undefined.

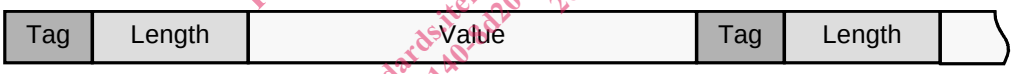

Figure 1: Tag-length-value encoding scheme

XML elements are all encoded in these binary structures as described in clause 4.2. Attributes are coded in a similar way (see clause 4.4). The hierarchical nature of the EPG XML is generally preserved in these binary structures, but the structure is not necessarily identical. Various common data types have been assigned efficient binary encodings as described in clause 4.7. For an example of a binary encoded XML object, see annex C.

Note that although the length of certain data types can be worked out from their encoding, there shall still be a length field in the attribute encoding (see clause 4.4).

4.1 Syntax specification

The specifications of syntax that appear in the present document are written using a form of pseudo-code that is similar to the procedural language "C"; this provides for easy specification of loops and conditional data structures. Within these specifications, the type of individual data fields is expressed using the mnemonics given in table 1.

Table 1: Data type mnemonics for syntax specification

Mnemonic	Description
Uimsbf	Unsigned integer, most significant bit first

4.2 Binary objects

The basic binary objects defined by the present document are defined in table 2. Each binary object carries a single top level element and shall be carried within a single MOT object.

Table 2: Structure of a binary object

Syntax	Size	Type
`binary_object() {` `  top_level_element()`		

top_level_element: A top level element as defined in clause 4.3.1.

4.3 Elements

All elements use basically the same encoding, as defined here.

Table 3: Structure of an element

Syntax	Size	Type
`element() {`		
`  element_tag`	8 bits	uimsbf
`  element_length`	8 bits	uimsbf
`  If (element_length == 0xFE) {`		
`  extended_element_length`	16 bits	uimsbf
`  }`		
`  If (element_length == 0xFF) {`		
`  extended_element_length`	24 bits	uimsbf
`  }`		
`  for (i=0; I<element_length or extended_element_length; i++) {`		
`  element_data_byte`	8 bits	uimsbf
`  }`		
`}`		

element_tag: This byte identifies the element. The tag uniquely identifies the element - i.e. there is a one to one mapping between a tag and an element. If new elements are required in the future then they will use new tag values. The possible values are defined in annex D. Elements with tags that are not defined here are reserved for future use; the tags and their associated content shall not be processed by receivers.

element_length: This field indicates the number of data bytes contained in this element, i.e. the number of bytes that follow the length byte up to the end of the element. The range of this is `0x00` to `0xFD` (i.e. 0 to 253). If this value is either `0xFE` or `0xFF` then the additional *extended_element_length* field defines the element length.

extended_element_length: When used, this field indicates the number of data bytes contained in this element, i.e. the number of bytes that follow the last extended length byte up to the end of the element.

element_data_byte: These bytes contain the element's attributes, CDATA (i.e. string data) and child elements. They shall be encoded in the following order:

1) Attributes.
2) Child elements.
3) CDATA content.

4.3.1 Top-level elements

There are two top-level elements defined in the present document; *epg* and *serviceInformation*. A top-level element shall be carried within a binary object (see clause 4.2) and it shall be the only element (apart from its nested children) in that object. The possible values of the *element_tag* for top-level elements are defined in table 4. Top-level elements with tags that are not defined here are reserved for future use; these tags and their associated content shall not be processed by receivers.

Table 4: Top-level element tags

Element	Tag
epg	0x02
serviceInformation	0x03

As well as the appropriate elements defined by the EPG XML specification the top-level elements may also, optionally, contain a string token table (see clause 4.9), a default contentID (see clause 4.10) and a default language (see clause 4.11). If present, these elements shall be the first elements to occur in the top-level element after the attributes.

A top-level element is encoded in the same way as a normal element (see clause 4.3) with the exception that the *element_data_bytes* shall be encoded in the following order:

1) Attributes.
2) String token table (if present).
3) Default contentID (if present).
4) Default language (if present).
5) Child elements.
6) CDATA content.

4.4 Attributes

All attributes use basically the same encoding, as defined here.

Table 5: Structure of an attribute

Syntax	Size	Type
`attribute() {`		
`    attribute_tag`	8 bits	uimsbf
`    attribute_length`	8 bits	uimsbf
`    if (attribute_length == 0xFE) {`		
`        extended_attribute_length`	16 bits	uimsbf
`    }`		
`    if (attribute_length == 0xFF) {`		
`        extended_attribute_length`	24 bits	uimsbf
`    }`		
`    for (i=0; i<attribute_length or extended_attribute_length; i++) {`		
`        attribute_data_byte`	8 bits	uimsbf
`    }`		
`}`		

attribute_tag: This byte uniquely identifies the attribute **within the parent element**. The possible values are defined in annex E.

Attributes with tags that are not defined here are reserved for future use and should not be processed by receivers.

attribute_length: This field indicates the number of data bytes contained in this attribute, i.e. the number of bytes that follow the length byte up to the end of the attribute. The range of this is `0x00` to `0xFD` (i.e. 0 to 253). If this value is either `0xFE` or `0xFF` then the additional *extended_attribute_length* field defines the attribute length.

extended_attribute_length: When used, this field indicates the number of data bytes contained in this attribute, i.e. the number of bytes that follow the last extended length byte up to the end of the attribute.

attribute_data_byte: These bytes contain either a string (clause 4.5.1) or an enumerated data value (see clause 4.6) or a common data type (clause 4.7).

NOTE: Any entity references should be expanded.

4.4.1 Default attributes

Where an attribute has a default value there is no need for it to be present in the binary encoding as the receiver shall always automatically use the default value.

4.5 CDATA and strings

All CDATA or text strings apart from textual attributes shall be encoded as detailed in table 6.

Table 6: Structure of a CDATA element

Syntax	Size	Type
`cdata() {`		
`    cdata_tag`	8 bits	uimsbf
`    cdata_length`	8 bits	uimsbf
`    If (cdata_length == 0xFE) {`		
`        extended_cdata_length`	16 bits	uimsbf
`    }`		
`    If (cdata_length == 0xFF) {`		
`        extended_cdata_length`	24 bits	uimsbf
`    }`		
`    for (i=0; i<cdata_length or extended_cdata_length; i++) {`		
`        cdata_data_byte`	8 bits	uimsbf
`    }`		
`}`		

cdata_tag: This shall always be `0x01`.

cdata_length: This field indicates the number of data bytes contained in this string. The range of this is `0x00` to `0xFD` (i.e. 0 to 253). If this value is either `0xFE` or `0xFF` then the additional *extended_cdata_length* field defines the attribute length.

extended_cdata_length: When used, this field indicates the number of data bytes contained in this string.

cdata_data_byte: These bytes contain the characters for this CDATA element.

NOTE 1: Attributes with text values are encoded as an attribute (see clause 4.4) with the *attribute_data_bytes* being the character data bytes only.

NOTE 2: Any entity references should be expanded.

4.5.1 Character sets

All CDATA and other strings shall use ISO/IEC 10646 [4] with UTF-8 encoding. The ISO/IEC 10646 [4] characters `0xE000` to `0xF8FF` shall not be included within any binary encoded strings.