# ETSI EN 303 560 V1.1.1 (2018-05)


**Digital Video Broadcasting (DVB);**
**TTML subtitling systems**

EBU
OPERATING EUROVISION

DVB

Reference
DEN/JTC-DVB-375

Keywords
broadcast, digital, DVB, subtitle, TV

***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***

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (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
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

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

***Copyright Notification***

# Contents

# Intellectual Property Rights

## Essential patents

IPRs essential or potentially essential to normative deliverables 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 (https://ipr.etsi.org/).

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.

## Trademarks

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

# Foreword

This European Standard (EN) 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: 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 Digital Video Broadcasting Project (DVB) is an industry-led consortium of broadcasters, manufacturers, network operators, software developers, regulatory bodies, content owners and others committed to designing global standards for the delivery of digital television and data services. DVB fosters market driven solutions that meet the needs and economic circumstances of broadcast industry stakeholders and consumers. DVB standards cover all aspects of digital television from transmission through interfacing, conditional access and interactivity for digital video, audio and data. The consortium came together in 1993 to provide global standardization, interoperability and future proof specifications.

| National transposition dates | |
|---|---|
| Date of adoption of this EN: | 30 April 2018 |
| Date of latest announcement of this EN (doa): | 31 July 2018 |
| Date of latest publication of new National Standard or endorsement of this EN (dop/e): | 31 January 2019 |
| Date of withdrawal of any conflicting National Standard (dow): | 31 January 2019 |

# Modal verbs terminology

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

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

# Introduction

Today's broadcast content can be distributed in many forms and via many different paths. Whilst broadcast services traditionally relied on bitmap-based subtitles because of processing simplicity, advanced processing and text rendering have made text-based approaches feasible. These offer more flexibility and better options to improve the user experience. TTML (Timed Text Markup Language) is established as a common solution for IP-based platforms, but multiple profiles exist.

In Europe, the EBU Group 'Subtitles in XML' published the EBU-TT-D TTML profile [20] which was adopted by the DVB DASH specification as defined in ETSI TS 103 285 [6] and by HbbTV® as defined in ETSI TS 102 79 [5]. The W3C later published the IMSC1 TTML profiles [4], which were adopted by ATSC.

There are only minor differences between EBU-TT-D [3], [20] and IMSC1 Text Profile.

The present specification builds on the existing widespread device support for EBU-TT-D [20], in particular through support for HbbTV®, to enable the distribution of TTML subtitles together with audio/video content via broadcast.

Clause 4 specifies TTML subtitle constraints for a default conformance point, to be supported by both EBU-TT-D [3], [20] and IMSC1 Text Profile compatible processors.

Clause 5 specifies subtitle delivery, including PSI/SI signalling, TS packetisation, TTML segmentation and synchronization requirements.

Finally clause 6 specifies IRD requirements.

# 1 Scope

The present document specifies the transport of TTML [2] **subtitle stream**s in DVB MPEG-2 **transport stream**s, based on the MPEG-2 system described in ISO/IEC 13818-1 [1]. TTML is an XML-based representation. The present document provides syntax for delivery of TTML **subtitle stream**s over MPEG-2 **transport stream**, and is based on EBU-TT-D [3] compatible with the IMSC1 [4] Text Profile of W3C TTML [2].

# 2 References

## 2.1 Normative references

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

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

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

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

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

[2] W3C Recommendation (TTML): "Timed Text Markup Language 1 (TTML1) (Second Edition)".

NOTE: Available at http://www.w3.org/TR/2013/REC-ttml1-20130924/.

[3] EBU Tech 3380 "EBU-TT-D Subtitling Distribution Format", version 1.0.1.

NOTE: Available at https://tech.ebu.ch/publications/tech3380.

[4] W3C Recommendation (IMSC1): "TTML Profiles for Internet Media Subtitles and Captions 1.0.1 (IMSC1)".

NOTE: Available at http://www.w3.org/TR/ttml-imsc1.0.1/.

[5] ETSI TS 102 796: "Hybrid Broadcast Broadband TV".

[6] ETSI TS 103 285: "Digital Video Broadcasting (DVB); MPEG-DASH Profile for Transport of ISO BMFF Based DVB Services over IP Based Networks".

[7] DVB BlueBook A038: "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".

NOTE: Available at http://www.dvb.org/resources/public/standards/a038_dvb_spec_december_2017.pdf.

[8] ISO 639-2: "Codes for the representation of names of languages -- Part 2: Alpha-3 code".

[9] ISO/IEC 8859-1: "Information technology -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1".

[10] W3C Working Group Note: "TTML Media Type Definition and Profile Registry".

NOTE: Available at http://www.w3.org/TR/ttml-profile-registry/.

[11] ETSI EN 301 192: "Digital Video Broadcasting (DVB); DVB specification for data broadcasting".

[12] ETSI TS 102 809: "Digital Video Broadcasting (DVB); Signalling and carriage of interactive applications and services in hybrid broadcast/broadband environments".

[13] ETSI TS 102 851: "Digital Video Broadcasting (DVB); Uniform Resource Identifiers (URI) for DVB Systems".

[14] IETF RFC 1952: "GZIP file format specification version 4.3".

[15] DVB BlueBook A126: "Digital Video Broadcasting (DVB); Allocation of identifiers and codes for Digital Video Broadcasting (DVB) systems".

NOTE: Available at https://www.dvb.org/resources/public/standards/a126_allocation_identifiers.pdf.

[16] W3C Recommendation (XML): "Extensible Markup Language (XML) 1.0 (Fifth Edition)".

NOTE: Available at https://www.w3.org/TR/2008/REC-xml-20081126/.

[17] EBU Tech 3381: "Carriage of EBU-TT-D in ISOBMFF", version 1.0.

NOTE: Available at https://tech.ebu.ch/publications/tech3381.

[18] W3C Recommendation 13 December 2012 (WOFF): "Web Open Font Format (WOFF) 1.0".

NOTE: Available at http://www.w3.org/TR/2012/REC-WOFF-20121213/.

[19] ISO/IEC 14496-22:2015: "Information technology -- Coding of audio-visual objects -- Part 22: Open Font Format".

[20] EBU Tech 3380: "EBU-TT-D Subtitling Distribution Format", version 1.0.

NOTE: Available at https://tech.ebu.ch/publications/tech3380.

[21] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

## 2.2 Informative references

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

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

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

[i.1] W3C Recommendation (WCAG): "Web Content Accessibility Guidelines (WCAG) 2.0".

NOTE: Available at https://www.w3.org/TR/2008/REC-WCAG20-20081211/.

[i.2] W3C Candidate Recommendation (CSS): "CSS Fonts Module Level 3: Font matching algorithm".

NOTE: Available at https://www.w3.org/TR/css-fonts-3/#font-matching-algorithm.

[i.3] IEC 61966-2-1:1999: "Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB".

# 3 Definitions and abbreviations

## 3.1 Definitions

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

**Intermediate Synchronic Document (ISD):** temporally bounded (possibly empty) subset of TTML content during which no elements change state between active and inactive

NOTE: A TTML document can conceptually be considered as a contiguous sequence of ISDs.

**maximum period of activation (MPA):** maximum duration a single **TTML segment** can be active, referred to as $T_{MPA}$

**Packet IDentifier (PID):** transport stream packet identifier

NOTE: See ISO/IEC 13818-1 [1].

**PES packet:** See ISO/IEC 13818-1 [1].

**Presentation Time Stamp (PTS):** See ISO/IEC 13818-1 [1].

**reserved_zero_future_use:** when used in the clause defining the coded bit stream, indicates that the value may be used in the future for ETSI defined extensions

NOTE: All "reserved_zero_future_use" bits are set to "0".

**subtitle segment:** basic syntactical element of a **subtitle stream**

NOTE: The present document is structured to allow other types of subtitle segments to be defined in the future. In this version of the specification, all **subtitle segment**s are **TTML segments**.

**subtitle service properties:** properties defined by TTML subtitling descriptor which include but are not limited to language, purpose, TTS_suitability, font usage, and qualifier

**subtitle service:** service that provides subtitling for an MPEG program (DVB service) with a set of **subtitle service properties**

**subtitle service qualifier variants:** set of **subtitle service**s whose **subtitle service properties** are identical except for the qualifier_present_flag and qualifier fields in the TTML subtitling descriptor

**subtitle stream:** stream of **subtitle segment**s carried in **transport stream packet**s identified by the same **PID**, containing a single **subtitle service**

**transport stream packet:** See ISO/IEC 13818-1 [1].

**transport stream:** stream of **transport stream packet**s carrying one or more MPEG programs

NOTE: See ISO/IEC 13818-1 [1].

**TTML document chunk:** self-contained and temporally bounded subset of a TTML document [2]

NOTE: Self-contained in this context means that the document chunk contains all elements to render this subset except the actual fonts used.

**TTML segment: subtitle segment** whose payload is an XML serialization of a TTML document chunk, concretely encoded as a well-formed XML 1.0 [16] document

## 3.2 Abbreviations

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

| | |
|---|---|
| ATSC | Advanced Television Systems Committee |
| BAT | Bouquet Association Table |
| bslbf | bit string, left bit first |

NOTE: The rightmost bit of a bslbf field is always referred to as $b_0$. This will be the last bit of the field transmitted.

| | |
|---|---|
| CRC | Cyclic Redundancy Check |
| CSS | Cascading Style Sheets |
| DASH | Dynamic Adaptive Streaming over HTTP |
| DFIS | Downloadable Font Information Section |
| DFIST | Downloadable Font Information sub_table |
| DFIT | Downloadable Font Information Table |
| DSI | Download Server Initiate |
| DSM-CC | Digital Storage Media - Command and Control |
| DVB | Digital Video Broadcasting |
| DVB URI | Uniform Resource Identifiers for DVB systems |

NOTE: As defined in ETSI TS 102 851 [13].

EIT Event Information Table

NOTE: As defined in DVB BlueBook A038 [7].

HbbTV® Hybrid broadcast broadband TV

NOTE: As defined in ETSI TS 102 796 [5].

| | |
|---|---|
| HDR | High Dynamic Range |
| HRM | Hypothetical Render Model |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol - Secure |
| IP | Internet Protocol |
| IRD | Integrated Receiver Decoder |
| ISD | Intermediate Synchronic Document |
| MPA | Maximum Period of Activation |
| MPEG | Moving Pictures Experts Group |
| NIT | Network Information Table |

NOTE: As defined in ISO/IEC 13818-1 [1].

PES Packetized Elementary Stream

NOTE: As defined in ISO/IEC 13818-1 [1].

PID transport stream Packet IDentifier

NOTE: As defined in ISO/IEC 13818-1 [1].

PMT Program Map Table

NOTE: As defined in ISO/IEC 13818-1 [1].

| | |
|---|---|
| PSI | Program Specific Information |
| PTS | Presentation Time Stamp |

NOTE: As defined in ISO/IEC 13818-1 [1].

RGB Red Green Blue

SDT Service Description Table

NOTE: As defined in DVB BlueBook A038 [7].

SI Service Information

NOTE: As defined in ISO/IEC 13818-1 [1].

sRGB standard Red Green Blue

NOTE: As defined in IEC as IEC 61966-2-1:1999 [i.3].

TLS Transport Layer Security
$T_{MPA}$ Time period equal to the maximum period of activation
TS Transport Stream

NOTE: As defined in ISO/IEC 13818-1 [1].

TTML Timed Text Markup Language

NOTE: As defined in W3C "Timed Text Markup Language 1 (TTML1)" [2].

TTS Text-To-Speech
uimsbf unsigned integer, most significant bit first

NOTE: When a uimsbf field is broken down into bit fields, the least significant bit is always referred to as $b_0$. This will be the last bit of the field transmitted.

URI Uniform Resource Identifiers

NOTE: Generic form of DVB URI as defined in IETF RFC 3986 [21].

UTF Unicode Transformation Format
UTF-8 Unicode (or Universal Coded Character Set) Transformation Format - 8-bit
WOFF Web Open Font Format

NOTE: As defined in W3C Web Open Font Format (WOFF) 1.0 [18].

# 4 Default DVB TTML subtitle conformance point

## 4.1 Introduction

In order to maximize interoperability DVB has chosen a default DVB TTML document conformance point that can be processed by either of two commonly used TTML processor profiles, EBU-TT-D [3] and IMSC1 [4] Text Profile.

## 4.2 Document Constraints

### 4.2.1 General

A DVB TTML subtitle stream conformant to the default conformance point shall be authored such that it can be processed by an IRD supporting any of the following DVB TTML processor profile combinations:

- only EBU-TT-D [3], configured to meet the additional IRD requirements for the default conformance point in clause 6.6;
- only IMSC1 [4] Text Profile;
- both EBU-TT-D [3] and IMSC1 [4] Text Profile.

The TTML profile identifier for the default conformance point from the TTML Profiles Registry [10] is described in clause 5.2.1.1 as "etd1†|im1t", indicating that either a suitably constrained EBU-TT-D processor (etd1†) or an IMSC1 Text Profile processor (im1t) may be used.

The relationship between IMSC1 Text Profile and EBU-TT-D is explored in IMSC1 [4], annex I.2.

Certain differences between the EBU-TT-D and IMSC1 Text Profile TTML profiles are addressed by applying additional constraints to IRDs, as opposed to further constraining TTML documents. These constraints apply to EBU-TT-D processors compliant with the default conformance point:

- support IMSC1 generic font family name mapping, see clause 6.4;
- support IMSC1's default font colour of white, in the absence of any colour signalled, see clause 6.6;
- support itts:fillLineGap, ebutts:linePadding and ebutts:multiRowAlign style attributes, see clause 6.6;
- tolerate attributes in foreign namespaces, see clause 6.6;
- ignore unrecognised attributes, see clause 6.6.

The remaining differences between the EBU-TT-D and IMSC1 Text Profile TTML profiles are addressed through the TTML document constraints specified in the following clauses. These constraints apply to TTML documents compliant with the default conformance point.

### 4.2.2 Regions

In order to maintain IMSC1 conformance, a TTML subtitle stream shall have no more than four regions active at the same time.

### 4.2.3 Document complexity

In order to maintain IMSC1 conformance, a TTML subtitle stream shall meet the requirements of the IMSC1 [4] Hypothetical Render Model (HRM).

### 4.2.4 Document encoding

In order to maintain IMSC1 conformance, a TTML subtitle stream shall use UTF-8 encoding.

### 4.2.5 Namespaces

Documents compliant to the default conformance point shall only use namespaces as permitted in EBU-TT-D [3].

NOTE: Foreign namespaces are permitted within metadata elements because EBU-TT-D allows this.

### 4.2.6 Guidance for document authors and content providers (informative)

#### 4.2.6.1 Overview

If a TTML document is authored to meet the default conformance points' document constraints with the intent of being presented using a general purpose EBU-TT-D processor the following considerations may cause unexpected presentation.

#### 4.2.6.2 Font mapping for generic font family names

An EBU-TT-D processor configured to meet the default conformance point (see clause 6.6) could map a generic font family name to a different font than a generic, EBU-TT-D processor would do, or an EBU-TT-D processor constrained by other applicable specifications such as HbbTV® as defined in ETSI TS 102 796 [5]. For example the "proportionalSansSerif" generic font family could be mapped to the Arial font according to the default conformance point, and the "default" generic font to Courier New, but an HbbTV® EBU-TT-D implementation could map both to the Tiresias™ font.

If the document contents were authored with an equivalent to Arial as the intended presentation font and a font such as Tiresias™ is used, which generally produces text that occupies more width, unexpected line breaks or overflows could be introduced.

NOTE: Tiresias™ is the trade name of a product supplied by Monotype. This information is given for the convenience of users of the present document and does not constitute an endorsement by ETSI of the product named. Equivalent products may be used if they can be shown to lead to the same results.

A possible workaround for unexpected line breaks is to author against the widest font that might be used for presentation purposes. Alternatively the TTML profile signalling can be set to exclude use of processors that could produce unexpected presentation.

#### 4.2.6.3 Default text colour

Whilst generic TTML and EBU-TT-D processors are not required to set a default text colour, an IMSC1 Text profile processor is required to adopt a default text colour of "white". It is therefore preferable to avoid doubt where consistent presentation is important by specifying the colour of text explicitly within documents.

#### 4.2.6.4 Requirement for support of optional features

An EBU-TT-D processor configured to meet the default conformance point (see clause 6.6) is required to support the EBU-TT-D style attributes ebutts:multiRowAlign and ebutts:linePadding; a generic EBU-TT-D processor may not support those attributes. This could result in unexpected positioning of subtitle text if those features are used and the processor profile signalling permits rendering by a generic EBU-TT-D processor.

#### 4.2.6.5 Specifying active area

IMSC1 [4] and EBU-TT-D [3] authors are recommended to use the ittp:activeArea attribute to indicate the area within the root container that contains content that is intended to always be visible.

# 5 Delivery

## 5.1 IP delivery

The present document does not support delivery of subtitles over the Internet.

NOTE: The HbbTV® specification as defined in ETSI TS 102 796 [5] provides a mechanism for synchronizing EBU-TT-D [20] subtitles with broadcast video and audio. In that specification, EBU-TT-D [20] subtitles can be delivered via MPEG-DASH segments using the DVB profile as defined in ETSI TS 103 285 [6] or via a HTTP file download.

## 5.2 Broadcast TS delivery

### 5.2.1 PSI/SI signalling

#### 5.2.1.1 TTML subtitling descriptor

The TTML subtitling descriptor presence in the ISO/IEC 13818-1 [1] PMT indicates that a **PID** contains a **subtitle stream** conveyed using the **PES** structure defined in clause 5.2.2.2. The value of the stream_type in the PMT shall be "0x06" indicating **PES** carrying private data.

This descriptor provides the following **subtitle service properties:**

- The ISO_639_language_code indicates the language of the **subtitle service**.
- The subtitle_purpose field indicates the intended purpose of the **subtitle service**.

- The TTS_suitability field indicates the suitability of the subtitles for conversion using a Text-To-Speech (TTS) processor.
- The essential_font_usage_flag indicates if one or more downloadable fonts are required to present this **subtitle service**.

Finally, the optional qualifier field provides additional informational presentation properties. It is defined below in clause 5.2.1.3.

The TTML_subtitling_descriptor shall be coded according to table 1.

**Table 1: TTML subtitling descriptor**

| Syntax | Number of bits | Identifier |
|---|---|---|
| TTML_subtitling_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| ISO_639_language_code | 24 | bslbf |
| subtitle_purpose | 6 | uimsbf |
| TTS_suitability | 2 | uimsbf |
| essential_font_usage_flag | 1 | bslbf |
| qualifier_present_flag | 1 | bslbf |
| reserved_zero_future_use | 2 | bslbf |
| dvb_ttml_profile_count | 4 | uimsbf |
| for(i=0;i<N;i++) { | | |
| dvb_ttml_profile | 8 | uimbsf |
| } | | |
| if (qualifier_present_flag == 1){ | | |
| qualifier | 32 | bslbf |
| } | | |
| if (essential_font_usage_flag == 1){ | | |
| font_count | 8 | uimsbf |
| for(i=0; i<font_count; i++){ | | |
| reserved_zero_future_use | 1 | bslbf |
| font_id | 7 | uimsbf |
| } | | |
| } | | |
| text_length | 8 | bslbf |
| for(i=0;i<N;i++) { | | |
| text_char | 8 | bslbf |
| } | | |
| for(i=0;i<N;i++) { | | |
| reserved_zero_future_use | 8 | bslbf |
| } | | |
| } | | |

**Semantics for the TTML subtitling descriptor:**

**descriptor_tag:** This 8-bit field shall have the value "0x7F" to indicate an extension_descriptor in accordance with DVB BlueBook A038 [7].

**descriptor_tag_extension:** This 8-bit field shall have the value "0x20" in accordance with DVB BlueBook A038 [7].

**ISO_639_language_code:** This 24-bit field contains the ISO 639-2 [8] three character language code of the language of the subtitles. Both ISO 639-2 [8]/B and ISO 639-2 [8]/T may be used. Each character is coded into 8-bits according to ISO/IEC 8859-1 [9] and inserted in order into the 24-bit field.

EXAMPLE**:** French has 3-character code "fre", which is coded as:
"0110 0110 0111 0010 0110 0101".

**subtitle_purpose:** This 6-bit field signals the purpose of the subtitle service. Depending on the purpose, the **subtitle stream** may include one or more features, as defined in table 2.

**Table 2: Subtitle purpose**

| subtitle_purpose | Short name | Hard-of-hearing | Translation dialogue | Non-translation dialogue | Audio description | Content-related commentary |
|---|---|---|---|---|---|---|
| 0x00 | same-lang-dialogue | No | No | Yes | No | No |
| 0x01 | other-lang-dialogue | No | Yes | No | No | No |
| 0x02 | all-dialogue | No | Yes | Yes | No | No |
| 0x03-0x0F | reserved for future use | | | | | |
| 0x10 | hard-of-hearing | Yes | No | Yes | No | No |
| 0x11 | other-lang-dialogue-with-hard-of-hearing | Yes | Yes | No | No | No |
| 0x12 | all-dialogue-with-hard-of-hearing | Yes | Yes | Yes | No | No |
| 0x13-0x2F | reserved for future use | | | | | |
| 0x30 | audio-description | No | No | No | Yes | No |
| 0x31 | content-related-commentary | No | No | No | No | Yes |
| 0x32-0x3F | reserved for future use | | | | | |

The meaning of each of these features is as follows:

- **Hard-of-hearing subtitle stream** includes descriptions of non-dialogue sounds. (e.g. gun fire, explosion, lions roar.)
- **Translation dialogue subtitle stream** includes a translation of foreign language dialogue.
- **Non-translation dialogue subtitle stream** includes a transcription of same language dialogue.
- **Audio description subtitle stream** includes description of the visual scene. (e.g. "a lion lies in the sun.")
- **Content-related commentary subtitle stream** includes commentary related information. (e.g. director's commentary.)

**TTS_suitability:** This 2-bit field indicates whether the subtitles are suitable for text-to-speech (TTS) as described in table 3.

**Table 3: TTS suitability**

| TTS_suitability | Meaning |
|---|---|
| 0x0 | unknown suitability for TTS |
| 0x1 | suitable for TTS |
| 0x2 | not suitable for TTS |
| 0x3 | reserved for future use |

Only some subtitle streams are suitable for text-to-speech conversion and use as spoken subtitles. The typical use of spoken subtitles is to provide an audible translation of "other language" dialogue into the viewer's preferred language, so that viewers who are perhaps unable to read the presented subtitle text are nevertheless able to follow the programme content. In addition, not all subtitles that are suitable for text-to-speech are also suitable for use as spoken subtitles; for example, they could be intended for use as alternate sources of audio description. Table 4 illustrates the different combinations.

**Table 4: Example combinations of subtitle_purpose and TTS_suitability**

| | | TTS_suitability | | |
|---|---|---|---|---|
| | | **Suitable for TTS** | **Unknown suitability for TTS** | **Not suitable for TTS** |
| **Subtitle_purpose** | **Dialogue (e.g. subtitle_purpose 0x00-0x02, 0x10-0x12)** | Suitable for Spoken Subtitles | Possibly suitable for Spoken Subtitles | Not suitable for Spoken Subtitles |
| | **Hard-of-hearing (e.g. subtitle_purpose 0x10-0x12)** | Suitable for alternative audio source for hearing impaired | Possibly suitable for alternative audio source for hearing impaired | Not suitable as alternative audio source for hearing impaired |
| | **Audio description (e.g. subtitle_purpose 0x30)** | Suitable for audio description source | Invalid combination | Invalid combination |
| | **Content-related commentary (e.g. subtitle_purpose 0x31)** | Suitable for Spoken Commentary | Possibly suitable for Spoken Commentary | Not suitable for Spoken Commentary |

**essential_font_usage_flag:** This 1-bit flag when set to "1" indicates that one or more downloadable fonts shall be used to display these subtitles. When set to "0" it indicates that no downloadable fonts are required to display these subtitles. In this case, one or more optional fonts may be available for download.

**qualifier_present_flag:** This 1-bit field indicates the presence of a presentation properties field named "qualifier". A qualifier_present_flag set to "0" indicates default broadcaster presentation properties apply, and is equivalent to a qualifier_present_flag set to "1" combined with a qualifier field with all values set to "0".

**dvb_ttml_profile_count:** This 4-bit field specifies the number of dvb_ttml_profile fields which immediately follow. This field shall not be set to zero.

A dvb_ttml_profile_count value of one or more indicates that every TTML segment sent on the respective **PID** can be processed by an IRD supporting any of the signalled DVB TTML processor profiles. If more than one profile is signalled, the order has no significance.

**dvb_ttml_profile:** This 8-bit field specifies a DVB TTML processor profile capable of processing every TTML segment sent on the respective **PID**. The possible values of dvb_ttml_profile are specified in table 5. Examples of TTML processor profile signalling are described in clause 5.2.1.2.

**Table 5: DVB TTML processor profile**

| dvb_ttml_profile | TTML processor profile identifier (from TTML registryTTML registry [10]) | Comment |
|---|---|---|
| 0x00 | etd1†\|im1t | Default conformance point. Requires EBU-TT-D [3] processor compliant with the additional constraints(†) of the default conformance point, defined in clause 6.6, or an IMSC1 Text Profile processor |
| 0x01 | im1t | IMSC1 [4] Text Profile |
| 0x02 | etd1 | EBU-TT-D [3] (without necessarily supporting the constraints of the default conformance point defined in clause 6.6) |
| 0x03-0xFF | | reserved for future use |

The use of the "|" operator in the table above is derived from the TTML Media Type Definition and Profile Registry [10] and means that either of the operands may be selected as a suitable processor.

The use of the "†" in the table above is intended to highlight that extra constraints are defined for the EBU-TT-D processor in the context of the default conformance point; this syntax is not present in the TTML Media Type Definition and Profile Registry.

Support for TTML profiles other than the default conformance point defined in clause 4 is optional for IRDs (see clause 6). If the TTML subtitle service intends to be interoperable with all IRDs compliant with this specification, DVB TTML subtitles shall conform to the default conformance point (clause 4).

Document conformance with the signalled processor profile(s) is relaxed for the empty TTML document, as defined in clause 5.2.3.5.

**qualifier:** This optional 32-bit field is intended to signal additional presentation characteristics to allow further selection amongst **subtitle stream**s, see clause 5.2.1.3.

**font_count:** The number of essential fonts (identified by font_id) required to process this **subtitle stream**. If the essential_font_flag is not equal to 0, the value of this field shall not be set to zero.

**font_id:** This 7-bit field provides a unique identifier to a downloadable font within the downloadable font information table (DFIT) that is in-scope (see clause 5.3.2.2).

**text_length:** This 8-bit field specifies the length in bytes of the following text describing the **subtitle service.**

**text_char:** This is an 8-bit field. A string of "char" fields specifies the text description of the **subtitle service**.

Text information is coded using the character sets and methods described in Annex A of DVB BlueBook A038 [7].

**reserved_zero_future_use:** These bits are reserved for future use, and they shall be set to value of "0".

### 5.2.1.2 Example TTML profile signalling

#### 5.2.1.2.1 Signalling conformance only to the default conformance point

Table 6 shows the example of signalling for default TTML conformance point. This indicates to the IRD that the subtitle_stream can be decoded by a EBU-TT-D subtitle processor, compliant with the additional constraints of the default conformance point (see clause 4.2), or a IMSC1 Text Profile subtitle processor.

**Table 6: Example of signalling for default TTML conformance point**

| Loop count i | Field name | Value |
|---|---|---|
| n/a | dvb_ttml_profile_count | 0x1 |
| 0 | dvb_ttml_profile | 0x00 (Default conformance point) |

#### 5.2.1.2.2 Signalling conformance to only EBU-TT-D

Table 7 shows the example of signalling to indicate to an IRD that the subtitle_stream can only be decoded by an EBU-TT-D [3] subtitle processor because the subtitle stream does not apply the additional constraints required by the default conformance point (see clause 4.2).

**Table 7: Example of signalling for EBU-TT-D**

| Loop count i | Field name | Value |
|---|---|---|
| n/a | dvb_ttml_profile_count | 0x1 |
| 0 | dvb_ttml_profile | 0x02 (EBU-TT-D [3]) |

#### 5.2.1.2.3 Signalling conformance with EBU-TT-D and the default conformance point

Table 8 shows the example of signalling to indicate to an IRD that the subtitle_stream can only be decoded by an EBU-TT-D [3] subtitle processor with or without the additional constraints of the default conformance point (see clause 4.2), or a IMSC1 Text Profile subtitle processor.

**Table 8: Example of signalling for EBU-TT-D and the default conformance point**

| Loop count i | Field name | Value |
|---|---|---|
| n/a | dvb_ttml_profile_count | 0x2 |
| 0 | dvb_ttml_profile | 0x00 (Default conformance point) |
| 1 | dvb_ttml_profile | 0x02 (EBU-TT-D [3]) |

#### 5.2.1.2.4 Signalling conformance to only IMSC1 Text Profile

Table 9 shows the example of signalling to indicate to an IRD that the subtitle_stream can only be decoded by a IMSC1 [4] Text Profile subtitle processor. This shall be used for subtitle streams using IMSC1 features not allowed by the default conformance point (see clause 4.2).

**Table 9: Example of signalling for IMSC1 text profile**

| Loop count i | Field name | Value |
|---|---|---|
| n/a | dvb_ttml_profile_count | 0x1 |
| 0 | dvb_ttml_profile | 0x01 (IMSC1 [4] Text Profile) |

### 5.2.1.3 Presentation properties in qualifier field

Selection of the subtitle stream by the IRD is based upon the **subtitle service properties** such as language and purpose. The TTML subtitling descriptor may also include the qualifier field as defined in table 10. The qualifier field is intended to signal additional presentation properties (e.g. large fonts) to allow further selection amongst **subtitle service qualifier variants.** If the feature is offered by the IRD and the service, it allows the user to proceed with a more detailed selection. A service may contain multiple sets of **subtitle service qualifier variants.**

Among **subtitle stream**s for which only the qualifier field differs (i.e. **subtitle service qualifier variants**), the default qualifier variant shall be the first of the **subtitle service qualifier variants** signalled in the PMT. Such an order allows IRDs to select the default variant without processing the subtitle qualifier.

A qualifier_present_flag set to "0" indicates the qualifier field is not present, this shall be interpreted as if a qualifier field is present with all of the values are set to "0".

**Table 10: Qualifier**

| Syntax | Number of bits | Identifier |
|---|---|---|
| `qualifier {` | | |
| `  size` | 4 | uimbf |
| `  cadence` | 4 | uimbf |
| `  monochrome_flag` | 1 | bslbf |
| `  enhanced_accessibility_contrast_flag` | 1 | bslbf |
| `  position` | 4 | uimbf |
| `  reserved_zero_future_use` | 26 | bslbf |
| `}` | | |

**Semantics for the qualifier:**

**size:** This 4-bit field indicates the size of the **subtitle stream** according to table 11.

**Table 11: Size**

| Size | Description |
|---|---|
| 0x0 | broadcaster's default size |
| 0x1 | small |
| 0x2 | medium |
| 0x3 | large |
| 0x4 - 0xF | reserved for future use |

**cadence:** This 4-bit field indicates the typical rate of change of subtitles display according to table 12.

**Table 12: Cadence**

| Cadence | Description |
|---|---|
| 0x0 | broadcaster's default cadence |
| 0x1 | slow |
| 0x2 | medium |
| 0x3 | fast |
| 0x4 - 0xF | reserved for future use |

**monochrome_flag:** This 1-bit field indicates whether the **subtitle stream** is monochrome or in colour. If set to "0" the subtitle may use colours. If set to "1" the subtitle shall be monochrome.

**enhanced_accessibility_contrast_flag:** This 1-bit field when set to "1" indicates that the related **subtitle stream** is authored using colour combinations designed to meet accessibility requirements, such as guideline 1.4.3 defined in WCAG 2.0 [i.1].

**position:** This 4-bit field indicates the position of the subtitles according to table 13.

**Table 13: Position**

| Position | Label | Description |
|---|---|---|
| 0x0 | default | broadcaster's default position |
| 0x1 | top | subtitles are placed at the top of the screen |
| 0x2 | bottom | subtitles are placed at the bottom of the screen |
| 0x3 | speaker | close to speaker mouth |
| 0x4 | dynamic | position of each subtitle may vary |
| 0x5 - 0xF | reserved for future use | |

**reserved_zero_future_use:** These bits are reserved for future use, and they shall be set to value of "0".

#### 5.2.1.4 Component descriptors for TTML subtitles

A component descriptor (as defined in DVB BlueBook A038 [7]) can be used in the EIT to indicate DVB events which carry TTML subtitles. Such a component descriptor shall have a stream_content of 0x9, a stream_content_ext of 0x2, and the value of the component_type field shall be coded according to table 14.

**Table 14: Component_type for TTML subtitles**

| Syntax | Number of bits | Identifier |
|---|---|---|
| `TTML_subtitling_component_type() {` | | |
| `  subtitle_purpose` | 6 | uimsbf |
| `  TTS_suitability` | 2 | uimsbf |
| `}` | | |

**Semantics for the TTML_subtitling_component_type:**

**subtitle_purpose:** This field indicates the purpose of the TTML subtitle track. For a given stream, it should be identical to the subtitle_purpose field in the TTML_subtitling_descriptor see clause 5.2.1.1.

**TTS_suitability:** This field indicates the suitability of the TTML subtitle track for TTS. For a given stream, it should be identical to the TTS_suitability field in the TTML_subtitling_descriptor see clause 5.2.1.1.

### 5.2.2 TS packetization

#### 5.2.2.1 PES packet format and carriage in the TS

DVB TTML **subtitle stream**s should be packetized in a way to facilitate fast random access for broadcasting system.

The basic "building block" of the **subtitle stream** is the subtitle segment. These segments are carried in **PES packet**s, which are in turn carried by **transport stream packet**s. The number of subtitle segments carried in a **PES packet** is

only limited by the maximum length of a **PES packet**, as defined by ISO/IEC 13818-1 [1]. However, limitations on the number of TTML segments are specified in clause 5.2.2.2.2.

For carriage of DVB TTML subtitles, the **PES packet** syntax and semantics as defined in ISO/IEC 13818-1 [1] shall be applied within the constraints in table 15.

**Table 15: PES header constraints for carriage of DVB TTML subtitles in a PES**

| Field name | Requirement |
|---|---|
| stream_id | Set to "1011 1101" (0xBD) indicating "private_stream_1". |
| PES_packet_length | Set to a value that specifies the length of the **PES packet**, as defined in ISO/IEC 13818-1 [1]. |
| data_alignment_indicator | Set to "1" indicating that the **subtitle segment**s are aligned with the **PES packet**s. |
| PTS_DTS_flags | Set to '10' to indicate the PTS is present. |
| PTS | Set to the presentation time stamp of the PES packet |
| PES_packet_data_byte | The PES_data_field specified in table 1 of the present document. |

A **PES packet** shall contain subtitle segments for no more than one **subtitle stream**. Each **subtitle stream** shall be carried in **transport stream packet**s identified by a separate **PID**.

## 5.2.2.2 PES packet payload format for TTML subtitling

### 5.2.2.2.1 General PES payload field syntax and segment types

When carrying a DVB TTML **subtitle stream**, the PES_packet_data_bytes shall be encoded as the PES_data_field defined in table 16.

**Table 16: PES data field**

| Syntax | Number of bits | Identifier |
|---|---|---|
| `PES_data_field() {` | | |
| `  segment_mediatime` | 48 | uimsbf |
| `  num_of_segments` | 8 | uimsbf |
| `  for(i=1; i<=num_of_segments; i++){` | | |
| `  segment_type` | 8 | uimsbf |
| `  segment_length` | 16 | uimsbf |
| `  segment_data_field()` | | |
| `  }` | | |
| `  CRC_32` | 32 | uimsbf |
| `}` | | |

**Semantics for PES data field:**

**segment_mediatime:** An unsigned 48-bit integer field that conveys the media time in units of 100 microseconds.

EXAMPLE: A media time of "00:02:03.5" would be represented by a binary encoded value equivalent to (decimal) "1,235,000".

The segment mediatime represents the instant on the timeline of the TTML segment (if present) that corresponds to the **PTS** of the **PES packet**. The segment mediatime and the **PTS** of the **PES packet** ($P_i$) enable decoders to calculate the value of the PTS that corresponds to any computed time in the TTML segment (if present).

**num_of_segments:** This 8-bit field indicates the number of segments in this payload. The value shall not be "0".

**segment_type:** This 8-bit field indicates the type of data contained in segment_data_field. The values of segment_type are defined in table 17. The segment is the basic syntactical element of the DVB TTML subtitling streams. The segment forms the common format shared amongst all elements of this TTML subtitling specification.

**Table 17: Segment type**

| Value | Meaning |
|---|---|
| 0x00 | reserved for future use |
| 0x01 | uncompressed TTML document (as defined in clause 5.2.2.2.3) |
| 0x02 | gzip compressed TTML document (as defined in clause 5.2.2.2.4) |
| 0x03 - 0xFF | reserved for future use |

**segment_length:** This field shall specify the number of bytes contained in the immediately following segment_data_field.

**segment_data_field:** This is the payload of the segment. The syntax of this payload depends on the subtitle segment type. The various types of segment_data_field listed in table 17 are defined in clauses 5.2.2.2.3 to 5.2.2.2.4.

**CRC_32:** This 32-bit field contains the CRC value that gives a zero output of the registers in the decoder defined in annex B of DVB BlueBook A038 [7] after processing the entire PES_data_field.

#### 5.2.2.2.2 Occurrences of segments in one PES packet

Multiple **subtitle segment**s types may be conveyed in the PES_data_field of a **PES packet**, however there shall be no more than one **TTML segment** in a single PES_data_field. Table 18 summarizes the occurrences of subtitles segments in one PES packet.

**Table 18: Subtitle segment occurrences in one PES packet**

| Segment_type | Permitted Occurrences | Type of subtitle segment |
|---|---|---|
| 0x00 | Not applicable | Reserved |
| 0x01 - 0x02 | 0 or 1 | TTML segment |
| 0x03 - 0xFF | Undefined | Reserved |

#### 5.2.2.2.3 Uncompressed TTML document subtitle segment

This segment_type (0x01) is used to convey an uncompressed TTML document. The segment_length shall be set to the length of the TTML document in bytes. The segment_data_field contains the TTML document.

#### 5.2.2.2.4 gzip compressed TTML document subtitle segment

This segment_type (0x02) is used to convey a gzip compressed TTML document. The segment_length shall be set to the length of the compressed TTML document in bytes. The segment_data_field contains the gzip according to IETF RFC 1952 [14] compressed TTML document.

## 5.2.3 TTML segmentation

### 5.2.3.1 General

TTML segments can be of either fixed or variable duration. Exactly zero or one TTML document shall be active at any time instance. In other words, multiple active documents are not allowed. These requirements facilitate the use of a single TTML processor and pipeline.

For pre-recorded programmes, a TTML subtitle document for the whole programme may exist. However, in order to deliver these subtitles via **transport stream** such a TTML document needs to be converted into a number of short duration standalone documents, in a similar way to delivery in DVB DASH profile as defined in ETSI TS 103 285 [6]. Each of these short standalone TTML document are conveyed in a TTML segment as defined in clause 5.2.2.2. The duration of a TTML segment is constrained by the rules defined in clause 5.2.3.4.

#### 5.2.3.2 Concept of ISDs

A TTML document can be considered conceptually as a contiguous sequence of **Intermediate Synchronic Document**s (ISD). As a consequence:

- Whenever some element of TTML content becomes active or inactive an ISD ends and a new ISD begins.
- A period with no content to display is indicated by an ISD with no content.

#### 5.2.3.3 Active Period of a TTML segment

A TTML segment presentation begins when the system clock matches the **PTS** (Pi) of the **PES packet** that conveys the segment. This is the point at which the TTML segment becomes active.

The maximum period of activation (MPA) of a TTML segment, $T_{MPA}$ shall be 5 seconds.

A TTML segment becomes inactive when a later TTML segment becomes active or after being active for $T_{MPA}$ whichever is the earlier. The active period of a TTML segment is the period from when it becomes active until it becomes inactive. Therefore, the active period of a TTML segment shall not exceed $T_{MPA}$. Only one TTML segment can be active at any point in time. Only the ISDs from the currently active TTML segment shall be displayed. Therefore, if there is no active TTML segment, no subtitles shall be presented.

In particular, an ISD may often have an endtime which is after the activation time of the following segment, however this ISD would be expected to be present in the following segment. At the activation time of the following segment (PTS) shall only display the ISDs from newly active segment. Consequently an ISD that starts before the activation time of a segment shall not be displayed before activation time of the segment. An example is shown below in figure 3. An ISD replicated in a following segment may have a different start and end time. In that case, IRDs shall avoid intermittent removal of the subtitle from the screen at the segment boundaries.

#### 5.2.3.4 TTML documents segmentation rules

A TTML segment shall not include any element whose computed end time is earlier than the segment mediatime ($T_i$) of the TTML segment.

A TTML segment shall not include any element whose computed begin time is later than $T_{MPA}$ after the segment mediatime ($T_i$) of the TTML segment, where $T_{MPA}$ is the maximum period of activation (MPA) defined in clause 5.2.3.3.

#### 5.2.3.5 Subtitle stream in the absence of subtitles

If there is no subtitle for more than $T_{MPA}$, an empty TTML segment shall be transmitted. $T_{MPA}$ is the maximum period of activation (MPA) defined in clause 5.2.3.3. It is recommended to use the empty TTML document reproduced from EBU Tech 3381 [17] and shown below:

```
<tt xml:lang="" xmlns="http://www.w3.org/ns/ttml" />
```

This empty TTML document can be conveyed within a PES packet that requires only a single TS packet. This empty document may be used even if it does not conform to the document requirements of the signalled processor profiles(s).

#### 5.2.3.6 Continuation of a subtitle across adjacent segments

If a subtitle continues across the boundary of adjacent segments, that subtitle shall be present in all segments in which it appears.

Even though adjacent TTML document chunks could originate from the same source TTML document, a subtitle present in multiple document chunks may not necessarily have identical timing information in every document chunk in which it is present. Therefore, it is not recommended to use timing information to match subtitles across segment boundaries, for example to meet the requirements of clause 5.2.3.3.

### 5.2.3.7 Untimed elements in TTML segments

When only untimed elements are present in a TTML segment, the segment corresponds to an ISD with unlimited timing boundaries. The elements are only active for the duration of the segment, and hence become active at the activation time of the segment (segment mediatime or $T_i$ or $P_i$) and become inactive at the activation time of the next segment or $T_{MPA}$ after the activation time (whichever is the earlier). $T_{MPA}$ is the maximum period of activation (MPA) defined in clause 5.2.3.3.

### 5.2.3.8 Examples of TTML segmentation

Figure 1 demonstrates an example of how a TTML document can be segmented into a set of conformant TTML documents suitable for delivery as the payload of TTML segments.

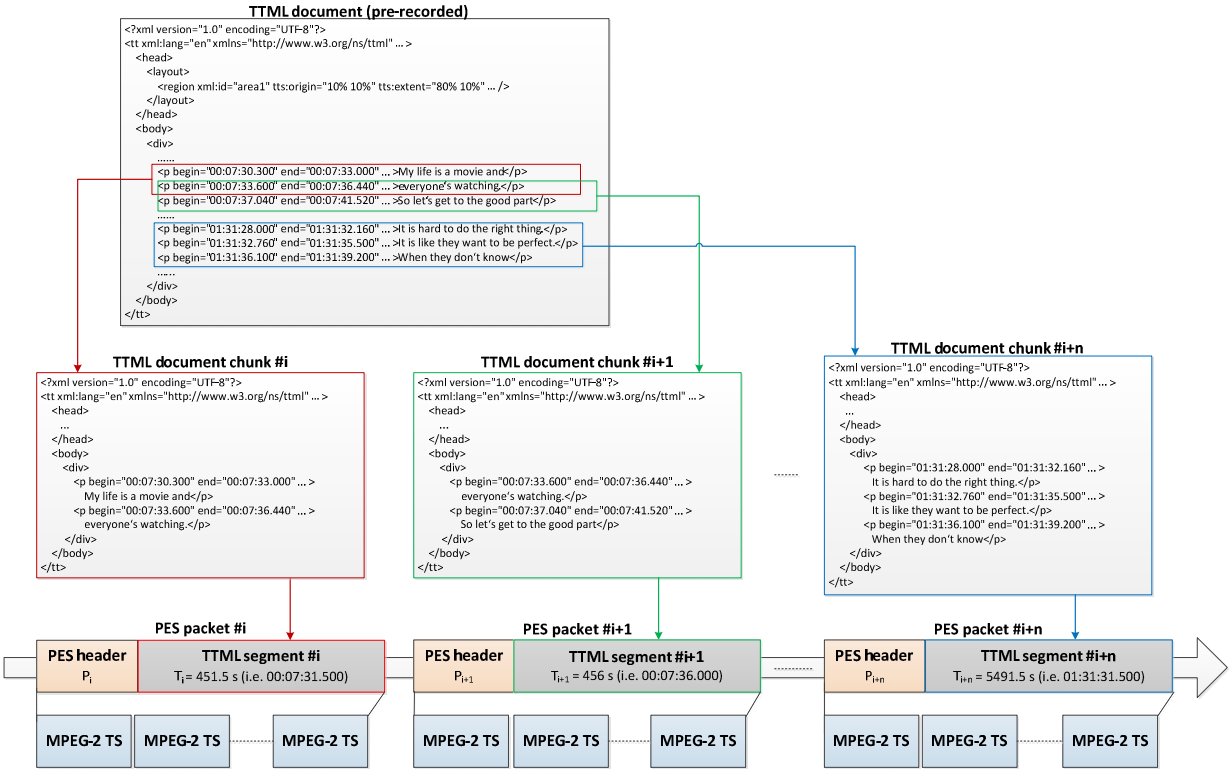


**Figure 1: Example of packetized TTML subtitle**

## 5.2.4 Synchronization

### 5.2.4.1 Relationship between MPEG system clock and TTML timeline

TTML documents are authored with their own timeline. In order to synchronize with other broadcast components such as video and audio a conversion between the TTML document timeline and the MPEG system clock timeline is required. Each TTML PES packet contains a presentation timestamp (PTS) based on the MPEG system clock. The PES payload structure (as defined in clause 5.2.2.2) contains a segment mediatime field which allows the conversion between the TTML and MPEG timelines.

In order to perform this conversion the segment mediatime field needs to be converted to a floating point number of seconds referred to below as $T_i$.

Correspondingly the time expressions in the TTML document also need to be converted to floating point number of seconds, referred to below as $T_x$.

The segment mediatime and its corresponding value $T_i$ do not necessarily coincide with the begin time of an ISD (see examples in clause 5.2.4.3).

Therefore PTS value PTS($T_x$) representing an arbitrary computed TTML time $T_x$ is defined as:

$$PTS(T_x) = ( P_i + (T_x - T_i) \times 90{,}000) \text{ modulo } (2^{33})$$

where:

- PTS($T_x$): indicates the **PTS** for arbitrary TTML time $T_x$;
- $T_x$: indicates an arbitrary internal time of an **ISD** in a TTML segment converted to a floating point number of seconds;
- $P_i$: indicates the **PTS** of the **PES packet** containing that TTML segment;
- $T_i$: indicates the segment mediatime converted to a floating point number of seconds, corresponding to $P_i$ on the MPEG system clock.

In calculating the presentation time of TTML content, the IRD should mitigate for the possibility that the PTS value has rolled-over to zero. Hence calculations using PTS shall use modulo 33 bit arithmetic.

### 5.2.4.2 Failure of TTML segment reception

In the case of failure to receive or corruption of the following segment, the current segment remains active until $T_{MPA}$ after its activation time $P_i$ or $T_i$ or until a valid segment is received whichever occurs first. $T_{MPA}$ is the maximum period of activation (MPA) defined in clause 5.2.3.3.

### 5.2.4.3 Synchronization examples

These example figures, for simplification of representation, only show the final ISD of segment i-1 and first ISD of segment i+1 if applicable. The "active part" of each ISD is shaded with darker orange colour, and the "inactive part" of each ISD in a lighter shade of orange (where applicable). Furthermore the previous (i-1) and following (i+1) PES packets are assumed to be carrying a TTML segment also. $T_{MPA}$ is the maximum period of activation (MPA) defined in clause 5.2.3.3.

Figure 2 shows a simple example of synchronization. The segment contain three ISD, the first ISD's starttime ($T_{b1}$) is coincidence with the activation time of the segment (Pi / Ti). In this example the first and third ISDs may contain no content. The third and last ISD of the segment ends at $T_{e3}$ which coincides with the activation of the next (i+1) segment (Pi+1 / Ti+1). In this simple example all three ISD have no inactive periods.

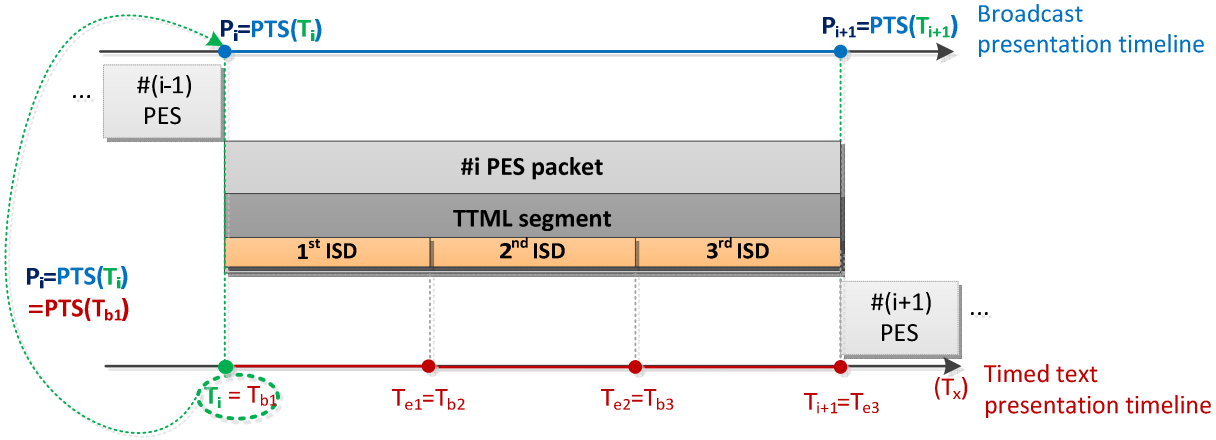


**Figure 2: Example of association between PTS and time for packetized TTML**

Figure 3 shows a common scenario where ISDs overlap between multiple segments. The #i segment contains 3 ISDs, the first ISD overlap from previous segment and has a starting time $T_{b1}$ which is earlier than the activation time of the segment (Pi / Ti). The 3rd and last ISD of the segment overlaps with the following segment, and its end time is after the activation time of the following segment (Pi+1 / Ti+1). The last ISD of segment #i-1 is expected to be replicated in the first ISD of segment #i. Similarly the last ISD of segment #i is expected to be replicated in the first ISD of segment #i+1.

In the time period between $T_{b1}$ and Ti, the IRD only displays the last ISD of segment #i-1. At the activation time of segment #i (Pi / Ti) the IRD switches from the last of ISD of segment #i-1 to the 1st ISD of segment #i in a seamless way. At time $T_{b2}$ the second ISD of the segment is displayed until time $T_{b3}$ when the third and last ISD of segment #i is displayed. At time Ti+1 (Pi+1) the IRD switches seamlessly to display the first ISD of segment #i+1.

Where the figures indicate that PTS(Ti) is not equal to PTS($T_{b1}$) instead of greater than PTS($T_{b1}$). This is because PTS could have rolled over to zero.

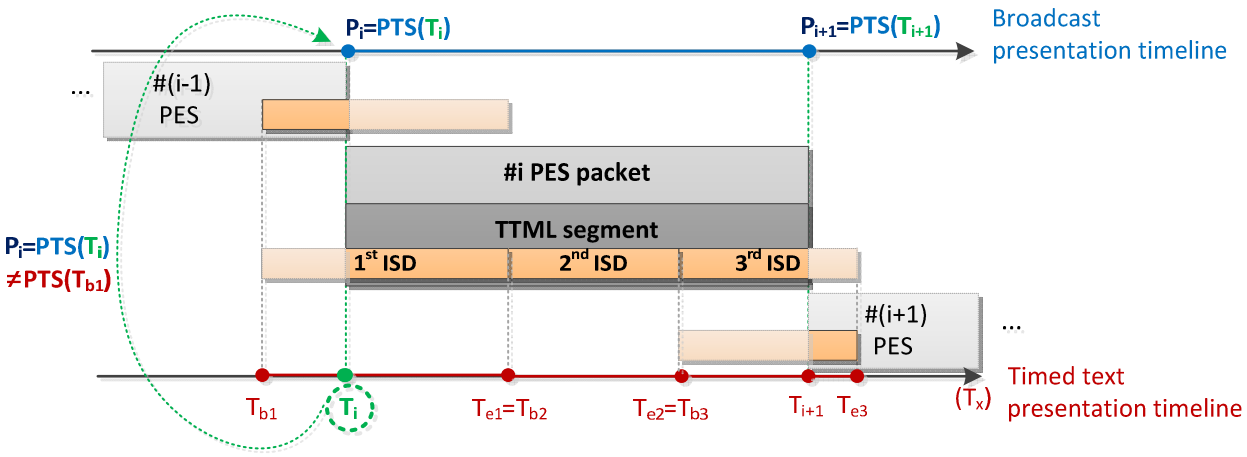


**Figure 3: Example of association between PTS and time for packetized TTML, where an ISD starts before $P_i$ and another ISD ends after $P_{i+1}$**

Figure 4 shows an example similar to figure 3 but where the endtime of the first ISD of segment #i is different to the endtime of last ISD of segment #i-1. However the ISDs are the same. For live subtitling the endtime of an ISD may be provisional and change when a new segment is created.

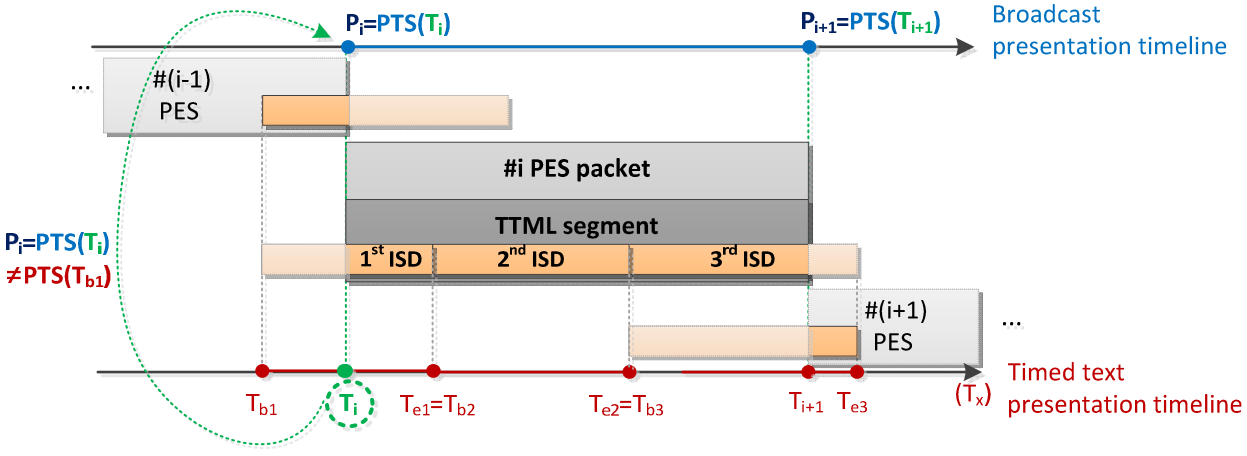


**Figure 4: Example of association between PTS and time for packetized TTML, where the first ISD of segment #i has a different endtime, than the last ISD of segment #i-1**

Figure 5 demonstrates an IRD joining a service. Since the IRD has just tuned to the service, the previous segment is not available. The IRD shall begin to display the first ISD of the segment (#i) at time Ti/Pi.

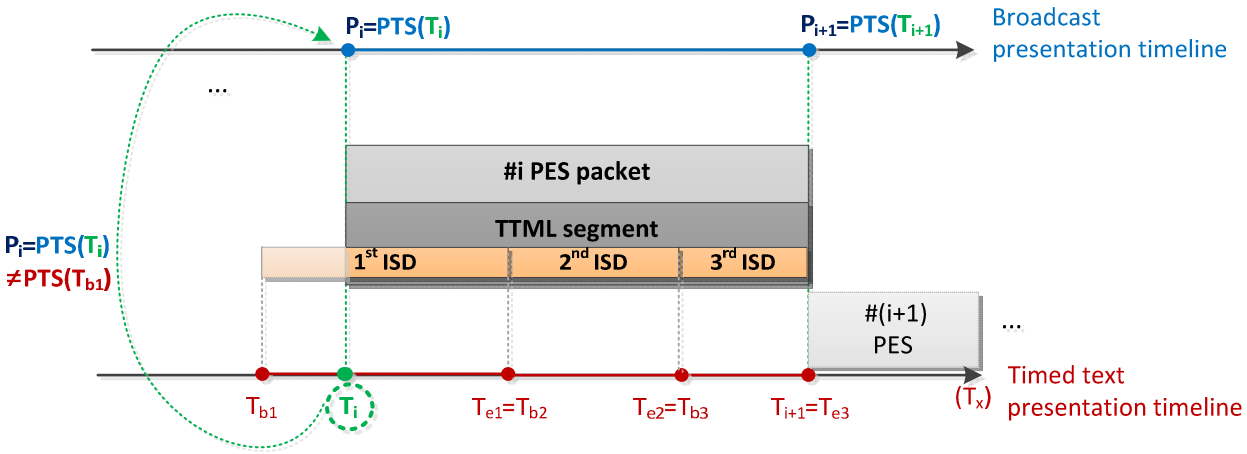


**Figure 5: Example of association between PTS and time for packetized TTML, where no previous PES packet has been received**

Figures 6 and 7 demonstrate the expect behaviour for example shown in figure 3 but where the following TTML segment is not received. This could be due to transmission errors, CRC mismatch, or a technical failure at the broadcast headend. When the following segment is not received the current segment shall remain active until $T_{MPA}$ after its activation time (Ti), therefore Ti+$T_{MPA}$. It is assumed in these examples that the third ISD includes content.

In figure 6, the third ISD ends before Ti+$T_{MPA}$, and is active until $T_{e3}$. IRD assumes an empty ISD between $T_{e3}$ and Ti+$T_{MPA}$.

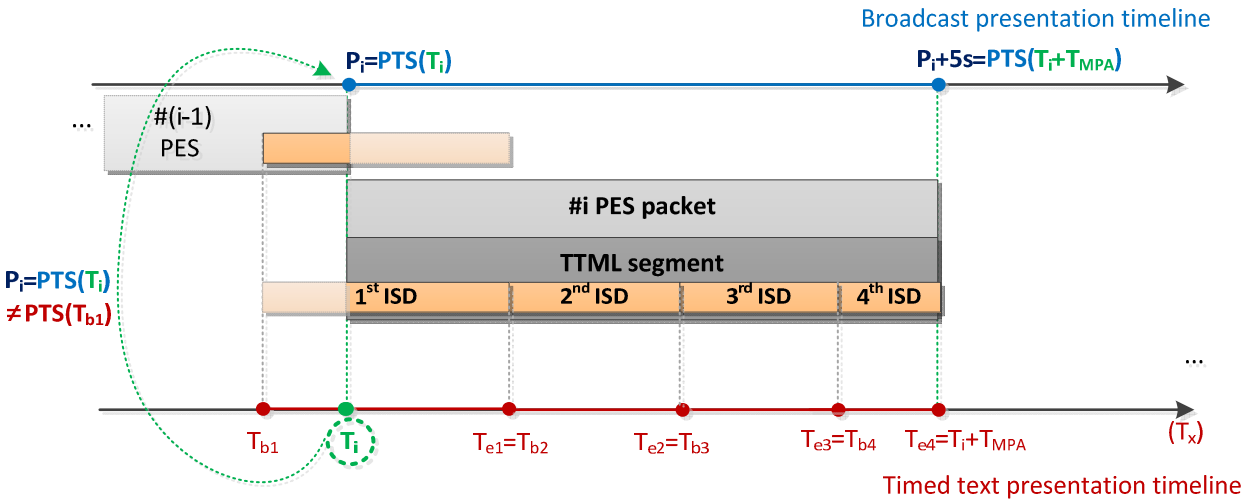


**Figure 6: Example of association between PTS and time for packetized TTML, where no following PES packet is received (e.g. due to temporary reception loss) and the last ISD with content ends before Ti+$T_{MPA}$**

In figure 7, the third ISD ends after Ti+$T_{MPA}$, but is only active until Ti+$T_{MPA}$, and the IRD does not display the ISD after Ti+$T_{MPA}$.

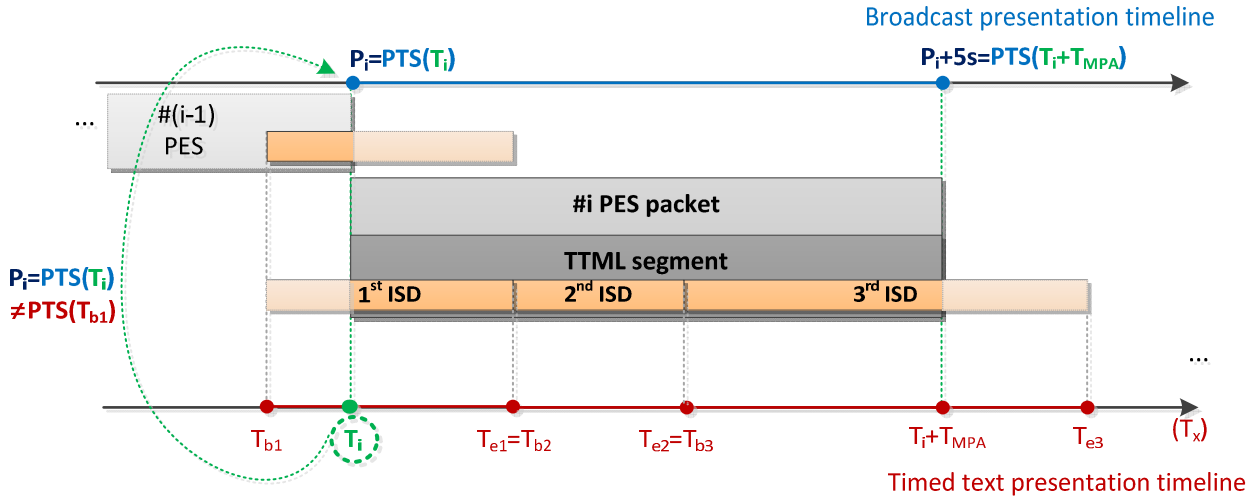


**Figure 7: Example of association between PTS and time for packetized TTML, where no following PES packet is received (e.g. due to temporary reception loss) and the last ISD with content ends after Ti+$T_{MPA}$**

### 5.2.5 Acquisition time

Typically subtitles have a shorter encoding time than video, so they are available for insertion into the transport stream earlier than the corresponding video PES packets. The subtitle packets' duration and insertion time is typically set for operational reasons in order to manage the peak and mean bitrates of the subtitle stream. However care needs to be taken with this approach since it affects the time taken by IRDs to acquire and decode subtitles.

The present document was designed with a target that IRDs can acquire and decode subtitles within a maximum of 5 seconds after joining a service.

For example, if every **TTML segment** has a duration of 3 seconds, and is inserted 2 seconds before its respective presentation time, and the IRD begins to decode the service at the "worst case" time, then the IRD would have to wait just less than 2 + 3 = 5 seconds before being able to present the subtitles. This could in an ideal scenario meet the target maximum acquisition time, ignoring real world decode and data transfer latencies within the IRD.

Conversely, if every **TTML segment** has a duration of 5 seconds, and is inserted 1 second before its respective presentation time, and the IRD begins to decode the service at the "worst case" time, then the IRD would have to wait just less than 5 + 1 = 6 seconds before being able to present subtitles. This could never achieve the target maximum acquisition time.

## 5.3 Font download

### 5.3.1 Introduction

TTML subtitles are intended to be rendered with a particular font or particular font family, however this font may or may not be present in the IRD. The present clause specifies the font delivery signalling mechanism (clause 5.3.2) and the actual font delivery formats (clause 5.3.2.3.2.3):

- The font delivery signalling mechanism uses typical DVB signalling methods to retrieve the information on the corresponding fonts and their locations.
- The font delivery format description includes the file format and the method of delivery of fonts reusing typical DVB mechanisms and formats. The delivery may be via broadcast or via IP.

## 5.3.2 Font delivery signalling mechanism

### 5.3.2.1 Overview

The intention of the signalling of font download information is to indicate which events are using downloadable fonts in advance of broadcast; so that IRDs can pre-acquire or cache the downloadable font(s) in question. Font information, including download location and metadata for each font, only needs to be carried in one location, which is the Downloadable Font Information Table (DFIT).

The linkage descriptor (with linkage_type 0x20 - downloadable font info linkage) allows an IRD to retrieve the **PID** carrying the DFIT. The DVB service, identified by the "DVB triplet" (original_network_id, transport_stream_id, service_id), that carries the DFIT is indicated by the linkage descriptor (with linkage_type 0x20 - downloadable font info linkage). The service indicated by the linkage descriptor will contain an entry in the PMT for the PID conveying the DFIT. The presence of a data_broadcast_id descriptor with a data_broadcast_id of 0x000D (assigned in DVB BlueBook A126 [15]) in the relevant ES_info descriptor loop of the PMT indicate the PID conveying the DFIT.

In addition, the linkage descriptor indicates the font_id of each downloadable font, and whether they are "essential" (required) for an IRD to render a subtitle service or not. Depending on the location of the linkage descriptor, it also indicates whether the font information applies to an event, service, transport stream, bouquet or network.

Information for each font is conveyed in a sub_table of the DFIT so IRDs can easily filter out the sections for the relevant fonts.

Figure 8 describes the process for locating the downloadable font information.

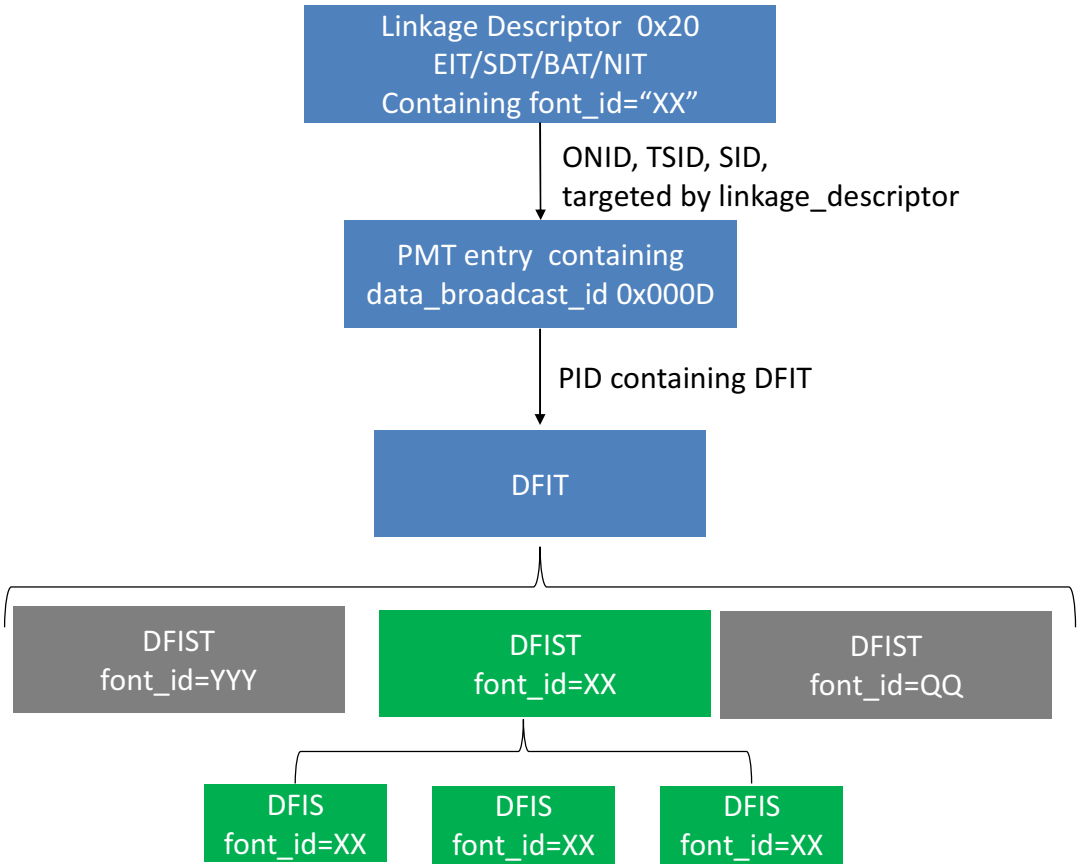


**Figure 8: Downloadable font information retrieval process**

All fonts signalled as "essential" are required for correctly rendering the subtitle service considered, therefore all essential fonts are intended to be retrieved for the applicable event/service/ multiplex,/network/bouquet defined by the linkage descriptor with type 0x20 (see clause 5.3.2.2) that is in scope.

### 5.3.2.2 Linkage descriptor with linkage type 0x20

#### 5.3.2.2.1 Introduction

The linkage descriptor with the linkage type of "0x20" (downloadable font info linkage) signals the associated font family for the event, service, transport stream, bouquet or network and associates it with a font_id, which is used to acquire detailed information on the font and its availability from the DFIT.

#### 5.3.2.2.2 Linkage descriptor with linkage type 0x20 structure

Table 19 specifies the linkage descriptor (as defined in DVB BlueBook A038 [7]) and its private_data_bytes for linkage_type "0x20".

**Table 19: Linkage descriptor with linkage_type 0x20**

| Syntax | Number of bits | Identifier |
|---|---|---|
| linkage_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| transport_stream_id | 16 | uimsbf |
| original_network_id | 16 | uimsbf |
| service_id | 16 | uimsbf |
| linkage_type | 8 | uimsbf |
| **if (linkage_type == 0x20){** | | |
| **font_count** | **8** | **uimsbf** |
| **for (i=0; i<font_count; i++){** | | |
| **essential_font_download_flag** | **1** | **bslbf** |
| **font_id** | **7** | **uimsbf** |
| **}** | | |
| **for (i=0; i<N; i++){** | | |
| **reserved_zero_future_use** | **8** | **bslbf** |
| **}** | | |
| **}** | | |
| } | | |

**Semantics for the linkage descriptor with linkage type 0x20:**

**transport_stream_id:** This is a 16-bit field, which identifies the TS containing the DFIT.

**original_network_id:** This 16-bit field gives the label identifying the network_id of the originating delivery system of the **transport stream** carrying the NIT.

**service_id:** This is a 16-bit field, which identifies the service carrying DFIT in its program map section.

**linkage_type:** This 8-bit field specifies the type of linkage, and shall be set to "0x20".

**font_count:** This 8-bit field specifies the number of downloadable fonts available.

**essential_font_download_flag:** This 1-bit flag when set to "1" indicates that an essential downloadable font is required to present these subtitles. When set to "0" it indicates that the font is a supplementary font.

**font_id:** This 7-bit field indicates a downloadable font within the referenced DFIT. User-defined font_ids in the range "0x00 - 0x6F" (see table 20) are only unique within the referenced DFIT.

**reserved_zero_future_use:** These bits are reserved for future use, and they shall be set to value of "0".

**Table 20: font_id allocation**

| font_id | Description |
|---|---|
| 0x00 - 0x6F | User-defined for downloadable fonts defined by DFIT |
| 0x70 - 0x7F | reserved for future use |

#### 5.3.2.2.3 Linkage descriptor with linkage type 0x20 location, occurrence and prioritization

The linkage descriptor with linkage_type "0x20" is a scoping descriptor (see clause 6.5 of DVB BlueBook A038 [7]).

One and only one linkage descriptor with linkage_type 0x20 is permitted in any descriptor loop. However, a NIT could contain a type 0x20 linkage descriptor in the network (1st) descriptor loop and in one or all of its transport streams (2nd) descriptor loops. Similarly a 0x20 linkage descriptor is allowed in the first or second or both loops of the BAT.

Table 21 summarizes the potential locations of the linkage descriptor with linkage type "0x20" in priority order from highest to lowest priority.

**Table 21: Possible locations of the linkage descriptor with linkage_type 0x20**

| Linkage location | Scope |
|---|---|
| EIT | Event |
| SDT | All events of the service |
| BAT TS (2nd) loop | All events in all services of the **transport stream** |
| NIT TS (2nd) loop | All events in all services of the **transport stream** |
| BAT network (1st) loop | All events in all services of the bouquet |
| NIT network (1st) loop | All events in all services of the network |

For example, a linkage_descriptor (0x20) in the 1st or 2nd loop of NIT or BAT can be overridden for a particular service by a linkage_descriptor (0x20) in the respective SDT.

### 5.3.2.3 Downloadable Font Information Table (DFIT)

#### 5.3.2.3.1 DFIT structure

The DFIT is divided into sub_tables (DFIST) using standard DVB table syntax. A DFIST is comprised of one or more download font info sections (DFIS as defined in table 22). The sections of a DFIT are conveyed together on a single PID. Each DFIST contains the download location and font metadata for a single font or font family.

Multiple services can reference the same DFIT via linkage descriptors (defined in DVB BlueBook A038 [7]) with a linkage_type of 0x20. The PID carrying the DFIT is signalled in the relevant program map section by using a data_broadcast_id descriptor (defined in DVB BlueBook A038 [7]) with a data_broadcast_id of 0x000D (assigned in DVB BlueBook A126 [15]).

Several DFITs may coexist on several PIDs. This may occur if one service is providing its own table whilst the network it belongs to has also a table that applies to the whole network.

The download font info sections (DFIS) use the section structure as defined in ISO/IEC 13818-1 [1].

**Table 22: Syntax of the DFIS**

| Syntax | Number of bits | Identifier |
|---|---|---|
| `downloadable_font_information_section() {` | | |
| `  table_id` | 8 | uimsbf |
| `  section_syntax_indicator` | 1 | bslbf |
| `  reserved_future_use` | 1 | bslbf |
| `  reserved` | 2 | bslbf |
| `  section_length` | 12 | uimsbf |
| `  font_id_extension` | 9 | bslbf |
| `  font_id` | 7 | uimsbf |
| `  reserved` | 2 | bslbf |
| `  version_number` | 5 | uimsbf |
| `  current_next_indicator` | 1 | bslbf |
| `  section_number` | 8 | uimbsf |
| `  last_section_number` | 8 | uimbsf |
| `  for (i=0; i<n; i++) {` | | |
| `  font_info_type` | 8 | uimbsf |
| `  if (font_info_type == 0x00) {` | | |
| `  font_style` | 3 | uimsbf |
| `  font_weight` | 4 | uimsbf |
| `  reserved_zero_future_use` | 1 | bslbf |

| Syntax | Number of bits | Identifier |
|---|---|---|
| }<br>if (font_info_type == 0x01) {<br>reserved_zero_future_use<br>font_file_format<br>uri_length<br>for (j=0; j<n; j++){<br>uri_char<br>}<br>}<br>if (font_info_type == 0x02) {<br>font_size<br>}<br>if (font_info_type >= 0x02) {<br>font_info_length<br>for (j=0; j<n; j++){<br>text_char<br>}<br>}<br>}<br>CRC_32<br>} | <br><br>4<br>4<br>8<br><br>8<br><br><br><br>16<br><br><br>8<br><br>8<br><br><br><br>32 | <br><br>bslbf<br>uimsbf<br>uimsbf<br><br>bslbf<br><br><br><br>uimsbf<br><br><br>uimsbf<br><br>uimsbf<br><br><br><br>rpchof |

**Semantics for the fields for the DFIS:**

**table id:** This value shall be set to "0x4C" according to DVB BlueBook A038 [7].

**section_syntax_indicator:** This 1-bit indicator shall be set to "1" as the section follows the generic section syntax beyond the private_section_length_field.

**section_length:** This is a 12-bit field. It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 4 093 bytes so that the entire section has a maximum length of 4 096 bytes.

**font_id_extension:** This 9-bit field bits shall all be set to zero.

**font_id:** This field identifies all sections of a single sub_table. Each sub_table describes exactly one font or font family. The font_id and font_id_extension field together constitute the 16-bit table_id extension field as defined in ISO/IEC 13818-1 [1].

**version_number:** This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table.

**current_next_indicator:** This 1-bit indicator, when set to "1" indicates that the sub_table is the currently applicable sub_table. When the bit is set to "0", it indicates that the sub_table sent is not yet applicable and shall be the next sub_table to be valid.

**section_number:** This 8-bit field gives the number of the section. The section_number of the first section in the sub_table shall be "0x00". The section_number shall be incremented by 1 with each additional section with the same font_id.

**last_section_number:** This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the sub_table of which this section is part.

**font_info_type:** See definition in clause 5.3.2.3.2.

**font_style:** See definition in clause 5.3.2.3.2.2.

**font_weight:** See definition in clause 5.3.2.3.2.2.

**font_file_format:** See definition in clause 5.3.2.3.2.3.

**uri_length:** See definition in clause 5.3.2.3.2.3.

**uri_char:** See definition in clause 5.3.2.3.2.3.

**font_size:** This 16-bit field defined the font size (height) in pixels.

**font_info_length:** This 8-bit field specifies the number of bytes that follow the character string field.

**text_char:** This is an 8-bit field containing a string of characters for the relevant font_info_type. This string shall be encoded in UTF-8.

**reserved_zero_future_use:** These bits are reserved for future use, and they shall be set to value of "0".

### 5.3.2.3.2 font_info_type in the loop of the DFIS

#### 5.3.2.3.2.1 Allowable font_info_type in the DFIS

Table 23 lists all possible font_info_type in the DFIS.

**Table 23: font_info_type values in the loop of the DFIS**

| font_info_type value | font_info_type label | Permitted Occurrences | Description |
|---|---|---|---|
| 0x00 | font_style_weight | 1 or more | This 8-bit field describes the font_style and the font_weight of the downloadable font. See clause 5.3.2.3.2.2. |
| 0x01 | font file URI | 1 or more | Specifies the DVB URI location of a font file which can be downloaded from the internet or an object data carousel (see clause 5.3.2.3.2.3). |
| 0x02 | font_size | 0 or more | The font size (height) in pixels |
| 0x03 | font_family | 1 | It specifies the font_family (see clause 8.3.5 in [2]) of the downloadable font. This string shall be encoded in UTF-8. |
| 0x04 - 0xFF | reserved | n/a | These field values are reserved for future use. |

#### 5.3.2.3.2.2 font_info_type 0x00 semantics

**font_style:** This 3-bit field indicate the style of the downloadable font, as defined by table 24.

**Table 24: font_style encoding**

| font_style | Meaning |
|---|---|
| 0 | undefined |
| 1 | normal |
| 2 | italic |
| 3 | oblique |
| 4 - 7 | reserved for future use |

**font_weight:** This 4-bit field indicate the weight of the downloadable font, as defined by table 25.

**Table 25: font_weight encoding**

| font_weight | Meaning |
|---|---|
| 0 | undefined |
| 1 | normal |
| 2 | bold |
| 3 - 15 | reserved for future use |

#### 5.3.2.3.2.3 font_info_type 0x01 semantics

**font_file_format:** This 4-bit field indicates the format of font file referenced, as defined in table 26.

**Table 26: font file format**

| font_file_format | Meaning |
|---|---|
| 0 | Open Font Format specified in ISO/IEC 14496-22:2015 [19] |
| 1 | WOFF (Web Open Font Format) File Format [18] |
| 2 - 15 | reserved for future use |

**uri_length:** This is an 8-bit field specifying the length of the following DVB URI string.

**uri_char:** This is an 8-bit field containing a string of characters that shall be encoded in UTF-8.This string of uri_char fields specifies a DVB URI string as defined in ETSI TS 102 851 [13]. The URI can refer to a download location on the Internet or a file within a DSM-CC object carousel (see clause 5.3.4).

#### 5.3.2.3.2.4 Multiple font file URIs

A DFIST may contain more than one font file URI. In that case, the contained font file URIs shall be processed in the order they are signalled in the DFIST until the referenced font file is successfully retrieved or all contained font file URIs have been processed. Fonts may be available in different formats each signalled by a separate URI.

The use of more than one font file URI allows prioritization e.g. it is possible to specify a main URI as first font file URI and backup URI(s) as subsequent font file URI(s) in the DFIST. Initially, the IRD will try to retrieve the font file from the main URI. If this fails, the IRD will instead try to query the backup URI(s).

### 5.3.3 IP download fonts

The URI for IP downloaded fonts shall use the https URI scheme for security reasons. IRDs shall only download the font if they can authenticate the TLS certificate.

### 5.3.4 Broadcast download fonts

The DFIT may include URI(s) which refer to fonts provided via a DSM-CC object carousel, as defined in clause 11 of ETSI EN 301 192 [11]. The URI to a font in DSM-CC object carousel can take many forms as defined in ETSI TS 102 851 [13]. A simple form of such a URI would have the format:

- dvb://original_network_id.transport_stream_id.service_id.component_tag/directory-path/filename.

EXAMPLE: dvb://233a.1041.10bf.64/fonts/english/SanSerif/Droid.otf.

The "DVB triplet" of Original Network Id (0x233a), Transport Stream Id (0x1041) and Service Id (0x10bf) identifies the service carrying the carousel.

The component_tag (100 or 0x64 in this example) indirectly indicates the PID (via a stream_identifier descriptor) containing the DSI of the object carousel.

The directory-path and filename identify the path to the font within the object carousel.

### 5.3.5 Downloaded font file, font name and font-id association

When a font is downloaded, it shall be associated in storage with font-id, font-family and the scope of the linkage descriptor that was used to locate the downloaded font, see clause 5.3.2.2.3.

# 6 IRD requirements

## 6.1 General support

IRDs shall support clause 5.2 and react according to clause 5.2.1 with the following restrictions:

- IRDs are required to support the default conformance point (see clause 4).
- IRDs are not required to show subtitle services unless enabled by the user.

## 6.2 Forward compatibility

For the purpose of future enhancements to the present document:

- IRDs shall ignore subtitle segment_types that are unknown or that they do not support.
- IRDs shall ignore fields that they do not support, as well as reserved fields in segments, descriptors and tables.

## 6.3 Support of PSI/SI signalling and subtitle service selection

### 6.3.1 PSI/SI signalling

IRDs shall support the PSI/SI signalling defined in clause 5.2 and react accordingly, with the following restrictions applicable to clause 5.2.1:

- IRDs are not required to parse the qualifier field in the TTML subtitling descriptor in the PMT.

For the avoidance of doubt, all the dvb_ttml_profile fields present within the TTML_subtitling_descriptor shall be parsed, because there are multiple ways of signalling the processor requirements for the subtitle stream, see clause 5.2.1.2.

### 6.3.2 Determining which subtitle services can be rendered

IRDs shall obey the following requirements:

- Subtitle services may have fonts flagged as "essential" in the subtitle stream's TTML subtitle descriptor, carried in the relevant ES_info descriptor loop of the PMT. If these essential fonts are not available to the IRD, the subtitle service shall not be presented to the viewer.
- IRDs shall only display subtitles if they support at least one of the signalled dvb_ttml_profiles.

### 6.3.3 Subtitle service selection

The IRD should select a subtitle service to be rendered according to user preferences like language, accessibility and purpose.

The IRD should process the qualifier and react accordingly to offer the user a better selection of subtitle services.

IRDs that do not process the qualifier field shall select the **subtitle stream** based upon other fields in the TTML subtitling descriptor (e.g. language, purpose). If the PMT contains multiple relevant **subtitle stream**s which only differ in terms of the qualifier field (i.e. **subtitle service qualifier variants**), IRDs shall select the first qualifier variant in the PMT.

# 6.4 Font support in IRDs

## 6.4.1 Supported fonts and font family name mapping

IRD shall support the following fonts listed below in table 27 and their respective mapping to generic font family names or family names.

**Table 27: Required fonts and their font family name mappings**

| font_family_name (one of:) | Font |
|---|---|
| monospaceSansSerif | "**Letter Gothic 12 Pitch**" or **another with same font metrics** with the support for the Unicode character range "Generic Western European character set" as defined in annex C of ETSI TS 102 809 [12] but excluding the Unicode character codes 0149 and 066B. |
| Tiresias | "**Tiresias™ Screenfont**" **v8.03** or **another with same font metrics** with the support for the Unicode character range " Generic Western European character set" as defined in annex C of ETSI TS 102 809 [12] but excluding the Unicode character codes 0149 and 066B. |
| proportionalSansSerif<br>sansSerif | **Arial** or **Helvetica** or **Liberation Sans** or another proportionally spaced sans serif font meeting the reference font requirements from IMSC1 [4]. The font should include all code points required to present text in the scripts used by the languages required in the IRD's target market. |
| default<br>monospace<br>monospaceSerif<br>serif | **Courier New** or **Liberation Mono** or another monospaced serif font meeting the reference font requirements from IMSC1 [4]. The font should include all code points required to present text in the scripts used by the languages required in the IRD's target market. |
| NOTE 1: The font mappings defined in table 27 for monospaceSerif & proportionalSansSerif allow IMSC1 Text Profile TTML subtitles to be interoperable with IRDs compliant with the default conformance point, with additional constraints (see clause 4.2.6).<br>NOTE 2: The only generic font name mapping not defined in table 27 is for "proportionalSerif". This is not defined because proportional serif fonts have not generally been used for subtitling. | |

## 6.4.2 Downloadable font retrieval

IRDs should support the specifications in clause 5.3.

The font download functionality allows for flexible variations that can be adapted to the broadcast operation structure in different markets. It may potentially lead to high IRD complexity and storage requirements unless local profiles constrain their operational use, to restrict the number of fonts to a manageable number, if this functionality is required in a market.

Furthermore, it should be noted that it is not guaranteed that a signalled font can be retrieved within any given time period for the following reasons:

- IRD in standby.
- IRD is recording.
- IRD is downloading software update.
- IRD is not tuned to the multiplex where the DFIT is located.
- IRD is not tuned to the multiplex carrying the font to be downloaded.
- IRD does not have an IP connection to download the font.
- IRD has insufficient storage to download a font.

### 6.4.3 Font selection for subtitle rendering

After successful selection of the appropriate subtitle service to be presented (see clause 6.3.1) the IRD should select the appropriate font by applying a suitable font matching algorithm. If font downloads are supported by the IRD then any downloaded fonts shall only be considered by the font matching algorithm if they are in scope of the linkage descriptor from where they were located (see clause 5.3.2.2.3).

The font matching algorithm shall have the following features:

- Using case insensitive matching, find all available fonts present in the IRD that match either the specific family name or the generic family name.
- Reduce the set of matching fonts by first using font-style then font-weight, then font-size.
- If there is only a single matching font in the reduced set, then this font shall be used.
- In case the algorithm returns multiple fonts, the font corresponding in the font_id in the TTML subtitling descriptor shall be used.
- In case no font present on the IRD matches any essential font signalled, the IRD shall not present the subtitles.

A suitable font matching algorithm can be found in the CSS font matching algorithm [i.2].

## 6.5 Rendering requirements

### 6.5.1 Playback free-running when A/V sync lost temporarily

In video, audio and subtitles are synchronized using the PES presentation timestamps. However in case of temporary corruption of the transport stream (e.g. transmission errors) IRDs should endeavour to continue playback of subtitles even if the video is interrupted and a IRD may skip frames for error concealment.

Subtitles are often a representation of the audio, and as such subtitles and audio allow viewers to continue to follow the programme even when video may have been corrupted.

For example, if a frame cannot be decoded successfully, but the audio can be played back continuously, then the subtitle presentation will progress smoothly.

### 6.5.2 Behaviour during trick play

During trick play the TTML timeline shall remain synchronized with the video presentation timeline.

Hence, in the "Pause" state, the TTML timeline is frozen and any subtitle being displayed when entering the "Pause" state shall remain displayed, since the timeline is frozen within a TTML segment active period (see clause 5.2.3.3).

### 6.5.3 Subtitle composition with HDR video

IRDs shall take into account the colour space and/or HDR format of the video when combining subtitles with video for display.

It is recommended that especially for HDR services that subtitle service providers provide subtitles with a background with suitable contrast.

Subtitles in TTML documents are expressed in standard dynamic range sRGB as defined in IEC 61966-2-1 [i.3]. Often subtitles use maximum values for primary colours, e.g. "white" often being expressed as (0xFF, 0xFF, 0xFF) in RGB. Therefore care should be taken when mapping these colours for compositing with high dynamic range video, since incorrect mapping of subtitle colours for a high dynamic range display could produce an unreadable result with extremely bright text.

### 6.5.4 Cropping and scaling of video

There are scenarios in which IRDs may need to scale or crop video for display purposes. In such scenarios the IRD should use ittp:activeArea to avoid cropping the subtitles and to maintain positional alignment with respect to the video.

## 6.6 Additional TTML processor requirements

The following constraints shall be applied by the IRD:

- The default value for the tts:color style attribute shall be white ("#FFFFFF").
- The processor shall not reject TTML documents that contain attributes in unrecognised namespaces, but unrecognised attributes shall be ignored.
- The processor shall support the itts:fillLineGap, ebutts:multiRowAlign and ebutts:linePadding style attributes.

When applied to an EBU-TT-D processor, these constraints ensure that the output is consistent with an IMSC1 Text Profile processor.

# Annex A (informative): Change History

| Version | Information about changes |
|---|---|
| 1.1.1 | Initial version of the specification |
| | |
| | |

# History

| Document history | | | |
|---|---|---|---|
| V1.1.1 | January 2018 | EN Approval Procedure | AP 20180429: 2018-01-29 to 2018-04-30 |
| V1.1.1 | May 2018 | Publication | |
| | | | |
| | | | |
| | | | |