



**Digital Video Broadcasting (DVB);
Specification for Service Information (SI)
in DVB systems**

DVB Document A038

December 2017





This page is left intentionally blank

Contents

| | |
|--|----|
| Intellectual Property Rights | 8 |
| Foreword..... | 8 |
| Modal verbs terminology | 8 |
| 1 Scope | 9 |
| 2 References | 9 |
| 2.1 Normative references | 9 |
| 2.2 Informative references | 12 |
| 3 Definitions and abbreviations..... | 12 |
| 3.1 Definitions | 12 |
| 3.2 Abbreviations..... | 15 |
| 4 Service Information (SI) description..... | 19 |
| 5 Service Information (SI) tables | 21 |
| 5.1 SI table mechanism | 21 |
| 5.1.1 Use of table sections..... | 21 |
| 5.1.2 Mapping of sections into Transport Stream (TS) packets | 22 |
| 5.1.3 Coding of PID and table_id fields | 22 |
| 5.1.4 Repetition rates and random access..... | 24 |
| 5.1.5 Scrambling | 24 |
| 5.1.6 Bit order and transmission order | 24 |
| 5.2 Table definitions | 26 |
| 5.2.0 Introduction | 26 |
| 5.2.1 Network Information Table (NIT) | 27 |
| 5.2.2 Bouquet Association Table (BAT) | 28 |
| 5.2.3 Service Description Table (SDT) | 30 |
| 5.2.4 Event Information Table (EIT)..... | 31 |
| 5.2.5 Time and Date Table (TDT)..... | 34 |
| 5.2.6 Time Offset Table (TOT)..... | 34 |
| 5.2.7 Running Status Table (RST) | 35 |
| 5.2.8 Stuffing Table (ST) | 36 |
| 5.2.9 Discontinuity Information Table (DIT)..... | 36 |
| 5.2.10 Selection Information Table (SIT) | 36 |
| 6 Descriptors | 36 |
| 6.0 Introduction..... | 36 |
| 6.1 Descriptor identification and location..... | 36 |
| 6.2 Descriptor coding..... | 38 |
| 6.2.0 General principles | 38 |
| 6.2.1 Adaptation field data descriptor | 38 |
| 6.2.2 Ancillary data descriptor | 39 |
| 6.2.3 Announcement support descriptor..... | 39 |
| 6.2.4 Bouquet name descriptor..... | 41 |
| 6.2.5 CA identifier descriptor..... | 41 |
| 6.2.6 Cell frequency link descriptor | 42 |
| 6.2.7 Cell list descriptor | 42 |
| 6.2.8 Component descriptor | 43 |
| 6.2.9 Content descriptor | 52 |
| 6.2.10 Country availability descriptor | 55 |
| 6.2.11 Data broadcast descriptor | 55 |
| 6.2.12 Data broadcast id descriptor | 56 |
| 6.2.13 Delivery system descriptors | 56 |
| 6.2.13.1 Cable delivery system descriptor..... | 56 |
| 6.2.13.2 Satellite delivery system descriptor | 58 |
| 6.2.13.3 S2 satellite delivery system descriptor | 59 |
| 6.2.13.4 Terrestrial delivery system descriptor | 60 |
| 6.2.14 DSNG descriptor | 62 |

| | | |
|----------|---|-----|
| 6.2.15 | Extended event descriptor | 62 |
| 6.2.16 | Extension descriptor | 64 |
| 6.2.17 | Frequency list descriptor | 64 |
| 6.2.18 | FTA content management descriptor | 65 |
| 6.2.18.0 | Semantics and syntax of the FTA content management descriptor | 65 |
| 6.2.18.1 | Scope of the FTA content management descriptor | 67 |
| 6.2.19 | Linkage descriptor | 67 |
| 6.2.19.0 | Semantics and syntax of the linkage descriptor | 67 |
| 6.2.19.1 | Mobile hand-over linkage | 69 |
| 6.2.19.2 | Event linkage | 69 |
| 6.2.19.3 | Extended event linkage | 70 |
| 6.2.20 | Local time offset descriptor | 73 |
| 6.2.21 | Mosaic descriptor | 74 |
| 6.2.22 | Multilingual bouquet name descriptor | 76 |
| 6.2.23 | Multilingual component descriptor | 77 |
| 6.2.24 | Multilingual network name descriptor | 77 |
| 6.2.25 | Multilingual service name descriptor | 78 |
| 6.2.26 | Near Video On Demand (NVOD) reference descriptor | 78 |
| 6.2.27 | Network name descriptor | 79 |
| 6.2.28 | Parental rating descriptor | 79 |
| 6.2.29 | Partial Transport Stream (TS) descriptor | 80 |
| 6.2.30 | PDC descriptor | 80 |
| 6.2.31 | Private data specifier descriptor | 81 |
| 6.2.32 | Scrambling descriptor | 81 |
| 6.2.33 | Service descriptor | 82 |
| 6.2.34 | Service availability descriptor | 83 |
| 6.2.35 | Service list descriptor | 84 |
| 6.2.36 | Service move descriptor | 84 |
| 6.2.37 | Short event descriptor | 85 |
| 6.2.38 | Short smoothing buffer descriptor | 85 |
| 6.2.39 | Stream identifier descriptor | 87 |
| 6.2.40 | Stuffing descriptor | 87 |
| 6.2.41 | Subtitling descriptor | 88 |
| 6.2.42 | Telephone descriptor | 89 |
| 6.2.43 | Teletext descriptor | 90 |
| 6.2.44 | Time shifted event descriptor | 91 |
| 6.2.45 | Time shifted service descriptor | 91 |
| 6.2.46 | Transport stream descriptor | 91 |
| 6.2.47 | VBI data descriptor | 92 |
| 6.2.48 | VBI teletext descriptor | 93 |
| 6.3 | Extended descriptor identification and location | 93 |
| 6.4 | Extended descriptor coding | 94 |
| 6.4.0 | Introduction | 94 |
| 6.4.1 | Audio preselection descriptor | 94 |
| 6.4.2 | CI ancillary data descriptor | 97 |
| 6.4.3 | CP descriptor | 97 |
| 6.4.4 | CP identifier descriptor | 98 |
| 6.4.5 | CPCM delivery signalling descriptor | 98 |
| 6.4.6 | Delivery system descriptors | 98 |
| 6.4.6.1 | C2 delivery system descriptor | 98 |
| 6.4.6.2 | SH delivery system descriptor | 100 |
| 6.4.6.3 | T2 delivery system descriptor | 104 |
| 6.4.6.4 | C2 bundle delivery system descriptor | 106 |
| 6.4.6.5 | S2X satellite delivery system descriptor | 107 |
| 6.4.7 | Image icon descriptor | 109 |
| 6.4.8 | Message descriptor | 112 |
| 6.4.9 | Network change notify descriptor | 112 |
| 6.4.10 | Service relocated descriptor | 114 |
| 6.4.11 | Supplementary audio descriptor | 114 |
| 6.4.12 | Target region descriptor | 116 |
| 6.4.13 | Target region name descriptor | 118 |
| 6.4.14 | T2-MI descriptor | 119 |

| | | |
|---|--|------------|
| 6.4.15 | URI linkage descriptor | 120 |
| 6.4.16 | Video depth range descriptor | 121 |
| 6.4.16.0 | Semantics and syntax of the video depth range descriptor | 121 |
| 6.4.16.1 | Production disparity hint | 122 |
| 6.5 | Scoping rules for scoping descriptors | 122 |
| 7 | Storage Media Interoperability (SMI) measures | 123 |
| 7.0 | Introduction..... | 123 |
| 7.1 | SMI tables..... | 123 |
| 7.1.0 | General principles | 123 |
| 7.1.1 | Discontinuity Information Table (DIT)..... | 123 |
| 7.1.2 | Selection Information Table (SIT) | 124 |
| 7.2 | SMI descriptors..... | 125 |
| 7.2.0 | Introduction | 125 |
| 7.2.1 | Partial Transport Stream (TS) descriptor | 125 |
| Annex A (normative): Coding of text characters..... | | 126 |
| A.0 | General principles | 126 |
| A.1 | Control codes | 126 |
| A.2 | Selection of character table | 126 |
| Annex B: Void | | 140 |
| Annex C (informative): Conversion between time and date conventions | | 141 |
| Annex D (normative): Service information implementation of AC-3, Enhanced AC-3, and AC-4 audio in DVB systems..... | | 143 |
| D.0 | Introduction | 143 |
| D.1 | AC-3 and Enhanced AC-3 component types | 143 |
| D.2 | AC-3 descriptor | 144 |
| D.3 | AC-3 descriptor syntax and semantics | 144 |
| D.4 | Enhanced_AC-3 descriptor | 146 |
| D.5 | Enhanced_AC-3 descriptor syntax and semantics | 146 |
| D.6 | AC-4 descriptor | 148 |
| D.7 | AC-4 descriptor syntax and semantics | 148 |
| D.8 | Use of the supplementary audio descriptor with AC-4 | 150 |
| Annex E (normative): Usage of the Scrambling_descriptor..... | | 151 |
| Annex F (informative): ISO 639 Language Descriptor for "original audio" Soundtrack | | 152 |
| Annex G (normative): Service information implementation of DTS® coded audio in DVB systems .. | | 153 |
| G.0 | Introduction | 153 |
| G.1 | DTS® and DTS-HD® descriptors | 153 |
| G.2 | DTS® Descriptor | 153 |
| G.2.0 | Use of the DTS® descriptor..... | 153 |
| G.2.1 | Syntax and semantics for the DTS® descriptor | 153 |
| G.3 | DTS-HD® descriptor | 156 |
| G.3.1 | DTS-HD® descriptor syntax | 156 |
| G.3.2 | Substream information..... | 157 |
| G.3.3 | Asset information..... | 159 |
| G.3.4 | Component type..... | 160 |

| | | |
|---|---|------------|
| G.4 | Use of DTS-HD® in Receiver Mixed Applications for Single PID and Multiple PID Implementations | 162 |
| Annex H (normative): Service information implementation of AAC coded audio in DVB systems..... | | 163 |
| H.0 | Introduction | 163 |
| H.1 | AAC Audio descriptor | 163 |
| H.2 | AAC descriptor | 163 |
| H.2.0 | Use of the AAC descriptor..... | 163 |
| H.2.1 | Syntax and semantics for the AAC descriptor | 163 |
| Annex I (normative): Assignment and interpretation of the service_type field | | 165 |
| I.1 | Background | 165 |
| I.2 | Assignment of service_type | 165 |
| I.2.0 | General principles | 165 |
| I.2.1 | service_type "digital television service" (0x01) | 165 |
| I.2.2 | service_type "H.264/AVC" (various) | 166 |
| I.2.3 | service_type "H.264/AVC frame compatible stereoscopic HD" (various) | 166 |
| I.2.4 | service_type "advanced codec digital radio sound service" (0x0A) | 166 |
| I.2.5 | service_type "HEVC digital television service" (0x1F) | 167 |
| I.2.5.0 | General principles | 167 |
| I.2.5.1 | Signalling for service frame compatible plano-stereoscopic 3DTV for HEVC coded services | 168 |
| I.2.5.2 | Signalling for HDR and/or frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz, but with a HEVC half frame rate temporal video sub-bitstream frame rate lower than or equal to 60 Hz | 169 |
| I.2.5.3 | Spatial, temporal, and dynamic range characteristics..... | 170 |
| I.2.5.4 | Summary of signalling for different bitstream profiles using service_type 0x1F | 171 |
| I.2.6 | service_type "HEVC digital television service with HDR and/or a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz" (0x20)..... | 171 |
| I.2.6.1 | General principles | 171 |
| I.2.6.1 | Summary of signalling for different bitstream profiles using service_type 0x20 | 174 |
| I.2.7 | Summary of signalling for HEVC bitstream profiles using service_type 0x1F or 0x20 (informative) | 174 |
| Annex J (normative): Signalling of Supplementary Audio..... | | 176 |
| J.1 | Overview | 176 |
| J.2 | Receiver-mix supplementary audio..... | 176 |
| J.2.1 | Introduction..... | 176 |
| J.2.2 | PSI PMT signalling..... | 177 |
| J.2.3 | EIT signalling | 177 |
| J.2.3.0 | General principles | 177 |
| J.2.3.1 | Visually impaired audio description..... | 177 |
| J.3 | Broadcast-mix supplementary audio | 177 |
| J.3.1 | Introduction..... | 177 |
| J.3.2 | PSI PMT signalling..... | 178 |
| J.3.3 | EIT signalling | 178 |
| J.3.3.0 | General principles | 178 |
| J.3.3.1 | Visually impaired audio description..... | 178 |
| J.4 | PSI signalling of audio purpose | 178 |
| J.5 | SAOC-DE parametric data streams..... | 179 |
| J.5.1 | Introduction..... | 179 |
| J.5.2 | PSI PMT signalling..... | 179 |
| J.5.3 | EIT signalling | 179 |
| Annex K (normative) Extended event linkage descriptor usage | | 181 |
| Annex L (normative) Service information implementation of DTS Neural Surround™ coded audio in DVB systems..... | | 185 |

| | | |
|---|---|------------|
| L.0 | Introduction | 185 |
| L.1 | DTS® Neural Descriptor | 185 |
| Annex M (normative): Signalling of next-generation audio | | 187 |
| M.1 | Overview | 187 |
| M.2 | PSI PMT Signalling | 187 |
| M.3 | Mapping of codec specific values to the audio preselection descriptor (informative) | 187 |
| Annex N (informative): Bibliography | | 189 |
| Annex O (informative): Change History | | 190 |
| History | | 193 |

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://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.

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 SACCONNEX (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 standardisation, interoperability and future proof specifications.

| Proposed national transposition dates | |
|--|---------------------------------|
| Date of latest announcement of this EN (doa): | 3 months after ETSI publication |
| Date of latest publication of new National Standard or endorsement of this EN (dop/e): | 6 months after doa |
| Date of withdrawal of any conflicting National Standard (dow): | 6 months after doa |

Modal verbs terminology

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

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

1 Scope

The present document specifies the Service Information (SI) data which forms a part of DVB bitstreams, in order that the user can be provided with information to assist in selection of services and/or events within the bitstream, and so that the Integrated Receiver Decoder (IRD) can automatically configure itself for the selected service. SI data for automatic configuration is mostly specified within ISO/IEC 13818-1 [15] as Program Specific Information (PSI).

The present document specifies additional data which complements the PSI by providing data to aid automatic tuning of IRDs, and additional information intended for display to the user. The manner of presentation of the information is not specified in the present document, and IRD manufacturers have freedom to choose appropriate presentation methods.

It is expected that Electronic Programme Guides (EPGs) will be a feature of Digital TV transmissions.

The definition of an EPG is outside the scope of the present document (i.e. the SI specification), but the data contained within the SI specified in the present document may be used as the basis for an EPG.

Rules of operation for the implementation of the present document are specified in TS 101 211 [i.1].

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 <http://docbox.etsi.org/Reference>.

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

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

- [1] ETSI EN 300 231: "Television systems; Specification of the domestic video Programme Delivery Control system (PDC)".
- [2] ETSI EN 300 401: "Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers".
- [3] ETSI EN 300 706: "Enhanced Teletext specification".
- [4] ETSI EN 301 192: "Digital Video Broadcasting (DVB); DVB specification for data broadcasting".
- [5] ETSI EN 301 210: "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for Digital Satellite News Gathering (DSNG) and other contribution applications by satellite".
- [6] ETSI EN 301 775: "Digital Video Broadcasting (DVB); Specification for the carriage of Vertical Blanking Information (VBI) data in DVB bitstreams".
- [7] ETSI EN 301 790: "Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems".
- [8] ETSI EN 302 307-1: "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 1: DVB-S2".
- [9] ETSI TS 101 154: "Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream".
- [10] ETSI TS 102 005: "Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in DVB services delivered directly over IP protocols".

- [11] ETSI TS 102 006: "Digital Video Broadcasting (DVB); Specification for System Software Update in DVB Systems".
- [12] ETSI TS 102 323: "Digital Video Broadcasting (DVB); Carriage and signalling of TV-Anytime information in DVB transport streams".
- [13] ETSI TS 102 812: "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1.1".
- [14] ISO/IEC 10646: "Information technology - Universal Coded Character Set (UCS)".
- [15] ISO/IEC 13818-1: "Information technology - Generic coding of moving pictures and associated audio information: Systems".
- [16] ISO/IEC 6937: "Information technology - Coded graphic character set for text communication - Latin alphabet".
- [17] ISO/IEC 8859-1: "Information technology - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No. 1".
- [18] ISO/IEC 8859-2: "Information technology - 8-bit single-byte coded graphic character sets - Part 2: Latin alphabet No. 2".
- [19] ISO/IEC 8859-3: "Information technology - 8-bit single-byte coded graphic character sets - Part 3: Latin alphabet No. 3".
- [20] ISO/IEC 8859-4: "Information technology - 8-bit single-byte coded graphic character sets - Part 4: Latin alphabet No. 4".
- [21] ISO/IEC 8859-5: "Information technology - 8-bit single-byte coded graphic character sets - Part 5: Latin/Cyrillic alphabet".
- [22] ISO/IEC 8859-6: "Information technology - 8-bit single-byte coded graphic character sets - Part 6: Latin/Arabic alphabet".
- [23] ISO/IEC 8859-7: "Information technology - 8-bit single-byte coded graphic character sets - Part 7: Latin/Greek alphabet".
- [24] ISO/IEC 8859-8: "Information technology - 8-bit single-byte coded graphic character sets - Part 8: Latin/Hebrew alphabet".
- [25] ISO/IEC 8859-9: "Information technology - 8-bit single-byte coded graphic character sets - Part 9: Latin alphabet No. 5".
- [26] ISO/IEC 8859-10: "Information technology - 8-bit single-byte coded graphic character sets - Part 10: Latin alphabet No. 6".
- [27] ISO/IEC 8859-11: "Information technology - 8-bit single-byte coded graphic character sets - Part 11: Latin/Thai alphabet".
- [28] ISO/IEC 8859-13: "Information technology - 8-bit single-byte coded graphic character sets - Part 13: Latin alphabet No. 7".
- [29] ISO/IEC 8859-14: "Information technology - 8-bit single-byte coded graphic character sets - Part 14: Latin alphabet No. 8 (Celtic)".
- [30] ISO/IEC 8859-15: "Information technology - 8-bit single-byte coded graphic character sets - Part 15: Latin alphabet No. 9".
- [31] CENELEC EN 50221: "Common interface specification for conditional access and other digital video broadcasting decoder applications".
- [32] IEC 61883 (parts 1 and 4): "Consumer audio/video equipment - Digital interface".
- [33] IEEE 1394.1: "IEEE Standard for High Performance Serial Bus Bridges".

- [34] ISO 8601: "Data elements and interchange formats - Information interchange - Representation of dates and times".
- [35] ISO 3166 (all parts): "Codes for the representation of names of countries and their subdivisions".
- [36] ISO 639-2: "Codes for the representation of names of languages - Part 2: Alpha-3 code".
- [37] KSX1001: "Code for Information Interchange (Hangeul and Hanja)", Korean Agency for Technology and Standards, Ref. No. KSX 1001-2004.

NOTE: Available at <http://unicode.org/Public/MAPPINGS/OBSOLETE/EASTASIA/KSC/KSX1001.TXT>.

- [38] ETSI ES 201 812: "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.0.3".
- [39] ETSI TS 102 825 (parts 1 to 5, 7, 9 and 10): "Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM)".
- [40] ETSI EN 302 755: "Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)".
- [41] ETSI TS 102 770: "Digital Video Broadcasting (DVB); System Renewability Messages (SRM) in DVB Systems".
- [42] ETSI EN 302 583: "Digital Video Broadcasting (DVB); Framing Structure, channel coding and modulation for Satellite Services to Handheld devices (SH) below 3 GHz".
- [43] ETSI TS 102 772: "Digital Video Broadcasting (DVB); Specification of Multi-Protocol Encapsulation - inter-burst Forward Error Correction (MPE-iFEC)".
- [44] IETF RFC 2045: "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies".
- [45] ETSI EN 302 769: "Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital transmission system for cable systems (DVB-C2)".
- [46] ETSI TS 101 547-2: "Digital Video Broadcasting (DVB); Plano-stereoscopic 3DTV; Part 2: Frame Compatible Plano-stereoscopic 3DTV".
- [47] ETSI TS 101 547-3: "Digital Video Broadcasting (DVB); Plano-stereoscopic 3DTV; Part 3: HDTV Service Compatible Plano-stereoscopic 3DTV".
- [48] ETSI TS 102 809: "Digital Video Broadcasting (DVB); Signalling and carriage of interactive applications and services in Hybrid Broadcast/Broadband environments".
- [49] ETSI TS 102 773: "Digital Video Broadcasting (DVB); Modulator Interface (T2-MI) for a second generation digital terrestrial television broadcasting system (DVB-T2)".
- [50] GB-2312-1980: "Code of Chinese graphic character set for information interchange, primary set".
- [51] IETF RFC 3986: "Uniform Resource Identifiers (URI): Generic Syntax".
- [52] ETSI TS 102 034: "Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks".
- [53] ETSI TS 101 547-4: "Digital Video Broadcasting (DVB); Plano-stereoscopic 3DTV; Part 4: Service frame compatible Plano-stereoscopic 3DTV for HEVC coded services".
- [54] ETSI EN 302 307-2: "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 2: DVB-S2 Extensions (DVB-S2X)".
- [55] ETSI EN 300 743: "Digital Video Broadcasting (DVB); Subtitling systems".

NOTE: This is also available as DVB BlueBook A156.

- [56] DVB BlueBook A174: "Digital Video Broadcasting (DVB); TTML subtitling systems".

- [57] ETSI TS 101 162: "Digital Video Broadcasting (DVB); Allocation of identifiers and codes for Digital Video Broadcasting (DVB) systems".

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] ETSI TS 101 211: "Digital Video Broadcasting (DVB); Guidelines on implementation and usage of Service Information (SI)".
- [i.2] ETSI TS 102 727: "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.2.2".
- [i.3] ETSI TR 102 825 (parts 6, 8, 11 to 13): "Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM)".
- [i.4] ETSI TS 102 201: "Digital Video Broadcasting (DVB); Interfaces for DVB Integrated Receiver Decoder (DVB-IRD)".
- [i.5] ETSI EN 300 429: "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for cable systems".
- [i.6] ETSI EN 300 421: "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services".
- [i.7] ETSI EN 300 744: "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television".
- [i.8] ETSI TS 103 205: "Digital Video Broadcasting (DVB); Extensions to the CI Plus Specification".
- [i.9] ETSI TS 103 286-2: "Digital Video Broadcasting (DVB); Companion Screens and Streams; Part 2: Content Identification and Media Synchronisation".

3 Definitions and abbreviations

3.1 Definitions

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

AC-3: refers to the coding of audio using the Dolby AC-3 audio compression method as defined in clause 6.2 of TS 101 154 [9]

NOTE: The Service Information requirements for AC-3 streams carried in DVB systems are described in annex D. The carriage of AC-3 elementary streams as private data within MPEG systems is described in clause 6.2 of TS 101 154 [9].

AC-4: refers to the coding of audio using the Dolby AC-4 audio compression method as defined in clauses 6.6 and 6.7 of TS 101 154 [9].

NOTE: The Service Information requirements for AC-4 streams carried in DVB systems are described in annex D. The carriage of AC-4 elementary streams as private data within MPEG systems is described in clauses 6.6 and 6.7 of TS 101 154 [9].

audio preselection: set of audio programme components representing a version of the audio programme that may be selected by a user for simultaneous decoding

NOTE: An audio preselection is a sub-selection from all available audio programme components of one audio programme. An audio preselection can be considered the NGA equivalent of audio services in predecessor systems, which each utilized complete mixes

audio programme: complete collection of all audio programme components and a set of accompanying audio preselections

NOTE: Not all audio programme components of an audio programme are necessarily meant to be presented at the same time. An audio programme may contain audio programme components that are always presented, and it may include optional audio programme components.

audio programme component: the smallest addressable unit of an audio programme

auxiliary NGA stream: NGA stream delivered using NGA multi-stream delivery, and containing additional audio programme components not contained in the main NGA stream

bouquet: collection of services marketed as a single entity

broadcaster (SERVICE Provider): organization which assembles a sequence of events or data streams to be delivered to the viewer; the delivery can be based upon a schedule

cell: geographical area that is covered with DVB signals delivering one or more particular transport or other DVB streams throughout the area by means of one or more transmitters

NOTE: The cell may in addition contain repeaters. Two neighbouring cells may be intersecting, or fully overlapping. The cell_id that is used to uniquely identify a cell is unique within each original_network_id. For hand-over purposes it is more convenient if the transport streams associated with the cell cover exactly the same area, or only one transport stream per cell is used.

component (ELEMENTARY Stream): one or more entities which together make up an event

EXAMPLE: Video, audio, teletext.

Conditional Access (CA) system: system to control subscriber access to services, data streams and events

EXAMPLE: Videoguard, Eurocrypt.

delivery system: physical medium by which one or more DVB transport streams are transmitted

EXAMPLE: Satellite system, wide-band coaxial cable, fibre optics, terrestrial channel of one emitting point.

dependent stream: a stream or component which relies on another stream or component in order to be rendered as intended

EXAMPLE: A dependent view of a 3D video, or a receiver-mix audio description stream.

DTS®: refers to the first generation audio coding system by DTS, Inc. as defined in clause 6.3 of TS 101 154 [9].

NOTE: The service information requirements for DTS® are found in Annex G.

DTS-HD®: refers to the enhanced DTS® audio coding system by DTS, Inc. as defined in clause 6.3 of TS 101 154 [9]. Note that DTS-HD® is a superset of DTS®.

NOTE: The service information requirements for DTS-HD® are found in Annex G.

DVB transport stream: an MPEG-2 transport stream [15] containing the mandatory DVB-SI signalling as defined in the present document

NOTE: It is recommended that the DVB-SI implementation specification TS 101 211 [i.1] is additionally followed. It defines further requirements for the signalling to help improve the quality of experience for viewers.

event: grouping of elementary broadcast data streams with a defined start time and duration belonging to a common service

EXAMPLE: First half of a football match, News Flash, first part of an entertainment show.

main NGA stream: NGA stream delivered using NGA multi-stream delivery, and containing at least all the audio programme components corresponding to at least one audio preselection

MPEG-2: See ISO/IEC 13818-1 [15].

MPEG-H audio: refers to the coding of audio using the audio compression method as defined in clause 6.8 of TS 101 154 [9].

multi-stream delivery: method for carrying audio programme components in several NGA streams

NOTE: e.g. when audio programme components offering additional languages are carried in separate elementary streams to facilitate remultiplexing or service aggregation

network: managed and navigable collection of DVB transport streams transmitted on one or more delivery systems generally based on the same physical medium

NOTE 1: It is possible to operate both, first and second generation delivery systems in the same network (e.g. DVB-T and DVB-T2).

NOTE 2: A network is identified by its network_id. It might be composed of one or more emitting sites.

n PSK: n-valued Phase Shift Keying (other than quaternary)

NGA stream: audio elementary stream containing one or more audio programme components of one audio programme

original_network_id: unique identifier of a broadcast platform operator

NOTE: This value is assigned by DVB.

repeater: special type of transmitter which receives a terrestrial DVB signal and re-transmits it unchanged

NOTE: Hence it does not support changing of the cell_id.

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

NOTE: Unless otherwise specified within the present document all "reserved" bits are set to "1".

reserved_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: Unless otherwise specified within the present document all "reserved_future_use" bits are set to "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".

section: syntactic structure used for mapping all service information defined in the present document into ISO/IEC 13818-1 [15] TS packets

service: a grouping (usually defined by a PMT) of one or more data streams which are offered as a whole to the user

service_id: unique identifier of a service within a DVB transport stream

NOTE: In areas where TS 101 211 [i.1] is applicable in addition to the present document, a service_id is also unique within the scope of an original_network_id.

Service Information (SI): describes the delivery system, and the content and scheduling/timing of services and events

NOTE: It includes MPEG-2 Program Specific Information (PSI) together with DVB-defined extensions.

single-stream delivery: method for carrying all audio programme components in a single NGA stream

subcell: geographical area that is part of the cell's coverage area and that is covered with DVB signals by means of a transposer

NOTE: In conjunction with the cell_id the cell_id_extension is used to uniquely identify a subcell.

sub_table: collection of sections with the same value of table_id and:

- generally: the same table_id_extension and version_number;

NOTE: In the case of a NIT, the table_id_extension field conveys the network_id.

NOTE: In the case of a BAT, the table_id_extension field conveys the bouquet_id.

- for a SDT: the same table_id_extension (transport_stream_id), the same original_network_id and version_number;

for a EIT: the same table_id_extension (service_id), the same transport_stream_id, the same original_network_id and version_number. NOTE: The table_id_extension field is equivalent to the fourth and fifth byte of a section when the section_syntax_indicator is set to a value of "1".

table: comprised of a number of sub_tables with the same value of table_id

transmitter: equipment that emits a modulated DVB stream

Transport Stream (TS): data structure defined in ISO/IEC 13818-1 [15]

NOTE: It is the basis of the DVB standards.

transport_stream_id: unique identifier of a TS within an original network

transposer: special type of repeater which receives a terrestrial DVB signal and re-transmits it on a different frequency

The relationships of some of these definitions are illustrated in the service delivery model in figure 1.

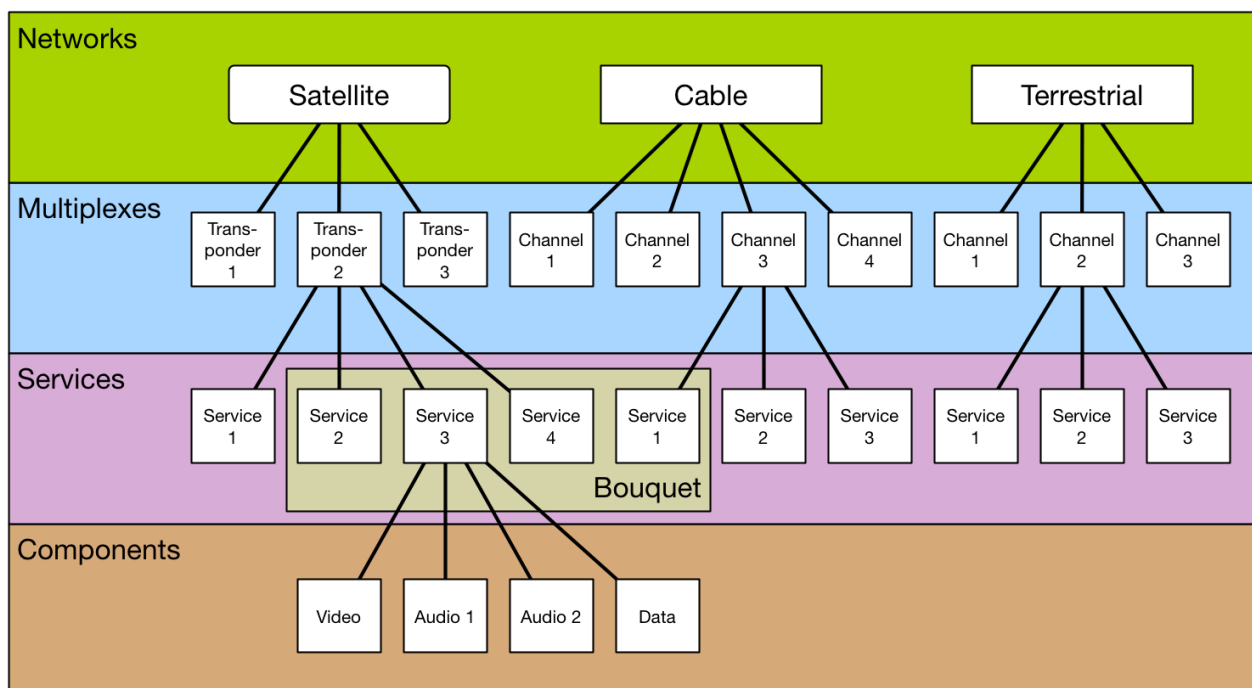


Figure 1: Digital broadcasting, service delivery model

3.2 Abbreviations

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

3DTV Three Dimensional Television
AAC Advanced Audio Coding

NOTE: See TS 101 154 [9].

AC-3 AC-3 audio coding

NOTE: See TS 101 154 [9].

AC-4 AC-4 audio coding

NOTE: See TS 101 154 [9].

ASCII American Standard Code for Information Interchange

ATIS Alliance for Telecommunications Industry Solutions

AVC Advanced Video Coding

NOTE: See TS 101 154 [9].

BAT Bouquet Association Table

BCD Binary Coded Decimal

BMP Basic Multilingual Plane

NOTE: See ISO/IEC 10646 [14].

bslbf bit string, left bit first

NOTE: See clause 5.1.6.

CA Conditional Access

CAT Conditional Access Table

CATV Cable Television

CIT Content Identifier Table

NOTE: See TS 102 323 [12].

CLUT Colour Look-Up Table

CP Content Protection

CPCM Content Protection Copy Management

CR Carriage Return character

CRC Cyclic Redundancy Check

CR/LF Carriage Return/Line Feed

DAB Digital Audio Broadcasting

DIT Discontinuity Information Table

DSNG Digital Satellite News Gathering

DTCP Digital Transmission Content Protection

DTS® DTS® audio coding

DTS-HD® DTS-HD® audio coding

NOTE: See TS 101 154 [9]

DVB Digital Video Broadcasting

DVB-C DVB Cable Framing and Modulation

NOTE: See EN 300 429 [i.5].

DVB-S DVB Satellite Framing and Modulation

NOTE: See EN 300 421 [i.6].

DVB-SH DVB Satellite to Handheld Framing and Modulation

NOTE: See EN 302 583.

DVB-T DVB Terrestrial Framing and Modulation

NOTE: See EN 300 744 [i.7].

DVD Digital Versatile Disc

EBU European Broadcasting Union

ECM Entitlement Control Message

NOTE: See ISO/IEC 13818-1 [15].

EIT Event Information Table

EMM Entitlement Management Message

NOTE: See ISO/IEC 13818-1 [15].

EPG Electronic Programme Guide

ES Elementary Stream

FC Frame Compatible

FEC Forward Error Correction

FFT Fast Fourier Transform

FLS Forward Link Signalling

NOTE: See EN 301 790 [7].

FM Frequency Modulation

FTA Free-To-Air

HD High Definition (Video)

HDCP High-bandwidth Digital Content Protection

HDR High Dynamic Range

HE AAC High Efficiency AAC

HE High Efficiency

HEVC High Efficiency Video Coding

HEVC_UHDTV_IRD: The initial ultra high definition IRD profile defined in TS 101 154 [9]

HFR High Frame Rate

HLG10: Hybrid Log Gamma with a bit depth of 10 bits

HP High Priority

IEC International Electrotechnical Commission

INT IP Notification Table

NOTE: See EN 301 192 [4].

IP Internet Protocol

IP/MAC Internet Protocol/Medium Access Control

IRD Integrated Receiver Decoder

ISI Input Stream Identifier

NOTE: See EN 302 307-1 [8].

ISO International Organization for Standardization

LC Low Complexity

LF Line Feed character

LP Low Priority

MHP Multimedia Home Platform

MI Modulator Interface

NOTE: See TS 102 773 [49].

MISO Multiple Input, Single Output

MJD Modified Julian Date

MPE Multiprotocol Encapsulation

MPEG Moving Pictures Expert Group

MPEG-H MPEG-H audio coding

NOTE: See TS 101 154 [9].

MSB Most Significant Bit

MVC Multi-View video Coding

NOTE: See ISO/IEC 14496-10.

NDA Non Disclosure Agreement

| | |
|-------|--|
| NGA | Next Generation Audio |
| NIT | Network Information Table |
| NVOD | Near Video On Demand |
| OFDM | Orthogonal Frequency Division Multiplex |
| PAT | Program Association Table |
| paTS | Partially available Transport Stream (DVB-SH) |
| PCM | Pulse-code modulation |
| PDC | Programme Delivery Control |
| PES | Packetised Elementary Stream |
| PID | Packet IDentifier |
| PIL | Programme Identification Label |
| PLP | Physical Layer Pipe |
| PMT | Program Map Table |
| PNG | Portable Network Graphics |
| PQ10: | Perceptual Quantizer with a bit depth of 10 bits |
| PSI | Program Specific Information |
| PSTN | Public Switched Telephone Network |
| QAM | Quadrature Amplitude Modulation |
| QPSK | Quaternary Phase Shift Keying |
| RAM | Random Access Memory |
| RAR | Resolution Authority Record |
| RCS | Return Channel Satellite |

NOTE: See EN 301 790 [7].

| | |
|--------|--|
| RDS | Radio Data System |
| RFC | Request For Comments |
| RNT | RAR Notification Table |
| rpchof | remainder polynomial coefficients, highest order first |

NOTE: See clause 5.1.6.

| | |
|---------|--|
| RS | Reed-Solomon |
| RST | Running Status Table |
| SAOC-DE | Spatial Audio Object Coding - Dialogue Enhancement |
| SC | Service Compatible |
| ScF | Scale Factor |
| SD | Standard Definition (Video) |
| SDR | Standard Dynamic Range |
| SDT | Service Description Table |
| SFN | Single Frequency Network |
| SI | Service Information |
| SISO | Single Input, Single Output |
| SIT | Selection Information Table |
| SMATV | Satellite Master Antenna Television |
| SMI | Storage Media Interoperability |
| SRM | System Renewability Message |
| SSU | System Software Update |

NOTE: See EN 301 192 [4].

| | |
|---------|---|
| ST | Stuffing Table |
| tcimsbf | Two's complement integer, most significant (sign) bit first |

NOTE: See clause 5.1.6.

| | |
|------|------------------------------------|
| TDM | Time-Domain Multiplex |
| TDT | Time and Date Table |
| TFS | Time Frequency Slicing (DVB-T2) |
| TOT | Time Offset Table |
| TPS | Transmission Parameter Signalling |
| TS | Transport Stream |
| TSDT | Transport Stream Description Table |

TTML Timed Text Markup Language

NOTE: see <https://www.w3.org/TR/ttml-profile-registry/>

TTS Text-to-Speech

TVA TV-Anytime

NOTE: See <http://www.tv-anytime.org>.

UCS Universal Character Set

UECP Universal Encoder Communication Protocol (RDS)

UHD Ultra High Definition (Video)

uimsbf unsigned integer most significant bit first

NOTE: See clause 5.1.6.

URI Uniform Resource Identifier

URL Uniform Resource Locator

USI Usage State Information

UTC Universal Time, Co-ordinated

UTF-8 UCS Transformation Format, 8-bit

NOTE: See ISO/IEC 10646 [14].

VBI Vertical Blanking Interval

VC-1 advanced Video Coding

NOTE: See TS 101 154 [9].

VPS Video Programme System

WSS Wide Screen Signalling

XAIT XML encoding for AIT

NOTE: See TS 102 727 [i.2].

4 Service Information (SI) description

ISO/IEC 13818-1 [15] specifies SI which is referred to as PSI. The PSI data provides information to enable automatic configuration of the receiver to demultiplex and decode the various streams of programs within the multiplex.

The PSI data is structured as four types of table. The tables are transmitted in sections.

1) Program Association Table (PAT):

- for each service in the multiplex, the PAT indicates the location (the Packet Identifier (PID) values of the Transport Stream (TS) packets) of the corresponding Program Map Table (PMT). It also gives the location of the Network Information Table (NIT).

2) Conditional Access Table (CAT):

- the CAT provides information on the CA systems used in the multiplex; the information is private (not defined within the present document) and dependent on the CA system, but includes the location of the EMM stream, when applicable.

3) Program Map Table (PMT):

- the PMT identifies and indicates the locations of the streams that make up each service and the location of the Program Clock Reference fields for a service.

4) Network Information Table (NIT):

- the location of the NIT is defined in the present document in compliance with ISO/IEC 13818-1 [15] specification, but the data format is outside the scope of ISO/IEC 13818-1 [15]. It is intended to provide information about the physical network. The syntax and semantics of the NIT are defined in the present document.

In addition to the PSI, data is needed to provide identification of services and events for the user. The coding of this data is defined in the present document. In contrast with the PAT, CAT, and PMT of the PSI, which give information only for the multiplex in which they are contained (the actual multiplex), the additional information defined within the present document can also provide information on services and events carried by different multiplexes, and even on other networks. This data is structured as nine tables:

- 1) Bouquet Association Table (BAT):
 - the BAT provides information regarding bouquets. As well as giving the name of the bouquet, it provides a list of services for each bouquet.
- 2) Service Description Table (SDT):
 - the SDT contains data describing the services in the system e.g. names of services, the service provider, etc.
- 3) Event Information Table (EIT):
 - the EIT contains data concerning events or programmes such as event name, start time, duration, etc.;
 - the use of different descriptors allows the transmission of different kinds of event information e.g. for different service types.
- 4) Running Status Table (RST):
 - the RST gives the status of an event (running/not running). The RST updates this information and allows timely automatic switching to events.
- 5) Time and Date Table (TDT):
 - the TDT gives information relating to the present time and date. This information is given in a separate table due to the frequent updating of this information.
- 6) Time Offset Table (TOT):
 - the TOT gives information relating to the present time and date and local time offset. This information is given in a separate table due to the frequent updating of the time information.
- 7) Stuffing Table (ST):
 - the ST is used to invalidate existing sections, for example at delivery system boundaries.
- 8) Selection Information Table (SIT):
 - the SIT is used only in "partial" (i.e. recorded) bitstreams. It carries a summary of the SI information required to describe the streams in the partial bitstream.
- 9) Discontinuity Information Table (DIT):
 - the DIT is used only in "partial" (i.e. recorded) bitstreams. It is inserted where the SI information in the partial bitstream may be discontinuous.

Where applicable the use of descriptors allows a flexible approach to the organization of the tables and allows for future compatible extensions.

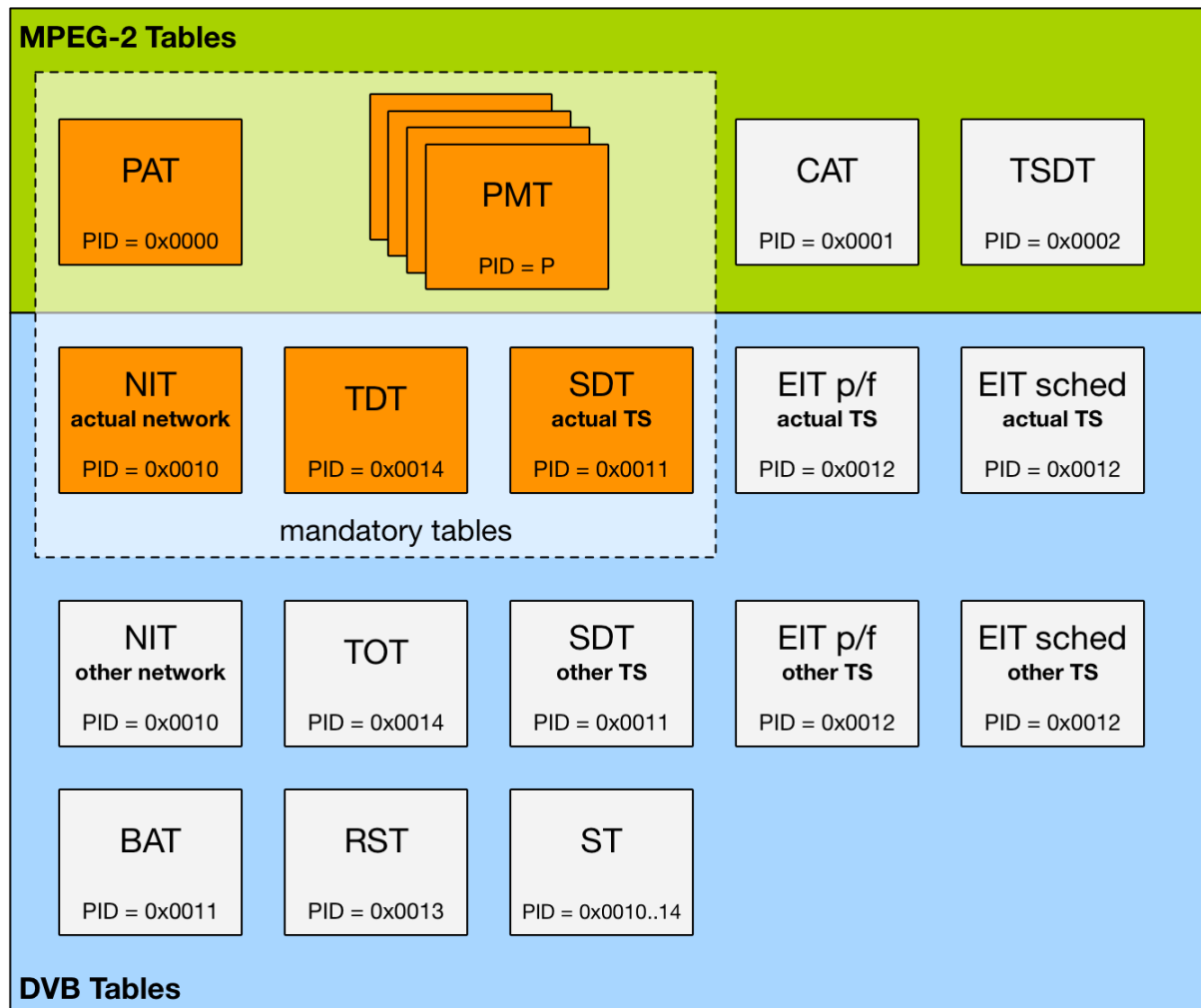


Figure 2: General organization of the Service Information (SI)

5 Service Information (SI) tables

5.1 SI table mechanism

5.1.1 Use of table sections

The SI specified in the present document and MPEG-2 PSI tables shall be segmented into one or more sections before being inserted into TS packets.

The tables listed in clause 4 are conceptual in that they need never be regenerated in a specified form within an IRD. The tables, when transmitted shall not be scrambled, with the exception of the EIT, which may be scrambled if required (see clause 5.1.5).

A section is a syntactic structure that shall be used for mapping all MPEG-2 tables and SI tables specified in the present document, into TS packets.

These SI syntactic structures conform to the private section syntax defined in ISO/IEC 13818-1 [15].

Sections may be variable in length. The sections within each table are limited to 1 024 bytes in length, except for sections within the EIT which are limited to 4 096 bytes. Each section is uniquely identified by the combination of the following elements:

- a) table_id:

- The `table_id` identifies to which table the section belongs.
 - Some `table_ids` have been defined by ISO and others by ETSI. Other values of the `table_id` can be allocated by the user for private purposes. The list of values of `table_id` is contained in table 2.
- b) `table_id_extension`:
- The `table_id_extension` is used for identification of a `sub_table`.
 - The interpretation of each `sub_table` is given in clause 5.2.
- c) `section_number`:
- The `section_number` field allows the sections of a particular `sub_table` to be reassembled in their original order by the decoder. It is recommended, that sections are transmitted in numerical order, unless it is desired to transmit some sections of the `sub_table` more frequently than others, e.g. due to random access considerations.
 - For the SI tables as specified in the present document, section numbering applies to `sub_tables`.
- d) `version_number`:
- When the characteristics of the TS described in the SI given in the present document change (e.g. new events start, different composition of elementary streams for a given service), then new SI data shall be sent containing the updated information. A new version of the SI data is signalled by sending a `sub_table` with the same identifiers as the previous `sub_table` containing the relevant data, but with the next value of `version_number`.
 - For the SI tables specified in the present document, the `version_number` applies to all sections of a `sub_table`.
- e) `Current_next_indicator`:
- Each section shall be numbered as valid "now" (current), or as valid in the immediate future (next). This allows the transmission of a future version of the SI in advance of the change, giving the decoder the opportunity to prepare for the change. There is however, no requirement to transmit the next version of a section in advance, but if it is transmitted, then it shall be the next correct version of that section.

5.1.2 Mapping of sections into Transport Stream (TS) packets

Sections shall be mapped directly into TS packets. Sections may start at the beginning of the payload of a TS packet, but this is not a requirement, because the start of the first section in the payload of a TS packet is pointed to by the `pointer_field`. There is never more than one `pointer_field` in a TS packet, as the start of any other section can be identified by counting the length of the first and any subsequent sections, since no gaps between sections within a TS packet are allowed by the syntax.

Within TS packets of any single PID value, one section is finished before the next one is allowed to be started, or else it is not possible to identify to which section header the data belongs. If a section finishes before the end of a TS packet, but it is not convenient to open another section, a stuffing mechanism may be used to fill up the space.

Stuffing may be performed by filling each remaining byte of the TS packet with the value "0xFF". Consequently the value "0xFF" shall not be used for the `table_id`. If the byte immediately following the last byte of a section takes the value of "0xFF", then the rest of the TS packet shall be stuffed with "0xFF" bytes. These bytes may be discarded by a decoder. Stuffing may also be performed using the `adaptation_field` mechanism.

For a more detailed description of the mechanism and functionality, specifically refer to clause 2.4.4 and annex C of ISO/IEC 13818-1 [15].

5.1.3 Coding of PID and `table_id` fields

Table 1 lists the PID values which shall be used for the TS packets which carry SI sections.

Table 1: PID allocation for SI

| Table | PID value |
|--------------------------------|------------------|
| PAT | 0x0000 |
| CAT | 0x0001 |
| TSDT | 0x0002 |
| reserved | 0x0003 to 0x000F |
| NIT, ST | 0x0010 |
| SDT, BAT, ST | 0x0011 |
| EIT, ST, CIT (TS 102 323 [12]) | 0x0012 |
| RST, ST | 0x0013 |
| TDT, TOT, ST | 0x0014 |
| network synchronization | 0x0015 |
| RNT (TS 102 323 [12]) | 0x0016 |
| reserved for future use | 0x0017 to 0x001B |
| link-local inband signalling | 0x001C |
| measurement | 0x001D |
| DIT | 0x001E |
| SIT | 0x001F |

The PID value 0x001C allocated to link-local inband signalling shall not be used on any broadcast signals. It shall only be used between devices in a controlled environment.

NOTE: The PID 0x001C can for example be used within a broadcast centre, between a receiver device and a CAM, or on private satellite links.

Table 2 lists the values which shall be used for table_id for the service information, defined in the present document.

Table 2: Allocation of table_id values

| Value | Description |
|--------------|--|
| 0x00 | program_association_section |
| 0x01 | conditional_access_section |
| 0x02 | program_map_section |
| 0x03 | transport_stream_description_section |
| 0x04 to 0x3F | reserved |
| 0x40 | network_information_section - actual_network |
| 0x41 | network_information_section - other_network |
| 0x42 | service_description_section - actual_transport_stream |
| 0x43 to 0x45 | reserved for future use |
| 0x46 | service_description_section - other_transport_stream |
| 0x47 to 0x49 | reserved for future use |
| 0x4A | bouquet_association_section |
| 0x4B | update notification table section (TS 102 006 [11]) |
| 0x4C | downloadable font info section (DVB BlueBook A174 [56]) |
| 0x4D | reserved for future use |
| 0x4E | event_information_section - actual_transport_stream, present/following |
| 0x4F | event_information_section - other_transport_stream, present/following |
| 0x50 to 0x5F | event_information_section - actual_transport_stream, schedule |
| 0x60 to 0x6F | event_information_section - other_transport_stream, schedule |
| 0x70 | time_date_section |
| 0x71 | running_status_section |
| 0x72 | stuffing_section |
| 0x73 | time_offset_section |
| 0x74 | application information section (TS 102 812 [13]) |
| 0x75 | container section (TS 102 323 [12]) |
| 0x76 | related content section (TS 102 323 [12]) |
| 0x77 | content identifier section (TS 102 323 [12]) |
| 0x78 | MPE-FEC section (EN 301 192 [4]) |
| 0x79 | resolution provider notification section (TS 102 323 [12]) |
| 0x7A | MPE-IFEC section (TS 102 772 [43]) |
| 0x7B | protection message section (TS 102 809 [48]) |

| Value | Description |
|--|-----------------------------------|
| 0x7C to 0x7D | reserved for future use |
| 0x7E | discontinuity_information_section |
| 0x7F | selection_information_section |
| 0x80 to 0xFE | user defined |
| 0xFF | reserved (see note) |
| NOTE: see annex C.4 in ISO/IEC 13818-1 [15] for why table_id of 0xFF can never be used | |

5.1.4 Repetition rates and random access

In systems where acquisition time of PSI and SI in DVB Transport Streams is important, it is recommended to continuously re-transmit these sections at regular intervals, even when no changes occur. Clause 4.4 of TS 101 211 [i.1] makes recommendations for how often PSI and SI sections should be re-transmitted.

For SI specified within the present document the minimum time interval between the arrival of the last byte of a section to the first byte of the next transmitted section with the same PID, table_id and table_id_extension and with the same or different section_number shall be 25 ms. This limit applies for TSs with a total data rate of up to 100 Mbit/s.

5.1.5 Scrambling

With the exception of the EIT carrying schedule information, all tables specified in the present document shall not be scrambled. One method for scrambling the EIT schedule table is given in annex E (Usage of the Scrambling_descriptor). If a scrambling method operating over TS packets is used, it may be necessary to use a stuffing mechanism to fill from the end of a section to the end of a packet so that any transitions between scrambled and unscrambled data occur at packet boundaries.

In order to identify the CA streams which control the descrambling of the EIT data, a scrambled EIT schedule table shall be identified in the PSI. Service_id value 0xFFFF is allocated to identifying a scrambled EIT, and the program map section for this service shall describe the EIT as a private stream and shall include one or more CA_descriptors (defined in ISO/IEC 13818-1 [15]) which give the PID values and optionally, other private data to identify the associated CA streams. Service_id value 0xFFFF shall not be used for any other service.

5.1.6 Bit order and transmission order

The present clause defines the bit order for encoding unsigned integer (uimbsf), bit string (bslbf), two's complement integer (tcimbsf), and remainder polynomial coefficients (rpchof) fields in tables (see clause 5) and descriptors (see clause 6), as well as the order in which these fields are to be transmitted. The present clause also explains and illustrates these provisions.

Within fields, the convention for assigning bit positions – the bit order – follows the big-endian convention, i.e. more significant bits are assigned positions with higher index values. This results in the most significant bit (msb) being assigned the highest index position, and the least significant bit (lsb) being assigned the lowest index position within a field. Since, by convention, the indices of bit positions count upwards starting from zero, the least significant bit appears at index position zero, denoted as "b₀". Consequently, when the field is N bits wide, the most significant bit appears at index position N-1. These principles equally apply to all types of fields. Some example fields are shown in figure 1.

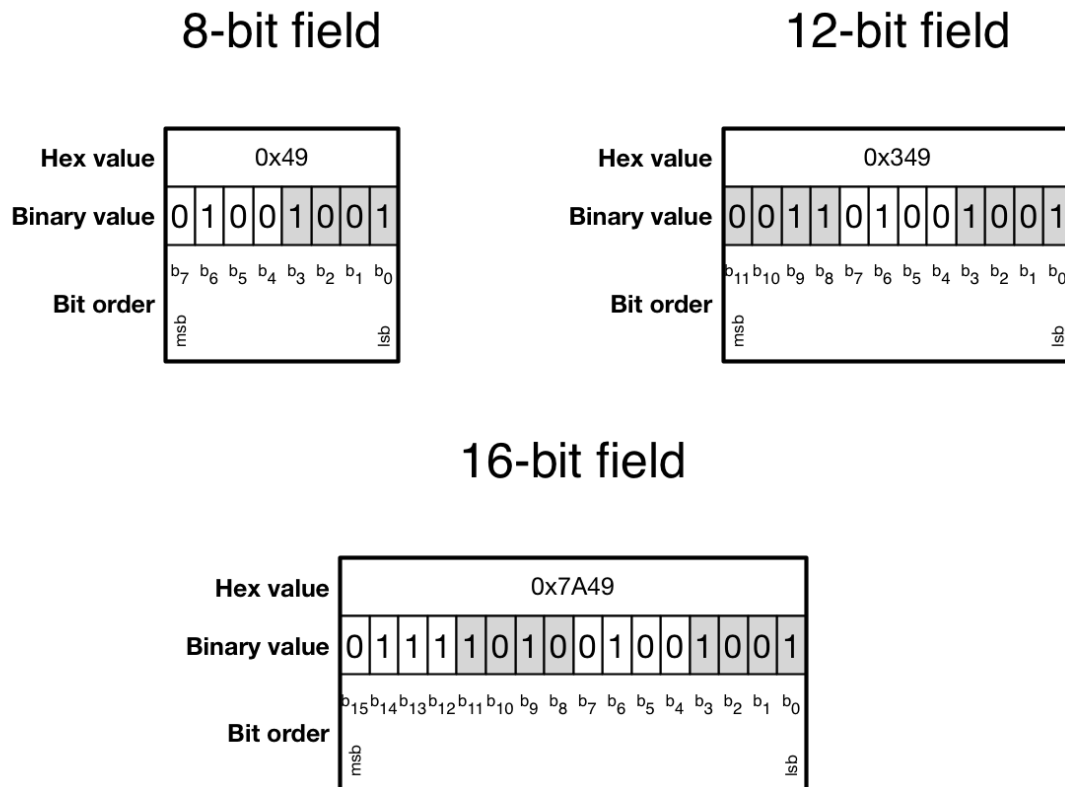


Figure 1: Examples of bit order within fields

To define useful elements for conveying service information, the present document defines higher level data structures, like e.g. tables and descriptors, by concatenating various fields of given width and semantics. When such a data structure is to be conveyed to IRDs, it is transmitted over a broadcast link using one of the DVB physical layer standards (e.g. DVB-T, DVB-S, or DVB-C). Since all DVB physical layers are bit pipes, any data needs to be serialised into a bit stream before transmission. This serialisation follows two rules:

- a) The fields comprising a data structure shall be transmitted in the order they appear in the structure's syntax definition, top to bottom.
- b) Within each field, the bits shall be transmitted in descending order of index position.

An example for applying these rules is shown in figure 2. Part (a) of figure 2 illustrates rule a), and shows the syntax definition for the country availability descriptor (see clause 6.2.10). The red arrow to the left of the syntax definition table indicates the top to bottom sequence of transmission of the fields. Part (b) of figure 2 illustrates rule b), and shows how the bits comprising the fields are serialised. Again, the red arrow below the serialisation indicates the sequence of transmission in descending order of index position. For reference, some example values for the fields have been chosen, and their hexadecimal representation is also given.

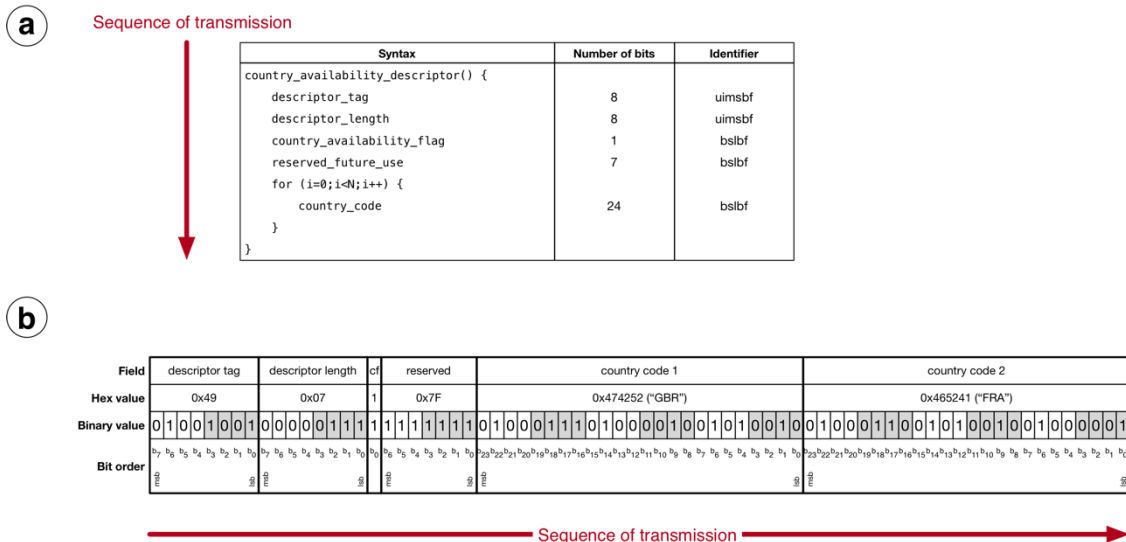


Figure 2: Example of transmission order

Based on these rules, the names "unsigned integer, most significant bit first" (uimsbf) for unsigned integer fields, and "bit string, left bit first" (bslbf) for bit string fields can be conveniently explained.

Unsigned integer fields represent numbers within a polyadic number system, which are composed of digits and where the position of each digit indicates to which power of the number system's base the digit refers. Due to the use of the big-endian convention for the bit order within each field, the most significant bit (msb) of an N-bit unsigned integer field is assigned the highest possible index N-1. Following rule b) (transmission in descending order of index position), the most significant bit of a field is transmitted first, and the least significant bit is transmitted last. Hence the name of "unsigned integer, most significant bit first" (uimsbf).

Bit string fields – as opposed to unsigned integer fields – do not represent numbers, and hence their bit index positions are not associated with higher or lower significance of the bits, but only serve to uniquely identify the bits within a field. To still allow unambiguous serialisation and deserialisation of bit string fields, the provisions of the present document use the bit index positions b₀, b₁, b₂, etc. to define the encoding and semantics of each bit in a bit string field. When all bits of a bit string field are rendered in printing according to the big-endian convention, the bit with the highest index position N-1 will appear on the left, and the bit with the lowest index position zero will appear on the right. Following rule b) (transmission in descending order of index position), the bit with the highest index position (appearing on the left) is transmitted first. Hence the name of "bit string, left bit first" (bslbf).

5.2 Table definitions

5.2.0 Introduction

The following clauses describe the syntax and semantics of the different types of table.

NOTE: The symbols and abbreviations, and the method of describing syntax used in the present document are the same as those defined in clauses 2.2 and 2.3 of ISO/IEC 13818-1 [15].

5.2.1 Network Information Table (NIT)

The NIT (see table 3) conveys information relating to the physical organization of the multiplexes/TSs carried via a given network, and the characteristics of the network itself. The combination of `original_network_id` and `transport_stream_id` allow each TS to be uniquely identified throughout the application area of the present document. Networks are assigned individual `network_id` values, which serve as unique identification codes for networks. The allocation of these codes is specified in TS 101 162 [57]. The `network_id` and the `original_network_id` can take the same value, or may have to take different values subject to the allocation constraints for `original_network_id` and `network_id` as are defined in TS 101 162 [57].

Guidelines for the processing of SI at transitions between delivery system boundaries, e.g. from satellite to cable or SMATV systems, can be found in TS 101 211 [i.1].

IRDs may be able to store the NIT information in non-volatile memory in order to minimize the access time when switching between channels ("channel hopping"). It is also possible to transmit a NIT for other networks in addition to the actual network. Differentiation between the NIT for the actual network and the NIT for other networks is achieved using different `table_id` values (see table 2).

The NIT shall be segmented into `network_information_sections` using the syntax of table 3. Any sections forming part of an NIT shall be transmitted in TS packets with a PID value of 0x0010. Any sections of the NIT which describe the actual network (that is, the network of which the TS containing the NIT is a part) shall have the `table_id` 0x40 with the same `table_id_extension` (`network_id`). The `network_id` field shall take the value assigned to the actual network according to TS 101 162 [57]. Any sections of a NIT which refer to a network other than the actual network shall take a `table_id` value of 0x41 and the `network_id` shall take the value assigned to the other network according to TS 101 162 [57].

Table 3: Network information section

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| <code>network_information_section() {</code> | | |
| <code>table_id</code> | 8 | uimsbf |
| <code>section_syntax_indicator</code> | 1 | bslbf |
| <code>reserved_future_use</code> | 1 | bslbf |
| <code>reserved</code> | 2 | bslbf |
| <code>section_length</code> | 12 | uimsbf |
| <code>network_id</code> | 16 | uimsbf |
| <code>reserved</code> | 2 | bslbf |
| <code>version_number</code> | 5 | uimsbf |
| <code>current_next_indicator</code> | 1 | bslbf |
| <code>section_number</code> | 8 | uimsbf |
| <code>last_section_number</code> | 8 | uimsbf |
| <code>reserved_future_use</code> | 4 | bslbf |
| <code>network_descriptors_length</code> | 12 | uimsbf |
| <code>for (i=0; i<N; i++) {</code> | | |
| <code>descriptor()</code> | | |
| <code>}</code> | | |
| <code>reserved_future_use</code> | 4 | bslbf |
| <code>transport_stream_loop_length</code> | 12 | uimsbf |
| <code>for (i=0; i<N; i++) {</code> | | |
| <code>transport_stream_id</code> | 16 | uimsbf |
| <code>original_network_id</code> | 16 | uimsbf |
| <code>reserved_future_use</code> | 4 | bslbf |
| <code>transport_descriptors_length</code> | 12 | uimsbf |
| <code>for (j=0; j<N; j++) {</code> | | |
| <code>descriptor()</code> | | |
| <code>}</code> | | |
| <code>}</code> | | |
| <code>CRC_32</code> | 32 | rpchof |
| <code>}</code> | | |

Semantics for the network information section:

table_id: See table 2.

section_syntax_indicator: The `section_syntax_indicator` is a 1-bit field which shall be set to "1".

section_length: This is a 12-bit field, the first two bits of which shall be "00". 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 1 021 so that the entire section has a maximum length of 1 024 bytes.

network_id: This is a 16-bit field which serves as a label to identify the delivery system, about which the NIT informs, from any other delivery system. It shall be coded according to TS 101 162 [57].

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 defined by the table_id and network_id. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table defined by the table_id and network_id.

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 table_id and network_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.

network_descriptors_length: This 12-bit field gives the total length in bytes of the following network descriptors.

transport_stream_loop_length: This is a 12-bit field specifying the total length in bytes of the TS loops that follow, ending immediately before the first CRC-32 byte.

transport_stream_id: This is a 16-bit field which serves as a label for identification of this TS from any other multiplex within the delivery system.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

transport_descriptors_length: This is a 12-bit field specifying the total length in bytes of TS descriptors that follow.

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex A of ISO/IEC 13181-1 [15] after processing the entire section.

5.2.2 Bouquet Association Table (BAT)

The BAT (see table 4) provides information regarding bouquets. A bouquet is a collection of services, which may traverse the boundary of a network.

The BAT shall be segmented into bouquet_association_sections using the syntax of table 4. Any sections forming part of a BAT shall be transmitted in TS packets with a PID value of 0x0011. The sections of a BAT sub_table describing a particular bouquet shall have the bouquet_id field taking the value assigned to the bouquet according to TS 101 162 [57]. All BAT sections shall take a table_id value of 0x4A.

Table 4: Bouquet association section

| Syntax | Number of bits | Identifier |
|---------------------------------|----------------|------------|
| bouquet_association_section() { | | |
| table_id | 8 | uimsbf |
| section_syntax_indicator | 1 | bslbf |
| reserved_future_use | 1 | bslbf |
| reserved | 2 | bslbf |
| section_length | 12 | uimsbf |
| bouquet_id | 16 | uimsbf |
| reserved | 2 | bslbf |
| version_number | 5 | uimsbf |
| current_next_indicator | 1 | bslbf |
| section_number | 8 | uimsbf |
| last_section_number | 8 | uimsbf |
| reserved_future_use | 4 | bslbf |
| bouquet_descriptors_length | 12 | uimsbf |

| Syntax | Number of bits | Identifier |
|--|--|--|
| <pre> for (i=0; i<N; i++) { descriptor() } reserved_future_use transport_stream_loop_length for (i=0; i<N; i++) { transport_stream_id original_network_id reserved_future_use transport_descriptors_length for (j=0; j<N; j++) { descriptor() } } CRC_32 </pre> | 4 12 16 16 4 12 32 | bslbf uimsbf uimsbf uimsbf bslbf uimsbf rpchof |

Semantics for the bouquet association section:

table_id: See table 2.

section_syntax_indicator: The section_syntax_indicator is a 1-bit field which shall be set to "1".

section_length: This is a 12-bit field, the first two bits of which shall be "00". 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 1 021 so that the entire section has a maximum length of 1 024 bytes.

bouquet_id: This is a 16-bit field which serves as a label to identify the bouquet. It shall be coded according to TS 101 162 [57].

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 defined by the table_id and bouquet_id. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table defined by the table_id and bouquet_id.

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 table_id and bouquet_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.

bouquet_descriptors_length: This 12-bit field gives the total length in bytes of the following descriptors.

transport_stream_loop_length: This is a 12-bit field specifying the total length in bytes of the TS loop that follows.

transport_stream_id: This is a 16-bit field which serves as a label for identification of this TS from any other multiplex within the delivery system.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

transport_descriptors_length: This is a 12-bit field specifying the total length in bytes of TS descriptors that follow.

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex A of ISO/IEC 13181-1 [15] after processing the entire private section.

5.2.3 Service Description Table (SDT)

Each sub_table of the SDT (see table 5) shall describe services that are contained within a particular TS. The services may be part of the actual TS or part of other TSs, these being identified by means of the table_id (see table 2).

The SDT shall be segmented into service_description_sections using the syntax of table 5. Any sections forming part of an SDT shall be transmitted in TS packets with a PID value of 0x0011. Any sections of the SDT which describe the actual TS (that is, the TS containing the SDT) shall have the table_id value 0x42 with the same table_id_extension (transport_stream_id) and with the same original_network_id. Any sections of an SDT which refer to a TS other than the actual TS shall take a table_id value of 0x46.

Table 5: Service description section

| Syntax | Number of bits | Identifier |
|---------------------------------|----------------|------------|
| service_description_section() { | | |
| table_id | 8 | uimbsf |
| section_syntax_indicator | 1 | bslbf |
| reserved_future_use | 1 | bslbf |
| reserved | 2 | bslbf |
| section_length | 12 | uimbsf |
| transport_stream_id | 16 | uimbsf |
| reserved | 2 | bslbf |
| version_number | 5 | uimbsf |
| current_next_indicator | 1 | bslbf |
| section_number | 8 | uimbsf |
| last_section_number | 8 | uimbsf |
| original_network_id | 16 | uimbsf |
| reserved_future_use | 8 | bslbf |
| for (i=0;i<N;i++){ | | |
| service_id | 16 | uimbsf |
| reserved_future_use | 6 | bslbf |
| EIT_schedule_flag | 1 | bslbf |
| EIT_present_following_flag | 1 | bslbf |
| running_status | 3 | uimbsf |
| free_CA_mode | 1 | bslbf |
| descriptors_loop_length | 12 | uimbsf |
| for (j=0;j<N;j++){ | | |
| descriptor() | | |
| } | | |
| } | | |
| CRC_32 | 32 | rpchof |
| } | | |

Semantics for the service description section:

table_id: See table 2.

section_syntax_indicator: The section_syntax_indicator is a 1-bit field which shall be set to "1".

section_length: This is a 12-bit field, the first two bits of which shall be "00". 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 1 021 so that the entire section has a maximum length of 1 024 bytes.

transport_stream_id: This is a 16-bit field which serves as a label for identification of the TS, about which the SDT informs, from any other multiplex within the delivery system.

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 table_id, transport_stream_id, and original_network_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.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

service_id: This is a 16-bit field which serves as a label to identify this service from any other service within the TS. The service_id is the same as the program_number in the corresponding program_map_section.

EIT_schedule_flag: This is a 1-bit field which when set to "1" indicates that EIT schedule information for the service is present in the current TS, see TS 101 211 [i.1] for information on maximum time interval between occurrences of an EIT schedule sub_table). If the flag is set to 0 then the EIT schedule information for the service should not be present in the TS.

EIT_present_following_flag: This is a 1-bit field which when set to "1" indicates that EIT_present_following information for the service is present in the current TS, see TS 101 211 [i.1] for information on maximum time interval between occurrences of an EIT present/following sub_table. If the flag is set to 0 then the EIT present/following information for the service should not be present in the TS.

running_status: This is a 3-bit field indicating the status of the service as defined in table 6.

Table 6: running_status

| Value | Meaning |
|--------|--|
| 0 | undefined |
| 1 | not running |
| 2 | starts in a few seconds (e.g. for video recording) |
| 3 | pausing |
| 4 | running |
| 5 | service off-air |
| 6 to 7 | reserved for future use |

For an NVOD reference service the value of the running_status shall be set to "0".

free_CA_mode: This 1-bit field, when set to "0" indicates that all the component streams of the service are not scrambled. When set to "1" it indicates that access to one or more streams may be controlled by a CA system.

descriptors_loop_length: This 12-bit field gives the total length in bytes of the following descriptors.

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex A of ISO/IEC 13181-1 [15] after processing the entire section.

5.2.4 Event Information Table (EIT)

The EIT (see table 7) provides information in chronological order regarding the events contained within each service. Four classifications of EIT have been identified, distinguishable by the use of different table_ids (see table 2):

- 1) actual TS, present/following event information = table_id = 0x4E;
- 2) other TS, present/following event information = table_id = 0x4F;
- 3) actual TS, event schedule information = table_id = 0x50 to 0x5F;
- 4) other TS, event schedule information = table_id = 0x60 to 0x6F.

All EIT sub_tables for the actual Transport Stream shall have the same transport_stream_id and original_network_id values.

The present/following table shall contain only information pertaining to the present event and the chronologically following event carried by a given service on either the actual TS or another TS, except in the case of a Near Video On Demand (NVD) reference service where it may have more than two event descriptions. The EIT present/following table is optional. Its presence or absence shall be signalled by setting the EIT_present_following_flag in the SDT.

The event schedule tables for either the actual TS or other TSs, contain a list of events, in the form of a schedule including events other than the present and following events. The EIT schedule tables are optional. Their presence or absence shall be signalled by setting the EIT_schedule_flag in the SDT. The event information shall be chronologically ordered.

The EIT shall be segmented into event_information_sections using the syntax of table 7. Any sections forming part of an EIT shall be transmitted in TS packets with a PID value of 0x0012.

Table 7: Event information section

| Syntax | Number of bits | Identifier |
|-------------------------------|----------------|------------|
| event_information_section() { | | |
| table_id | 8 | uimsbf |
| section_syntax_indicator | 1 | bslbf |
| reserved_future_use | 1 | bslbf |
| reserved | 2 | bslbf |
| section_length | 12 | uimsbf |
| service_id | 16 | uimsbf |
| reserved | 2 | bslbf |
| version_number | 5 | uimsbf |
| current_next_indicator | 1 | bslbf |
| section_number | 8 | uimsbf |
| last_section_number | 8 | uimsbf |
| transport_stream_id | 16 | uimsbf |
| original_network_id | 16 | uimsbf |
| segment_last_section_number | 8 | uimsbf |
| last_table_id | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| event_id | 16 | uimsbf |
| start_time | 40 | bslbf |
| duration | 24 | uimsbf |
| running_status | 3 | uimsbf |
| free_CA_mode | 1 | bslbf |
| descriptors_loop_length | 12 | uimsbf |
| for (i=0; i<N; i++) { | | |
| descriptor() | | |
| } | | |
| } | | |
| CRC_32 | 32 | rpchof |
| } | | |

Semantics for the event information section:

table_id: See table 2.

section_syntax_indicator: The section_syntax_indicator is a 1-bit field which shall be set to "1".

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.

service_id: This is a 16-bit field which serves as a label to identify this service from any other service within a TS. The service_id is the same as the program_number in the corresponding program_map_section.

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 table_id, service_id, transport_stream_id, and original_network_id. In this case, the sub_table may be structured as a number of segments. Within each segment the section_number shall increment by 1 with each additional section, but a gap in numbering is permitted between the last section of a segment and the first section of the adjacent segment.

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.

transport_stream_id: This is a 16-bit field which serves as a label for identification of the TS, about which the EIT informs, from any other multiplex within the delivery system.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

segment_last_section_number: This 8-bit field specifies the number of the last section of this segment of the sub_table. For sub_tables which are not segmented, this field shall be set to the same value as the last_section_number field.

last_table_id: This 8-bit field identifies the last table_id used (see table 2). For EIT present/following tables, this field shall be set to the same value as the table_id field. For EIT schedule tables with table_id in the range 0x50 to 0x5F, this field shall be set to the largest table_id transmitted in this range for this service. For EIT schedule tables with table_id in the range 0x60 to 0x6F, this field shall be set to the largest table_id transmitted in this range for this service.

NOTE: This implies that the value of last_table_id may be different for each service.

EXAMPLE 1: The table below shows some examples of last_table_id values for two services.

| Transmitted EIT table_id | Service | last_table_id |
|--------------------------|---------|---------------|
| 0x4E | A | 0x4E |
| 0x4F | A | 0x4F |
| 0x50, 0x51 | A | 0x51 |
| 0x60, 0x61, 0x62 | A | 0x62 |
| 0x4E | B | 0x4E |
| 0x4F | B | 0x4F |
| 0x50, 0x51, 0x52, 0x53 | B | 0x53 |
| 0x60 | B | 0x60 |

event_id: This 16-bit field contains the identification number of the described event (uniquely allocated within a service definition).

start_time: This 40-bit field contains the start time of the event in Universal Time, Co-ordinated (UTC) and Modified Julian Date (MJD) (see annex C). This field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 digits in 4-bit Binary Coded Decimal (BCD). If the start time is undefined (e.g. for an event in a NVOD reference service) all bits of the field are set to "1".

EXAMPLE 2: 93/10/13 12:45:00 is coded as "0xC079124500".

duration: A 24-bit field containing the duration of the event in hours, minutes, seconds. format: 6 digits, 4-bit BCD = 24 bit.

EXAMPLE 3: 01:45:30 is coded as "0x014530".

running_status: This is a 3-bit field indicating the status of the event as defined in table 6. For an NVOD reference event the value of the running_status shall be set to "0".

free_CA_mode: This 1-bit field, when set to "0" indicates that all the component streams of the event are not scrambled. When set to "1" it indicates that access to one or more streams is controlled by a CA system.

descriptors_loop_length: This 12-bit field gives the total length in bytes of the following descriptors.

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex A of ISO/IEC 13181-1 [15] after processing the entire private section.

5.2.5 Time and Date Table (TDT)

The TDT (see table 8) carries only the UTC-time and date information.

The TDT shall consist of a single section using the syntax of table 8. This TDT section shall be transmitted in TS packets with a PID value of 0x0014, and the table_id shall take the value 0x70.

Table 8: Time and date section

| Syntax | Number of bits | Identifier |
|--------------------------|----------------|------------|
| time_date_section() { | | |
| table_id | 8 | uimsbf |
| section_syntax_indicator | 1 | bslbf |
| reserved_future_use | 1 | bslbf |
| reserved | 2 | bslbf |
| section_length | 12 | uimsbf |
| UTC_time | 40 | bslbf |
| } | | |

Semantics for the time and date section:

table_id: See table 2.

section_syntax_indicator: This is a one-bit indicator which shall be set to "0".

section_length: This is a 12-bit field, the first two bits of which shall be "00". It specifies the number of bytes of the section, starting immediately following the section_length field and up to the end of the section.

UTC_time: This 40-bit field contains the current time and date in UTC and MJD (see annex C). This field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 digits in 4-bit BCD.

EXAMPLE: 93/10/13 12:45:00 is coded as "0xC079124500".

5.2.6 Time Offset Table (TOT)

The TOT (see table 9) carries the UTC-time and date information and local time offset. The TOT shall consist of a single section using the syntax of table 9. This TOT section shall be transmitted in TS packets with a PID value of 0x0014, and the table_id shall take the value 0x73.

Table 9: Time offset section

| Syntax | Number of bits | Identifier |
|--------------------------|----------------|------------|
| time_offset_section() { | | |
| table_id | 8 | uimsbf |
| section_syntax_indicator | 1 | bslbf |
| reserved_future_use | 1 | bslbf |
| reserved | 2 | bslbf |
| section_length | 12 | uimsbf |
| UTC_time | 40 | bslbf |
| reserved | 4 | bslbf |
| descriptors_loop_length | 12 | uimsbf |
| for (i=0; i<N; i++) { | | |
| descriptor() | | |
| } | | |
| CRC_32 | 32 | rpchof |
| } | | |

Semantics for the time offset section:

table_id: See table 2.

section_syntax_indicator: This is a one-bit indicator which shall be set to "0".

section_length: This is a 12-bit field, the first two bits of which shall be "00". It specifies the number of bytes of the section, starting immediately following the section_length field and up to the end of the section.

UTC_time: This 40-bit field contains the current time and date in UTC and MJD (see annex C). This field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 digits in 4-bit BCD.

EXAMPLE: 93/10/13 12:45:00 is coded as "0xC079124500".

descriptors_loop_length: This 12-bit field gives the total length in bytes of the following descriptors.

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex A of ISO/IEC 13181-1 [15] after processing the entire private section.

5.2.7 Running Status Table (RST)

The RST (see table 10) allows accurate and rapid updating of the timing status of one or more events. This may be necessary when an event starts early or late due to scheduling changes. The use of a separate table enables fast updating mechanism to be achieved.

The RST shall be segmented into running_status_sections using the syntax of table 10. Any sections forming part of an RST shall be transmitted in TS packets with a PID value of 0x0013, and the table_id shall take the value 0x71.

Table 10: Running status section

| Syntax | Number of bits | Identifier |
|----------------------------|----------------|------------|
| running_status_section() { | | |
| table_id | 8 | uimsbf |
| section_syntax_indicator | 1 | bslbf |
| reserved_future_use | 1 | bslbf |
| reserved | 2 | bslbf |
| section_length | 12 | uimsbf |
| for (i=0; i<N; i++) { | | |
| transport_stream_id | 16 | uimsbf |
| original_network_id | 16 | uimsbf |
| service_id | 16 | uimsbf |
| event_id | 16 | uimsbf |
| reserved_future_use | 5 | bslbf |
| running_status | 3 | uimsbf |
| } | | |
| } | | |

Semantics for the running status section:

table_id: See table 2.

section_syntax_indicator: This is a one-bit indicator which shall be set to "0".

section_length: This is a 12-bit field, the first two bits of which shall be "00". It specifies the number of bytes of the section, starting immediately following the section_length field and up to the end of the section. The section_length shall not exceed 1 021 so that the entire section has a maximum length of 1 024 bytes.

transport_stream_id: This is a 16-bit field which serves as a label for identification of the TS, about which the RST informs, from any other multiplex within the delivery system.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

service_id: This is a 16-bit field which serves as a label to identify this service from any other service within the TS. The service_id is the same as the program_number in the corresponding program_map_section.

event_id: This 16-bit field contains the identification number of the related event.

running_status: This is a 3-bit field indicating the status of the event, as defined in table 6.

5.2.8 Stuffing Table (ST)

The purpose of this clause (see table 11) is to invalidate existing sections at a delivery system boundary e.g. at a cable head-end. When one section of a sub_table is overwritten, then all the sections of that sub_table shall also be overwritten (stuffed) in order to retain the integrity of the section_number field.

Table 11: Stuffing section

| Syntax | Number of bits | Identifier |
|--------------------------|----------------|------------|
| stuffing_section() { | | |
| table_id | 8 | uimsbf |
| section_syntax_indicator | 1 | bslbf |
| reserved_future_use | 1 | bslbf |
| reserved | 2 | bslbf |
| section_length | 12 | uimsbf |
| for (i=0; i<N; i++) { | | |
| data_byte | 8 | uimsbf |
| } | | |
| } | | |

Semantics for the stuffing section:

table_id: See table 2.

section_syntax_indicator: This 1-bit field may take either the value "1" or "0".

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 up to the end of the section. The section_length shall not exceed 4 093 so that the entire section has a maximum length of 4 096 bytes.

data_byte: This 8-bit field may take any value and has no meaning.

5.2.9 Discontinuity Information Table (DIT)

See clause 7.1.1.

5.2.10 Selection Information Table (SIT)

See clause 7.1.2.

6 Descriptors

6.0 Introduction

This clause describes the different descriptors that can be used within the SI (for further information refer to the document, see TS 101 211 [i.1]).

6.1 Descriptor identification and location

Table 12 lists the descriptors declared or defined within the present document, giving the descriptors-tag values and the intended placement within the SI tables. This does not imply that their use in other tables is restricted.

Table 12: Possible locations of descriptors

| Descriptor | Tag value | NIT | BAT | SDT | EIT | TOT | PMT | SIT (note 1) |
|--------------------------------------|-----------|-----|-----|-----|-----|-----|-----|-----------------|
| network_name_descriptor | 0x40 | * | - | - | - | - | - | - |
| service_list_descriptor | 0x41 | * | * | - | - | - | - | - |
| stuffing_descriptor | 0x42 | * | * | * | * | - | - | * |
| satellite_delivery_system_descriptor | 0x43 | * | - | - | - | - | - | - |
| cable_delivery_system_descriptor | 0x44 | * | - | - | - | - | - | - |

| Descriptor | Tag value | NIT | BAT | SDT | EIT | TOT | PMT | SIT (note 1) |
|---|-----------|-----|-----|-----|-----|-----|-----|-----------------|
| VBI_data_descriptor | 0x45 | - | - | - | - | - | * | - |
| VBI_teletext_descriptor | 0x46 | - | - | - | - | - | * | - |
| bouquet_name_descriptor | 0x47 | - | * | - | - | - | - | * |
| service_descriptor | 0x48 | - | - | * | - | - | - | * |
| country_availability_descriptor | 0x49 | - | * | * | - | - | - | * |
| linkage_descriptor | 0x4A | * | * | * | * | - | - | * |
| NVOD_reference_descriptor | 0x4B | - | - | * | - | - | - | * |
| time_shifted_service_descriptor | 0x4C | - | - | * | - | - | - | * |
| short_event_descriptor | 0x4D | - | - | - | * | - | - | * |
| extended_event_descriptor | 0x4E | - | - | - | * | - | - | * |
| time_shifted_event_descriptor | 0x4F | - | - | - | * | - | - | * |
| component_descriptor | 0x50 | - | - | * | * | - | - | * |
| mosaic_descriptor | 0x51 | - | - | * | - | - | * | * |
| stream_identifier_descriptor | 0x52 | - | - | - | - | - | * | - |
| CA_identifier_descriptor | 0x53 | - | * | * | * | - | - | * |
| content_descriptor | 0x54 | - | - | - | * | - | - | * |
| parental_rating_descriptor | 0x55 | - | - | - | * | - | - | * |
| teletext_descriptor | 0x56 | - | - | - | - | - | * | - |
| telephone_descriptor | 0x57 | - | - | * | * | - | - | * |
| local_time_offset_descriptor | 0x58 | - | - | - | - | * | - | - |
| subtitling_descriptor | 0x59 | - | - | - | - | - | * | - |
| terrestrial_delivery_system_descriptor | 0x5A | * | - | - | - | - | - | - |
| multilingual_network_name_descriptor | 0x5B | * | - | - | - | - | - | - |
| multilingual_bouquet_name_descriptor | 0x5C | - | * | - | - | - | - | - |
| multilingual_service_name_descriptor | 0x5D | - | - | * | - | - | - | * |
| multilingual_component_descriptor | 0x5E | - | - | - | * | - | - | * |
| private_data_specifier_descriptor | 0x5F | * | * | * | * | - | * | * |
| service_move_descriptor | 0x60 | - | - | - | - | - | * | - |
| short_smoothing_buffer_descriptor | 0x61 | - | - | - | * | - | - | * |
| frequency_list_descriptor | 0x62 | * | - | - | - | - | - | - |
| partial_transport_stream_descriptor (see note 1) | 0x63 | - | - | - | - | - | - | * |
| data_broadcast_descriptor | 0x64 | - | - | * | * | - | - | * |
| scrambling_descriptor | 0x65 | - | - | - | - | - | * | - |
| data_broadcast_id_descriptor | 0x66 | - | - | - | - | - | * | - |
| transport_stream_descriptor (see note 2) | 0x67 | - | - | - | - | - | - | - |
| DSNG_descriptor (see note 2) | 0x68 | - | - | - | - | - | - | - |
| PDC_descriptor | 0x69 | - | - | - | * | - | - | - |
| AC-3_descriptor (see annex D) | 0x6A | - | - | - | - | - | * | - |
| ancillary_data_descriptor | 0x6B | - | - | - | - | - | * | - |
| cell_list_descriptor | 0x6C | * | - | - | - | - | - | - |
| cell_frequency_link_descriptor | 0x6D | * | - | - | - | - | - | - |
| announcement_support_descriptor | 0x6E | - | - | * | - | - | - | - |
| application_signalling_descriptor (see [48]) | 0x6F | - | - | - | - | - | * | - |
| adaptation_field_data_descriptor | 0x70 | - | - | - | - | - | * | - |
| service_identifier_descriptor (see [13]) | 0x71 | - | - | * | - | - | - | - |
| service_availability_descriptor | 0x72 | - | - | * | - | - | - | - |
| default_authority_descriptor (TS 102 323 [12]) | 0x73 | * | * | * | - | - | - | - |
| related_content_descriptor (TS 102 323 [12]) | 0x74 | - | - | - | - | - | * | - |
| TVA_id_descriptor (TS 102 323 [12]) | 0x75 | - | - | - | * | - | - | - |
| content_identifier_descriptor (TS 102 323 [12]) | 0x76 | - | - | - | * | - | - | - |

| Descriptor | Tag value | NIT | BAT | SDT | EIT | TOT | PMT | SIT (note 1) |
|---|--------------|-----|-----|-----|-----|-----|-----|-----------------|
| time_slice_fec_identifier_descriptor (EN 301 192 [4]) (see note 3) | 0x77 | * | - | - | - | - | - | - |
| ECM_repetition_rate_descriptor (EN 301 192 [4]) | 0x78 | - | - | - | - | - | * | - |
| S2_satellite_delivery_system_descriptor | 0x79 | * | - | - | - | - | - | - |
| enhanced_AC-3_descriptor (see annex D) | 0x7A | - | - | - | - | - | * | - |
| DTS@ descriptor (see annex G) | 0x7B | - | - | - | - | - | * | - |
| AAC descriptor (see annex H) | 0x7C | - | - | - | - | - | * | - |
| XAIT location descriptor (see [i.2]) | 0x7D | * | * | * | * | * | * | * |
| FTA_content_management_descriptor | 0x7E | * | * | * | * | - | - | - |
| extension descriptor (see note 5) | 0x7F | * | * | * | * | * | * | * |
| user defined | 0x80 to 0xFE | | | | | | | |
| reserved for future use | 0xFF | | | | | | | |

NOTE 1: Only found in Partial Transport Streams.
NOTE 2: Only in the TSDT (Transport Streams Description Table).
NOTE 3: May also be located in the CAT (ISO/IEC 13818-1 [15]) and INT (TS 102 006 [11]).
NOTE 4: * Possible location.
NOTE 5: See also clauses 6.3 and 6.4.

6.2 Descriptor coding

6.2.0 General principles

When the construct "descriptor ()" appears in the sections of clause 5.2, this indicates that zero or more of the descriptors defined within this clause shall occur.

The following semantics apply to all the descriptors defined in this clause.

descriptor_tag: The descriptor tag is an 8-bit field which identifies each descriptor. Those values with MPEG-2 normative meaning are described in ISO/IEC 13818-1 [15]. The values of descriptor_tag are defined in table 12.

descriptor_length: The descriptor length is an 8-bit field specifying the total number of bytes of the data portion of the descriptor following the byte defining the value of this field.

The bit and transmission ordering rules defined in clause 5.1.6 shall apply.

6.2.1 Adaptation field data descriptor

The adaptation field data descriptor (see table 13) provides a means of indicating the type of data fields supported within the private data field of the adaptation field. These data fields shall be coded according to annex D of TS 101 154 [9]. This descriptor shall be inserted into the corresponding ES_info loop of the PMT if the stream contains one or more of the data fields listed in table 14.

Table 13: Adaptation field data descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|---------------------------|
| adaptation_field_data_descriptor() { descriptor_tag descriptor_length adaptation_field_data_identifier } | 8 8 8 | uimsbf uimsbf bslbf |

Semantics for adaptation field data descriptor:

adaptation_field_data_identifier: This is an 8-bit field identifying data fields transmitted in the private data bytes of the adaptation field. It shall be coded according to table 14. If a bit in the adaptation_field_data_identifier is set to "1" it indicates that the transmission of the corresponding data field (as specified in the standard identified in the description column) is supported. The reserved_0 fields shall be set to 0.

NOTE: The data field does not necessarily occur in every adaptation field.

Table 14: Adaptation field data identifier coding

| adaptation field_data_identifier (Bit number) | Description |
|--|-----------------------------------|
| b ₀ (see note) | announcement switching data field |
| b ₁ | AU_information data field |
| b ₂ | PVR_assist_information_data_field |
| b ₃ | reserved_0 for future use |
| b ₄ | reserved_0 for future use |
| b ₅ | reserved_0 for future use |
| b ₆ | reserved_0 for future use |
| b ₇ | reserved_0 for future use |
| NOTE: This bit is transmitted last (see clause 5.1.6). | |

6.2.2 Ancillary data descriptor

The ancillary data descriptor provides a means of indicating the presence and the type of ancillary data in audio elementary streams coded according to clause 6.1 of TS 101 154 [9]. It shall be inserted into the corresponding ES_info loop of the PMT. If the ancillary data adheres to one of the formats in table 16, the descriptor shall be present.

Table 15: Ancillary data descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|---------------------------|
| ancillary_data_descriptor() { descriptor_tag descriptor_length ancillary_data_identifier } | 8 8 8 | uimbsf uimbsf bslbf |

Semantics for ancillary data descriptor:

ancillary_data_identifier: This is an 8-bit field identifying ancillary data coded in the audio elementary stream. It shall be coded according to table 16. If a bit in the ancillary_data_identifier field is set to "1" it indicates that ancillary data includes the corresponding data field.

Table 16: Ancillary data identifier coding

| ancillary_data_identifier (Bit number) | Description |
|--|---|
| b ₀ (see note) | DVD-Video Ancillary Data (TS 101 154 [9]) |
| b ₁ | Extended Ancillary Data (TS 101 154 [9]) |
| b ₂ | Announcement Switching Data (TS 101 154 [9]) |
| b ₃ | DAB Ancillary Data (EN 300 401 [2]) |
| b ₄ | Scale Factor Error Check (ScF-CRC) (TS 101 154 [9]) |
| b ₅ | MPEG-4 ancillary data (TS 101 154 [9], clause C.5) |
| b ₆ | RDS via UECF (TS 101 154 [9]) |
| b ₇ | Reserved for future use |
| NOTE: This bit is transmitted last (see clause 5.1.6). | |

6.2.3 Announcement support descriptor

The announcement support descriptor (see table 17) identifies the type of announcements that are supported by the service. Furthermore, it informs about the transport method of the announcement and gives the necessary linkage information so that the announcement stream can be monitored.

Table 17: Announcement support descriptor

| Syntax | Number of bits | Identifier |
|-------------------------------------|----------------|------------|
| announcement_support_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| announcement_support_indicator | 16 | bslbf |
| for (i=0; i<N; i++){ | | |
| announcement_type | 4 | uimsbf |
| reserved_future_use | 1 | bslbf |
| reference_type | 3 | uimsbf |
| if (reference_type == 0x01 | | |
| reference_type == 0x02 | | |
| reference_type == 0x03){ | | |
| original_network_id | 16 | uimsbf |
| transport_stream_id | 16 | uimsbf |
| service_id | 16 | uimsbf |
| component_tag | 8 | uimsbf |
| } | | |
| } | | |
| } | | |

Semantics of the announcement support descriptor:

announcement_support_indicator: The announcement support indicator is a 16-bit flag field specifying which types of announcements are supported by the service. The field should be coded according to table 18. If a specific type of announcement is not supported the corresponding bit shall be set to "0", if the announcement is supported the corresponding bit shall be set to "1".

Table 18: Coding of the announcement support indicator

| Bit flag | Description |
|--|-------------------------|
| b ₀ (see note) | Emergency alarm |
| b ₁ | Road Traffic flash |
| b ₂ | Public Transport flash |
| b ₃ | Warning message |
| b ₄ | News flash |
| b ₅ | Weather flash |
| b ₆ | Event announcement |
| b ₇ | Personal call |
| b ₈ to b ₁₅ | Reserved for future use |
| NOTE: This bit is transmitted last (see clause 5.1.6). | |

announcement_type: This 4-bit field specifies the type of announcement for which the following fields in the loop are valid, see table 19 coding.

Table 19: Coding of the announcement type

| Announcement type | Description |
|-------------------|-------------------------|
| 0000 | Emergency alarm |
| 0001 | Road Traffic flash |
| 0010 | Public Transport flash |
| 0011 | Warning message |
| 0100 | News flash |
| 0101 | Weather flash |
| 0110 | Event announcement |
| 0111 | Personal call |
| 1000 to 1111 | Reserved for future use |

reference type: This is a 3-bit field. It specifies the transport method of the announcement according to table 20.

Table 20: Coding of the reference type

| Reference type | Description |
|----------------|---|
| 000 | Announcement is broadcast in the usual audio stream of the service |
| 001 | Announcement is broadcast in a separate audio stream that is part of the service |
| 010 | Announcement is broadcast by means of a different service within the same transport stream |
| 011 | Announcement is broadcast by means of a different service within a different transport stream |
| 100 to 111 | Reserved for future use |

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system of the announcement service indicated.

transport_stream_id: This is a 16-bit field which uniquely identifies the TS containing the announcement service indicated.

service_id: This is a 16-bit field which uniquely identifies the service containing the announcements indicated.

component_tag: This 8-bit field has the same value as the component_tag field in the stream identifier descriptor that shall be present in the PSI program map section for the audio stream on which the announcement is broadcast.

6.2.4 Bouquet name descriptor

The bouquet name descriptor provides the bouquet name in text form, see table 21.

Table 21: Bouquet name descriptor

| Syntax | Number of bits | Identifier |
|-----------------------------|----------------|------------|
| bouquet_name_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| char | 8 | uimsbf |
| } | | |
| } | | |

Semantics for the bouquet name descriptor:

char: This is an 8-bit field, a sequence of which conveys the name of the bouquet about which the BAT sub_table informs. Text information is coded using the character sets and methods described in annex A.

6.2.5 CA identifier descriptor

The CA identifier descriptor (see table 22) indicates whether a particular bouquet, service or event is associated with a conditional access system and identifies the CA system type by means of the CA_system_id.

Table 22: CA identifier descriptor

| Syntax | Number of bits | Identifier |
|------------------------------|----------------|------------|
| CA_identifier_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| CA_system_id | 16 | uimsbf |
| } | | |
| } | | |

Semantics for the CA identifier descriptor:

CA_system_id: This 16-bit field identifies the CA system. It shall be coded according to TS 101 162 [57].

6.2.6 Cell frequency link descriptor

The cell frequency link descriptor (see table 23) may be used in the Network Information Table (NIT) that describes a terrestrial network. It gives a complete list of cells and identifies the frequencies that are in use in these cells for the multiplex described.

Table 23: Cell frequency link descriptor

| Syntax | Number of bits | Identifier |
|------------------------------------|----------------|------------|
| cell_frequency_link_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0;i<N;i++) { | | |
| cell_id | 16 | uimsbf |
| frequency | 32 | uimsbf |
| subcell_info_loop_length | 8 | uimsbf |
| for (j=0;j<N;j++) { | | |
| cell_id_extension | 8 | uimsbf |
| transposer_frequency | 32 | uimsbf |
| } | | |
| } | | |
| } | | |

Semantics for the cell frequency link descriptor:

cell_id: This is a 16-bit field which uniquely identifies a cell.

frequency: This 32-bit field identifies the main frequency that is used in the cell indicated. The coding is according to the coding of the centre_frequency in the terrestrial_delivery_system_descriptor.

subcell_info_loop_length: This 8-bit field gives the total length in bytes of the following loop that indicates the frequencies used in subcells.

cell_id_extension: This 8-bit field is used to identify a subcell within a cell.

transposer_frequency: This 32-bit field identifies the frequency that is used by a transposer in the subcell indicated. The coding of the frequency is according to the coding of the centre_frequency in the terrestrial_delivery_system_descriptor.

6.2.7 Cell list descriptor

The cell list descriptor (see table 24) may be used in the Network Information Table (NIT) that describes a terrestrial network. It provides a list of all cells of the network about which the NIT sub_table informs and describes their coverage areas.

Table 24: Cell list descriptor

| Syntax | Number of bits | Identifier |
|-----------------------------|----------------|------------|
| cell_list_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0;i<N;i++){ | | |
| cell_id | 16 | uimsbf |
| cell_latitude | 16 | uimsbf |
| cell_longitude | 16 | uimsbf |
| cell_extent_of_latitude | 12 | uimsbf |
| cell_extent_of_longitude | 12 | uimsbf |
| subcell_info_loop_length | 8 | uimsbf |
| for (j=0;j<N;j++){ | | |
| cell_id_extension | 8 | uimsbf |
| subcell_latitude | 16 | uimsbf |
| subcell_longitude | 16 | uimsbf |
| subcell_extent_of_latitude | 12 | uimsbf |
| subcell_extent_of_longitude | 12 | uimsbf |
| } | | |
| } | | |

| | | |
|-------------|--|--|
| } } } | | |
|-------------|--|--|

Semantics for the cell list descriptor:

cell_id: This is a 16-bit field which uniquely identifies a cell.

cell_latitude: This 16-bit field, coded as a two's complement number, shall specify the latitude of the south-western corner of a spherical rectangle that approximately describes the coverage area of the cell indicated. It shall be calculated by multiplying the value of the latitude field by $(90^\circ/2^{15})$. Southern latitudes shall be considered negative and northern latitudes positive.

cell_longitude: This 16-bit field, coded as a two's complement number, shall specify the longitude of the south-western corner of a spherical rectangle that approximately describes the coverage area of the cell indicated. It shall be calculated by multiplying the value of the longitude field by $(180^\circ/2^{15})$. Western longitudes shall be considered negative and eastern longitudes positive.

cell_extent_of_latitude: This 12-bit field, coded as an unsigned binary number, shall specify the extent of latitude of a spherical rectangle that approximately describes the coverage area of the cell indicated. It shall be calculated by multiplying the value of the extent_of_latitude field by $(90^\circ/2^{15})$.

cell_extent_of_longitude: This 12-bit field, coded as an unsigned binary number, shall specify the extent of longitude of a spherical rectangle that approximately describes the coverage area of the cell indicated. It shall be calculated by multiplying the value of the extent_of_longitude field by $(180^\circ/2^{15})$.

subcell_info_loop_length: This 8-bit field gives the total length in bytes of the following loop that describes the subcells.

cell_id_extension: This 8-bit field is used to identify a subcell within a cell.

subcell_latitude: This 16-bit field, coded as a two's complement number, shall specify the latitude of the corner of a spherical rectangle that approximately describes the coverage area of the subcell indicated. It shall be calculated by multiplying the value of the latitude field by $(90^\circ/2^{15})$. Southern latitudes shall be considered negative and northern latitudes positive.

subcell_longitude: This 16-bit field, coded as a two's complement number, shall specify the longitude of the corner of a spherical rectangle that approximately describes the coverage area of the subcell indicated. It shall be calculated by multiplying the value of the longitude field by $(180^\circ/2^{15})$. Western longitudes shall be considered negative and eastern longitudes positive.

subcell_extent_of_latitude: This 12-bit field, coded as an unsigned binary number, shall specify the extent of latitude of a spherical rectangle that approximately describes the coverage area of the subcell indicated. It shall be calculated by multiplying the value of the extent_of_latitude field by $(90^\circ/2^{15})$.

subcell_extent_of_longitude: This 12-bit field, coded as an unsigned binary number, shall specify the extent of longitude of a spherical rectangle that approximately describes the coverage area of the subcell indicated. It shall be calculated by multiplying the value of the extent_of_longitude field by $(180^\circ/2^{15})$.

6.2.8 Component descriptor

The component descriptor identifies the type of component stream and may be used to provide a text description of the elementary stream (see table 25). When used in the SDT and EIT these component stream characteristics, as are defined in table 26, describe the editorial characteristics of the event or service. These are intended to be used by receivers to inform the user of these characteristics.

Note that the use of some characteristics relating to HEVC video components (stream_content 0x9 and stream_content_ext 0x0) listed in table 26 imply additional semantics and are subject to additional rules (see also note 9 in table 26).

Table 25: Component descriptor

| Syntax | Number of bits | Identifier |
|---|----------------|------------|
| <code>component_descriptor() {</code> | | |
| <code>descriptor_tag</code> | 8 | uimsbf |
| <code>descriptor_length</code> | 8 | uimsbf |
| <code>stream_content_ext</code> | 4 | uimsbf |
| <code>stream_content</code> | 4 | uimsbf |
| <code>component_type</code> | 8 | uimsbf |
| <code>component_tag</code> | 8 | uimsbf |
| <code>ISO_639_language_code</code> | 24 | bslbf |
| for (<code>i=0; i<N; i++</code>) { | | |
| <code>text_char</code> | 8 | uimsbf |
| } | | |
| <code>}</code> | | |

Semantics for the component descriptor:

stream_content_ext: This 4-bit field in combination with the `stream_content` field specifies the type of stream. The coding of this field is specified in table 26. If the value of the `stream_content` field is in the range 0x1 to 0x8, the `stream_content_ext` field shall be set to 0xF to maintain backward compatibility with previous versions of this specification.

stream_content: This 4-bit field in combination with the `stream_content_ext` field specifies the type of stream. The coding of this field is specified in table 26.

component_type: This 8-bit field specifies the type of the component. The coding of this field is specified in table 26.

component_tag: This 8-bit field has the same value as the `component_tag` field in the stream identifier descriptor (if present in the PSI program map section) for the component stream. Multiple component descriptors for the same `component_tag` value are permitted as defined in table 26.

ISO_639_language_code: This 24-bit field identifies the language of the component (in the case of audio or EBU-data) and of the text description which may be contained in this descriptor. The `ISO_639_language_code` contains a 3-character code as specified by ISO 639-2 [36]. Both ISO 639-2 [36]/B and ISO 639-2 [36]/T may be used. Each character is coded into 8 bits according to ISO/IEC 8859-1 [17] 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".

text_char: This is an 8-bit field. A string of "text_char" fields specifies a text description of the component stream. Text information is coded using the character sets and methods described in annex A.

Table 26: stream_content, stream_content_ext and component_type

| stream_content | stream_content_ext | component_type | Description |
|----------------|--------------------|----------------|---|
| 0x0 | 0x0 to 0xF | 0x00 to 0xFF | reserved for future use |
| 0x1 | n/a (see note 8) | 0x00 | reserved for future use |
| | | 0x01 | MPEG-2 video, 4:3 aspect ratio, 25 Hz (see note 2) |
| | | 0x02 | MPEG-2 video, 16:9 aspect ratio with pan vectors, 25 Hz (see note 2) |
| | | 0x03 | MPEG-2 video, 16:9 aspect ratio without pan vectors, 25 Hz (see note 2) |
| | | 0x04 | MPEG-2 video, > 16:9 aspect ratio, 25 Hz (see note 2) |
| | | 0x05 | MPEG-2 video, 4:3 aspect ratio, 30 Hz (see note 2) |
| | | 0x06 | MPEG-2 video, 16:9 aspect ratio with pan vectors, 30 Hz (see note 2) |
| | | 0x07 | MPEG-2 video, 16:9 aspect ratio without pan vectors, 30 Hz (see note 2) |
| | | 0x08 | MPEG-2 video, > 16:9 aspect ratio, 30 Hz (see note 2) |
| | | 0x09 | MPEG-2 high definition video, 4:3 aspect ratio, 25 Hz (see note 2) |
| | | 0x0A | MPEG-2 high definition video, 16:9 aspect ratio with pan vectors, 25 Hz (see note 2) |
| | | 0x0B | MPEG-2 high definition video, 16:9 aspect ratio without pan vectors, 25 Hz (see note 2) |
| | | 0x0C | MPEG-2 high definition video, > 16:9 aspect ratio, 25 Hz (see note 2) |
| | | 0x0D | MPEG-2 high definition video, 4:3 aspect ratio, 30 Hz (see note 2) |
| | | 0x0E | MPEG-2 high definition video, 16:9 aspect ratio with pan vectors, 30 Hz (see note 2) |
| | | 0x0F | MPEG-2 high definition video, 16:9 aspect ratio without pan vectors, 30 Hz (see note 2) |
| | | 0x10 | MPEG-2 high definition video, > 16:9 aspect ratio, 30 Hz (see note 2) |
| | | 0x11 to 0xAF | reserved for future use |
| | | 0xB0 to 0xFE | user defined |
| | | 0xFF | reserved for future use |
| 0x2 | n/a (see note 8) | 0x00 | reserved for future use |
| | | 0x01 | MPEG-1 Layer 2 audio, single mono channel |
| | | 0x02 | MPEG-1 Layer 2 audio, dual mono channel |
| | | 0x03 | MPEG-1 Layer 2 audio, stereo (2 channel) |
| | | 0x04 | MPEG-1 Layer 2 audio, multi-lingual, multi-channel |
| | | 0x05 | MPEG-1 Layer 2 audio, surround sound |
| | | 0x06 to 0x3F | reserved for future use |
| | | 0x40 | MPEG-1 Layer 2 audio description for the visually impaired (see note 5) |
| | | 0x41 | MPEG-1 Layer 2 audio for the hard of hearing |
| | | 0x42 | receiver-mix supplementary audio as per annex E of TS 101 154 [9] |
| | | 0x43 to 0x46 | reserved for future use |
| | | 0x47 | MPEG-1 Layer 2 audio, receiver-mix audio description |
| | | 0x48 | MPEG-1 Layer 2 audio, broadcast-mix audio description |
| | | 0x49 to 0xAF | reserved for future use |
| | | 0xB0 to 0xFE | user-defined |
| | | 0xFF | reserved for future use |
| 0x3 | n/a (see note 8) | 0x00 | reserved for future use |
| | | 0x01 | EBU Teletext subtitles |
| | | 0x02 | associated EBU Teletext |
| | | 0x03 | VBI data |
| | | 0x04 to 0x0F | reserved for future use |
| | | 0x10 | DVB subtitles [55] (normal) with no monitor aspect ratio criticality |

| stream_content | stream_content_ext | component_type | Description |
|----------------|--------------------|----------------|---|
| | | 0x11 | DVB subtitles [55] (normal) for display on 4:3 aspect ratio monitor |
| | | 0x12 | DVB subtitles [55] (normal) for display on 16:9 aspect ratio monitor |
| | | 0x13 | DVB subtitles [55] (normal) for display on 2.21:1 aspect ratio monitor |
| | | 0x14 | DVB subtitles [55] (normal) for display on a high definition monitor |
| | | 0x15 | DVB subtitles [55] (normal) with plano-stereoscopic disparity for display on a high definition monitor |
| | | 0x16 | DVB subtitles [55] (normal) for display on an ultra high definition monitor |
| | | 0x17 to 0x1F | reserved for future use |
| | | 0x20 | DVB subtitles [55] (for the hard of hearing) with no monitor aspect ratio criticality |
| | | 0x21 | DVB subtitles [55] (for the hard of hearing) for display on 4:3 aspect ratio monitor |
| | | 0x22 | DVB subtitles [55] (for the hard of hearing) for display on 16:9 aspect ratio monitor |
| | | 0x23 | DVB subtitles [55] (for the hard of hearing) for display on 2.21:1 aspect ratio monitor |
| | | 0x24 | DVB subtitles [55] (for the hard of hearing) for display on a high definition monitor |
| | | 0x25 | DVB subtitles [55] (for the hard of hearing) with plano-stereoscopic disparity for display on a high definition monitor |
| | | 0x26 | DVB subtitles [55] (for the hard of hearing) for display on an ultra high definition monitor |
| | | 0x27 to 0x2F | reserved for future use |
| | | 0x30 | open (in-vision) sign language interpretation for the deaf (see note 7) |
| | | 0x31 | closed sign language interpretation for the deaf (see note 7) |
| | | 0x32 to 0x3F | reserved for future use |
| | | 0x40 | video spatial resolution has been upscaled from lower resolution source material (see note 7) |
| | | 0x41 | Video is standard dynamic range (SDR) (see note 7) |
| | | 0x42 | Video is high dynamic range (HDR) remapped from standard dynamic range (SDR) source material (see note 7) |
| | | 0x43 | Video is high dynamic range (HDR) up-converted from standard dynamic range (SDR) source material (see note 7) |
| | | 0x44 | Video is standard frame rate, less than or equal to 60 Hz (see note 7) |
| | | 0x45 | High frame rate video generated from lower frame rate source material (see note 7) |
| | | 0x46 to 0x7F | reserved for future use |
| | | 0x80 | dependent SAOC-DE data stream |
| | | 0x81 to 0xAF | reserved for future use |
| | | 0xB0 to 0xFE | user defined |
| | | 0xFF | reserved for future use |
| 0x4 | n/a (see note 8) | 0x00 to 0x7F | reserved for AC-3 audio modes (refer to table D.1) |
| | | 0x80 to 0xFF | reserved for enhanced AC-3 audio modes (refer to table D.1) |
| 0x5 | n/a (see note 8) | 0x00 | reserved for future use |
| | | 0x01 | H.264/AVC standard definition video, 4:3 aspect ratio, 25 Hz (see note 2) |
| | | 0x02 | reserved for future use |
| | | 0x03 | H.264/AVC standard definition video, 16:9 aspect ratio, 25 Hz (see note 2) |
| | | 0x04 | H.264/AVC standard definition video, > 16:9 aspect ratio, 25 Hz (see note 2) |

| stream_content | stream_content_ext | component_type | Description |
|----------------|--------------------|----------------|--|
| | | 0x05 | H.264/AVC standard definition video, 4:3 aspect ratio, 30 Hz (see note 2) |
| | | 0x06 | reserved for future use |
| | | 0x07 | H.264/AVC standard definition video, 16:9 aspect ratio, 30 Hz (see note 2) |
| | | 0x08 | H.264/AVC standard definition video, > 16:9 aspect ratio, 30 Hz (see note 2) |
| | | 0x09 to 0x0A | reserved for future use |
| | | 0x0B | H.264/AVC high definition video, 16:9 aspect ratio, 25 Hz (see note 2) |
| | | 0x0C | H.264/AVC high definition video, > 16:9 aspect ratio, 25 Hz (see note 2) |
| | | 0x0D to 0x0E | reserved for future use |
| | | 0x0F | H.264/AVC high definition video, 16:9 aspect ratio, 30 Hz (see note 2) |
| | | 0x10 | H.264/AVC high definition video, > 16:9 aspect ratio, 30 Hz (see note 2) |
| | | 0x11 to 0x7F | reserved for future use |
| | | 0x80 | H.264/AVC plano-stereoscopic frame compatible high definition video, 16:9 aspect ratio, 25 Hz, Side-by-Side (see notes 2 and 3) |
| | | 0x81 | H.264/AVC plano-stereoscopic frame compatible high definition video, 16:9 aspect ratio, 25 Hz, Top-and-Bottom (see notes 2 and 3) |
| | | 0x82 | H.264/AVC plano-stereoscopic frame compatible high definition video, 16:9 aspect ratio, 30 Hz, Side-by-Side (see notes 2, 3 and 4) |
| | | 0x83 | H.264/AVC stereoscopic frame compatible high definition video, 16:9 aspect ratio, 30 Hz, Top-and-Bottom (see notes 2, 3 and 4) |
| | | 0x84 | H.264/MVC dependent view, plano-stereoscopic service compatible video (see note 3) |
| | | 0x85 to 0xAF | reserved for future use |
| | | 0xB0 to 0xFE | user-defined |
| | | 0xFF | reserved for future use |
| 0x6 | n/a (see note 8) | 0x00 | reserved for future use |
| | | 0x01 | HE AAC audio, single mono channel (see note 6) |
| | | 0x02 | reserved for future use |
| | | 0x03 | HE AAC audio, stereo (see note 6) |
| | | 0x04 | reserved for future use |
| | | 0x05 | HE AAC audio, surround sound (see note 6) |
| | | 0x06 to 0x3F | reserved for future use |
| | | 0x40 | HE AAC audio description for the visually impaired (see notes 5 and 6) |
| | | 0x41 | HE AAC audio for the hard of hearing (see note 6) |
| | | 0x42 | HE AAC receiver-mix supplementary audio as per annex E of TS 101 154 [9] (see note 6) |
| | | 0x43 | HE AAC v2 audio, stereo |
| | | 0x44 | HE AAC v2 audio description for the visually impaired (see note 5) |
| | | 0x45 | HE AAC v2 audio for the hard of hearing |
| | | 0x46 | HE AAC v2 receiver-mix supplementary audio as per annex E of TS 101 154 [9] |
| | | 0x47 | HE AAC receiver-mix audio description for the visually impaired |
| | | 0x48 | HE AAC broadcast-mix audio description for the visually impaired |
| | | 0x49 | HE AAC v2 receiver-mix audio description for the visually impaired |
| | | 0x4A | HE AAC v2 broadcast-mix audio description for the visually impaired |
| | | 0x4B to 0x9F | reserved for future use |
| | | 0xA0 | HE AAC, or HE AAC v2 with SAOC-DE ancillary data (see notes 6 and 7) |
| | | 0xA1 to 0xAF | reserved for future use |

| stream_content | stream_content_ext | component_type | Description |
|----------------|--------------------|----------------|---|
| | | 0xB0 to 0xFE | user-defined |
| | | 0xFF | reserved for future use |
| | | 0x00 to 0x7F | reserved for DTS® and DTS-HD® audio modes (refer to annex G) |
| 0x7 | n/a (see note 8) | 0x80 to 0xFF | reserved for future use |
| 0x8 | n/a (see note 8) | 0x00 | reserved for future use |
| | | 0x01 | DVB SRM data [41] |
| | | 0x02 to 0xFF | reserved for DVB CPCM modes [39], [i.3] |
| 0x9 | 0x0 | 0x00 | HEVC Main Profile high definition video, 50 Hz (notes 2 and 9) |
| | | 0x01 | HEVC Main 10 Profile high definition video, 50 Hz (notes 2 and 9) |
| | | 0x02 | HEVC Main Profile high definition video, 60 Hz (notes 2, 4 and 9) |
| | | 0x03 | HEVC Main 10 Profile high definition video, 60 Hz (notes 2, 4 and 9) |
| | | 0x04 | HEVC ultra high definition video (notes 2, 9, and 10) conformant to one of the following UHDTV bit stream conformance points defined in table 18b in TS 101 154 [9]: <ul style="list-style-type: none"> • SDR frame rate up to 60 Hz • SDR HFR dual PID with temporal scalability • HDR with HLG10 frame rate up to 60 Hz • HDR with HLG10 HFR dual PID and temporal scalability |
| | | 0x05 | HEVC ultra high definition video with PQ10 HDR with a frame rate lower than or equal to 60 Hz (see notes 2 and 11) conformant to one of the following UHDTV bit stream conformance points defined in table 18b in TS 101 154 [9]: <ul style="list-style-type: none"> • HDR with PQ10 frame rate up to 60 Hz or HEVC ultra high definition video with PQ10 HDR with a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz with a half frame rate HEVC temporal video sub-bit-stream (see notes 2 and 11) conformant to one of the following UHDTV bit stream conformance points defined in table 18b in TS 101 154 [9]: <ul style="list-style-type: none"> • HDR with PQ10 HFR dual PID and temporal scalability |
| | | 0x06 | HEVC ultra high definition video, frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz without a half frame rate HEVC temporal video sub-bit-stream (see notes 2 and 11) conformant to one of the following UHDTV bit stream conformance points defined in table 18b in TS 101 154 [9]: <ul style="list-style-type: none"> • SDR HFR single PID • HDR with HLG10 HFR single PID |
| | | 0x07 | HEVC ultra high definition video with PQ10 HDR, frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz without a half frame rate HEVC temporal video sub-bit-stream (see notes 2 and 11) conformant to one of the following UHDTV bit stream conformance points defined in table 18b in TS 101 154 [9]: <ul style="list-style-type: none"> • HDR with PQ10 HFR single PID |
| | | 0x08 to 0xFF | reserved for future use |
| | 0x1 | 0x00 | AC-4 main audio, mono (see note 13) |
| | | 0x01 | AC-4 main audio, mono, dialogue enhancement enabled (see note 13) |
| | | 0x02 | AC-4 main audio, stereo (see note 13) |
| | | 0x03 | AC-4 main audio, stereo, dialogue enhancement enabled (see note 13) |
| | | 0x04 | AC-4 main audio, multichannel (see note 13) |

| stream_content | stream_content_ext | component_type | Description |
|----------------|--------------------|----------------|---|
| | | 0x05 | AC-4 main audio, multichannel, dialogue enhancement enabled (see note 13) |
| | | 0x06 | AC-4 broadcast-mix audio description, mono, for the visually impaired (see note 13) |
| | | 0x07 | AC-4 broadcast-mix audio description, mono, for the visually impaired, dialogue enhancement enabled (see note 13) |
| | | 0x08 | AC-4 broadcast-mix audio description, stereo, for the visually impaired (see note 13) |
| | | 0x09 | AC-4 broadcast-mix audio description, stereo, for the visually impaired, dialogue enhancement enabled (see note 13) |
| | | 0x0A | AC-4 broadcast-mix audio description, multichannel, for the visually impaired (see note 13) |
| | | 0x0B | AC-4 broadcast-mix audio description, multichannel, for the visually impaired, dialogue enhancement enabled (see note 13) |
| | | 0x0C | AC-4 receiver-mix audio description, mono, for the visually impaired (see note 13) |
| | | 0x0D | AC-4 receiver-mix audio description, stereo, for the visually impaired (see note 13) |
| | | 0x0E | AC-4 Part-2 |
| | | 0x0F | MPEG-H Audio LC Profile |
| | | 0x10 to 0xFF | reserved for future use |
| | 0x2 | 0x00 to 0xFF | TTML subtitles [56] |
| | 0x3 to 0xF | 0x00 to 0xFF | reserved for future use |
| 0xA | 0x0 to 0xF | 0x00 to 0xFF | reserved for future use |
| 0xB | 0x0 to 0xD | 0x00 to 0xFF | reserved for future use |
| | 0xE | 0x00 to 0xFF | NGA component type feature flags according to table 27 |
| | 0xF (see note 7) | 0x00 | less than 16:9 aspect ratio |
| | | 0x01 | 16:9 aspect ratio |
| | | 0x02 | greater than 16:9 aspect ratio |
| | | 0x03 | plano-stereoscopic top and bottom (TaB) frame-packing |
| | | 0x04 | HLG10 HDR (see notes 7, 11, and 12) |
| | | 0x05 | HEVC temporal video subset for a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz (see notes 11, and 12) |
| | | 0x06 to 0xFF | reserved for future use |
| 0xC to 0xF | n/a | 0x00 to 0xFF | user defined |

| stream_content | stream_content_ext | component_type | Description |
|--|--------------------|----------------|-------------|
| NOTE 1: The profiles and levels of the codecs mentioned in table 26 are as defined in TS 101 154 [9] and TS 102 005 [10]. | | | |
| NOTE 2: In table 26, the terms "standard definition", "high definition", "ultra high definition", "25 Hz", "30 Hz", "50 Hz", and "60 Hz" are used as defined in TS 101 154 [9] clauses 5.1 to 5.4 for MPEG-2 and 5.5 to 5.7 for H.264/AVC, and clauses 5.14.2 and 5.14.3 for HEVC respectively. The terms "HDR", "HLG10" and "PQ10" are used as defined in clause 5.14.3 of TS 101 154 [9]. | | | |
| NOTE 3: See TS 101 547-3 [47] for further information on stereoscopic modes. | | | |
| NOTE 4: 24 Hz video will also use this component_type. | | | |
| NOTE 5: The specific audio description types indicating the use of broadcast-mix or receiver-mix audio should be preferred over these generic types. For more details see annex J. | | | |
| NOTE 6: Audio streams using AAC audio shall use the corresponding HE AAC values. The AAC profile includes low-complexity AAC. | | | |
| NOTE 7: These component descriptor values are intended to be present in combination with one or more component descriptors with the same component_tag value. | | | |
| <ul style="list-style-type: none"> For example, two component descriptors with the same component tag value, and stream_content/stream_content_ext/component_type values of 0x5/0xF/0x0B and 0x3/0xF/0x40 respectively, would indicate H.264/AVC high definition video, 16:9 aspect ratio, 25 Hz that has been up-sampled from a standard definition source. For example, two component descriptors with the same component tag value, and stream_content/stream_content_ext/component_type values of 0x6/0xF/0x03 and 0x6/0xF/0xA0 respectively, would indicate stereo HE AAC audio with SAOC-DE parametric data embedded as ancillary data. For example, two component descriptors with the same component tag value, and stream_content/stream_content_ext/component_type values 0x9/0x0/0x01 and 0xB/0xF/0x01 respectively, would indicate HEVC Main 10 profile high definition video at 50 Hz with a 16:9 aspect ratio. For example, two components which together constitute a video stream of HLG10 HDR, remapped from standard dynamic range (SDR) source material, frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz with half frame rate HEVC temporal video sub-bitstream. The main component which is decodable by HEVC UHD TV IRD would be signaled using three component descriptors with stream_content/stream_content_ext/component_type values 0x9/0x0/0x04, 0x03/ 0xF/0x42 and 0xB/0xF/0x04 respectively. The other component would be signaled using stream_content/stream_content_ext/component_type value 0xB/0xF/0x05 would indicate HEVC temporal video subset for a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz. There is no need to include component descriptors with stream_content/stream_content_ext/component_type values 0x9/0x0/0x04, 0x03/ 0xF/0x42 for the secondary component since this information is implied from signaling of the main component. | | | |
| NOTE 8: In order to maintain backwards compatibility, the value of the stream_content_ext field is not applicable (n/a) for stream_content values in the range 0x01 to 0x8, and is set to 0xF. | | | |
| NOTE 9: For rules on the use of these values, see clause I.2.5 and TS 101 547-4 [53]. | | | |
| NOTE 10: This value should be used for backward compatible HLG10 HDR services, and/or backward compatible high frame rate services (HFR) which are decodable by HEVC_UHDTV_IRD as defined in TS 101 154 [9]. See also clause I.2.5.2. | | | |
| NOTE 11 For the rules on the use of these values, see annex I.2.6 | | | |
| NOTE 12 For the rules on the use of these values, see annex I.2.5.2 | | | |
| NOTE 13: These values should be used for elementary streams that convey a single presentation only. | | | |

Table 27 shows the assignment of component_type values in the component_descriptor in the case that the stream_content field is set to 0xB and stream_content_ext field is set to 0xE. If a bit in the component_type is set to "1", this indicates that a specific NGA feature (as specified in the description column) is supported. The reserved zero for future use field shall be set to 0.

Table 27: Next generation audio component_type value assignments

| component_type bits | | Description |
|---|----------------|---|
| b ₇ (MSB) | | reserved zero for future use |
| b ₆ | | content is pre-rendered for consumption with headphones |
| b ₅ | | content enables interactivity |
| b ₄ | | content enables dialogue enhancement (see note) |
| b ₃ | | content contains spoken subtitles |
| b ₂ | | content contains audio description |
| b ₁ | b ₀ | preferred reproduction channel layout: |
| 0 | 0 | no preference |
| 0 | 1 | stereo |
| 1 | 0 | two-dimensional |
| 1 | 1 | three-dimensional |
| NOTE: Content enabling dialogue enhancement also offers support for clean audio for the hearing impaired. | | |

The component descriptor signalling the available NGA features (stream_content 0xB, stream_content_ext 0xE) can be used alone without the additional component descriptor signalling the NGA codec type (stream_content 0x9, stream_content_ext 0x1, component_type 0xE or 0xF), in which case the codec type would be known from stream_type and, if present, descriptors within the PMT.

6.2.9 Content descriptor

The intention of the content descriptor (see table 28) is to provide classification information for an event.

Table 28: Content descriptor

| Syntax | Number of bits | Identifier |
|------------------------|----------------|------------|
| content_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| content_nibble_level_1 | 4 | uimsbf |
| content_nibble_level_2 | 4 | uimsbf |
| user_byte | 8 | uimsbf |
| } | | |
| } | | |

Semantics of the content descriptor:

content_nibble_level_1: This 4-bit field represents the first level of a content identifier. This field shall be coded according to table 29.

content_nibble_level_2: This 4-bit field represents the second level of a content identifier. This field shall be coded according to table 29.

user_byte: This 8-bit field is defined by the broadcaster.

Table 29: Content nibble level 1 and 2 assignments

| content_nibble_level_1 | content_nibble_level_2 | Description |
|------------------------|------------------------|-----------------------|
| 0x0 | 0x0 to 0xF | undefined content |
| | | Movie/Drama: |
| 0x1 | 0x0 | movie/drama (general) |

| content_nibble_level_1 | content_nibble_level_2 | Description |
|--------------------------------------|------------------------|--|
| | 0x1 | detective/thriller |
| | 0x2 | adventure/western/war |
| | 0x3 | science fiction/fantasy/horror |
| | 0x4 | comedy |
| | 0x5 | soap/melodrama/folkloric |
| | 0x6 | romance |
| | 0x7 | serious/classical/religious/historical movie/drama |
| | 0x8 | adult movie/drama |
| | 0x9 to 0xE | reserved for future use |
| | 0xF | user defined |
| News/Current affairs: | | |
| 0x2 | 0x0 | news/current affairs (general) |
| | 0x1 | news/weather report |
| | 0x2 | news magazine |
| | 0x3 | documentary |
| | 0x4 | discussion/interview/debate |
| | 0x5 to 0xE | reserved for future use |
| | 0xF | user defined |
| Show/Game show: | | |
| 0x3 | 0x0 | show/game show (general) |
| | 0x1 | game show/quiz/contest |
| | 0x2 | variety show |
| | 0x3 | talk show |
| | 0x4 to 0xE | reserved for future use |
| | 0xF | user defined |
| Sports: | | |
| 0x4 | 0x0 | sports (general) |
| | 0x1 | special events (Olympic Games, World Cup, etc.) |
| | 0x2 | sports magazines |
| | 0x3 | football/soccer |
| | 0x4 | tennis/squash |
| | 0x5 | team sports (excluding football) |
| | 0x6 | athletics |
| | 0x7 | motor sport |
| | 0x8 | water sport |
| | 0x9 | winter sports |
| | 0xA | equestrian |
| | 0xB | marital sports |
| | 0xC to 0xE | reserved for future use |
| | 0xF | user defined |
| Children's/Youth programmes: | | |
| 0x5 | 0x0 | children's/youth programmes (general) |
| | 0x1 | pre-school children's programmes |
| | 0x2 | entertainment programmes for 6 to 14 |
| | 0x3 | entertainment programmes for 10 to 16 |
| | 0x4 | informational/educational/school programmes |
| | 0x5 | cartoons/puppets |
| | 0x6 to 0xE | reserved for future use |
| | 0xF | user defined |
| Music/Ballet/Dance: | | |
| 0x6 | 0x0 | music/ballet/dance (general) |
| | 0x1 | rock/pop |
| | 0x2 | serious music/classical music |
| | 0x3 | folk/traditional music |
| | 0x4 | jazz |
| | 0x5 | musical/opera |
| | 0x6 | ballet |
| | 0x7 to 0xE | reserved for future use |
| | 0xF | user defined |
| Arts/Culture (without music): | | |
| 0x7 | 0x0 | arts/culture (without music, general) |

| content_nibble_level_1 | content_nibble_level_2 | Description |
|------------------------|------------------------|---|
| | 0x1 | performing arts |
| | 0x2 | fine arts |
| | 0x3 | religion |
| | 0x4 | popular culture/traditional arts |
| | 0x5 | literature |
| | 0x6 | film/cinema |
| | 0x7 | experimental film/video |
| | 0x8 | broadcasting/press |
| | 0x9 | new media |
| | 0xA | arts/culture magazines |
| | 0xB | fashion |
| | 0xC to 0xE | reserved for future use |
| | 0xF | user defined |
| | | Social/Political issues/Economics: |
| 0x8 | 0x0 | social/political issues/economics (general) |
| | 0x1 | magazines/reports/documentary |
| | 0x2 | economics/social advisory |
| | 0x3 | remarkable people |
| | 0x4 to 0xE | reserved for future use |
| | 0xF | user defined |
| | | Education/Science/Factual topics: |
| 0x9 | 0x0 | education/science/factual topics (general) |
| | 0x1 | nature/animals/environment |
| | 0x2 | technology/natural sciences |
| | 0x3 | medicine/physiology/psychology |
| | 0x4 | foreign countries/expeditions |
| | 0x5 | social/spiritual sciences |
| | 0x6 | further education |
| | 0x7 | languages |
| | 0x8 to 0xE | reserved for future use |
| | 0xF | user defined |
| | | Leisure hobbies: |
| 0xA | 0x0 | leisure hobbies (general) |
| | 0x1 | tourism/travel |
| | 0x2 | handicraft |
| | 0x3 | motoring |
| | 0x4 | fitness and health |
| | 0x5 | cooking |
| | 0x6 | advertisement/shopping |
| | 0x7 | gardening |
| | 0x8 to 0xE | reserved for future use |
| | 0xF | user defined |
| | | Special characteristics: |
| 0xB | 0x0 | original language |
| | 0x1 | black and white |
| | 0x2 | unpublished |
| | 0x3 | live broadcast |
| | 0x4 | plano-stereoscopic |
| | 0x5 | local or regional |
| | 0x6 to 0xE | reserved for future use |
| | 0xF | user defined |
| | | Adult: |
| 0xC | 0x0 | adult (general) |
| | 0x1 to 0xE | reserved for future use |
| | 0xF | user defined |
| | | Reserved for future use: |
| 0xD to 0xE | 0x0 to 0xF | reserved for future use |
| | | User defined: |
| 0xF | 0x0 to 0xF | user defined |

6.2.10 Country availability descriptor

In order to identify various combinations of countries efficiently, the descriptor may appear twice for each service, once giving a list of countries and/or groups of countries where the service is intended to be available, and the second giving a list of countries and/or groups where it is not. The latter list overrides the former list. If only one descriptor is used, which lists countries where the service is intended to be available, then it indicates that the service is not intended to be available in any other country. If only one descriptor is used, which lists countries where the service is not intended to be available, then it indicates that the service is intended to be available in every other country. If no descriptor is used, then it is not defined for which countries the service is intended to be available (see table 30).

Table 30: Country availability descriptor

| Syntax | Number of bits | Identifier |
|--|----------------------------|---|
| country_availability_descriptor() { descriptor_tag descriptor_length country_availability_flag reserved_future_use for (i=0; i<N; i++){ country_code } } | 8 8 1 7 24 | uimbsf uimbsf bslbf bslbf bslbf |

Semantics for the country availability descriptor:

country_availability_flag: This 1-bit field indicates whether the following country codes represent the countries in which the reception of the service is intended or not. If country_availability_flag is set to "1" the following country codes specify the countries in which the reception of the service is intended. If set to "0", the following country codes specify the countries in which the reception of the service is not intended.

country_code: This 24-bit field identifies a country or group of countries. It shall be coded as three characters, where each character is coded into 8-bits according to ISO/IEC 8859-1 [17] and inserted in order into the field. Individual countries shall be identified by using the alpha-3 code as specified in ISO 3166 [35]. A group of countries shall be identified by the string representation of a decimal number in the range 900 to 999. These numbers shall be coded according to TS 101 162 [57].

EXAMPLE: United Kingdom has 3-character code "GBR", which is coded as:
 "0100 0111 0100 0010 0101 0010".

6.2.11 Data broadcast descriptor

The data broadcast descriptor identifies the type of the data component and may be used to provide a text description of the data component (see table 31).

Table 31: Data broadcast descriptor

| Syntax | Number of bits | Identifier |
|--|---|---|
| data_broadcast_descriptor() { descriptor_tag descriptor_length data_broadcast_id component_tag selector_length for (i=0; i<selector_length; i++){ selector_byte } ISO_639_language_code text_length for (i=0; i<text_length; i++){ text_char } } | 8 8 16 8 8 8 24 8 8 | uimbsf uimbsf uimbsf uimbsf uimbsf uimbsf bslbf uimbsf uimbsf |

Semantics of the data broadcast descriptor:

data_broadcast_id: This 16-bit field identifies the data broadcast specification that is used to broadcast the data in the broadcast network. It shall be coded according to TS 101 162 [57].

component_tag: This optional 8-bit field has the same value as the component_tag field in the stream identifier descriptor that may be present in the PSI program map section for the stream on which the data is broadcast. If this field is not used it shall be set to the value 0x00.

selector_length: This 8-bit field specifies the length in bytes of the following selector field.

selector_byte: This is an 8-bit field. The sequence of selector_byte fields specifies the selector field. The syntax and semantics of the selector field shall be defined by the data broadcast specification that is identified in the data_broadcast_id field. The selector field may contain service specific information that is necessary to identify an entry-point of the broadcast data.

ISO_639_language_code: This 24-bit field contains the ISO 639-2 [36] three character language code of the following text fields. Both ISO 639-2/B and ISO 639-2/T may be used. Each character is coded into 8 bits according to ISO/IEC 8859-1 [17] and inserted in order into the 24-bit field.

text_length: This 8-bit field specifies the length in bytes of the following text describing the data component.

text_char: This is an 8-bit field. A string of "char" fields specifies the text description of the data component. Text information is coded using the character sets and methods described in annex A.

6.2.12 Data broadcast id descriptor

The data broadcast id descriptor identifies the type of the data component (see table 32). It is a short form of the data_broadcast descriptor and it may be placed in the component loop of the PSI PMT table.

Table 32: Data broadcast id descriptor

| Syntax | Number of bits | Identifier |
|---|----------------|------------|
| <code>data_broadcast_id_descriptor() {</code> | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| data_broadcast_id | 16 | uimsbf |
| for(i=0; i < N;i++){ | | |
| id_selector_byte | 8 | uimsbf |
| } | | |
| <code>}</code> | | |

Semantics of the data broadcast id descriptor:

data_broadcast_id: This 16-bit field identifies the data broadcast specification that is used to broadcast the data in the broadcast network. It shall be coded according to TS 101 162 [57].

id_selector_byte: For the purpose of application selection the id_selector_bytes might be used. The definition of the id_selector_bytes of the data_broadcast_id_descriptor will depend on the data broadcast id.

NOTE: The id_selector_bytes may differ from the selector_bytes of the corresponding data_broadcast_descriptor.

6.2.13 Delivery system descriptors

6.2.13.1 Cable delivery system descriptor

See table 33.

Table 33: Cable delivery system descriptor

| Syntax | Number of bits | Identifier |
|---|----------------|------------|
| <code>cable_delivery_system_descriptor() {</code> | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| frequency | 32 | bslbf |
| reserved_future_use | 12 | bslbf |
| FEC_outer | 4 | bslbf |

| | | |
|-------------|----|-------|
| modulation | 8 | bslbf |
| symbol_rate | 28 | bslbf |
| FEC_inner | 4 | bslbf |
| } | | |

Semantics for cable delivery system descriptor:

frequency: The frequency is a 32-bit field giving the 4-bit BCD values specifying 8 characters of the frequency value. For the cable_delivery_system_descriptor, the frequency is coded in MHz, where the decimal occurs after the fourth character (e.g. 0312,0000 MHz).

FEC_outer: The FEC_outer is a 4-bit field specifying the outer Forward Error Correction (FEC) scheme used according to table 34.

Table 34: Outer FEC scheme

| FEC_outer bit 3210 | Description |
|--------------------|-------------------------|
| 0000 | not defined |
| 0001 | no outer FEC coding |
| 0010 | RS(204/188) |
| 0011 to 1111 | reserved for future use |

modulation: This is an 8-bit field. It specifies the modulation scheme used on a cable delivery system according to table 35.

Table 35: Modulation scheme for cable

| Modulation (hex) | Description |
|------------------|-------------------------|
| 0x00 | not defined |
| 0x01 | 16-QAM |
| 0x02 | 32-QAM |
| 0x03 | 64-QAM |
| 0x04 | 128-QAM |
| 0x05 | 256-QAM |
| 0x06 to 0xFF | reserved for future use |

symbol_rate: The symbol_rate is a 28-bit field giving the 4-bit BCD values specifying 7 characters of the symbol_rate in Msymbol/s where the decimal point occurs after the third character (e.g. 027,4500).

FEC_inner: The FEC_inner is a 4-bit field specifying the inner FEC scheme used according to table 36.

Table 36: Inner FEC scheme

| FEC_inner bit 3210 | Description |
|--|-------------------------|
| 0000 | not defined |
| 0001 | 1/2 conv. code rate |
| 0010 | 2/3 conv. code rate |
| 0011 | 3/4 conv. code rate |
| 0100 | 5/6 conv. code rate |
| 0101 | 7/8 conv. code rate |
| 0110 | 8/9 conv. code rate |
| 0111 | 3/5 conv. code rate |
| 1000 | 4/5 conv. code rate |
| 1001 | 9/10 conv. code rate |
| 1010 to 1110 | reserved for future use |
| 1111 | no conv. Coding |
| NOTE: Not all convolutional code rates apply for all modulation schemes. | |

6.2.13.2 Satellite delivery system descriptor

See table 37.

Table 37: Satellite delivery system descriptor

| Syntax | Number of bits | Identifier |
|---|----------------|------------|
| satellite_delivery_system_descriptor(){ | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| frequency | 32 | bslbf |
| orbital_position | 16 | bslbf |
| west_east_flag | 1 | bslbf |
| polarization | 2 | bslbf |
| if (modulation_system == "1") { | | |
| roll_off | 2 | bslbf |
| } else { | | |
| reserved_zero_future_use | 2 | bslbf |
| } | | |
| modulation_system | 1 | bslbf |
| modulation_type | 2 | bslbf |
| symbol_rate | 28 | bslbf |
| FEC_inner | 4 | bslbf |
| } | | |

Semantics for satellite delivery system descriptor:

frequency: The frequency is a 32-bit field giving the 4-bit BCD values specifying 8 characters of the frequency value. For the satellite_delivery_system_descriptor the frequency is coded in GHz, where the decimal point occurs after the third character (e.g. 011,75725 GHz).

orbital_position: The orbital_position is a 16-bit field giving the 4-bit BCD values specifying 4 characters of the orbital position in degrees where the decimal point occurs after the third character (e.g. 019,2°).

west_east_flag: The west_east_flag is a 1-bit field indicating if the satellite position is in the western or eastern part of the orbit. A value "0" indicates the western position and a value "1" indicates the eastern position.

polarization: The polarization is a 2-bit field specifying the polarization of the transmitted signal. The first bit defines whether the polarization is linear or circular (see table 38).

Table 38: Polarization

| Polarization | Description |
|--------------|---------------------|
| 00 | linear - horizontal |
| 01 | linear - vertical |
| 10 | Circular - left |
| 11 | Circular - right |

roll_off: This 2 bit field specifies the roll-off factor used in DVB-S2. It shall be coded according to table 39.

Table 39: Roll-off factor

| roll-off | Description |
|----------|-----------------|
| 00 | $\alpha = 0,35$ |
| 01 | $\alpha = 0,25$ |
| 10 | $\alpha = 0,20$ |
| 11 | reserved |

modulation_system: This is a 1-bit field. It specifies the broadcast scheme used on a satellite delivery system according to table 40.

If DVB-S2 is used with scrambling sequences other than the default in physical layer scrambling, or with multiple input streams, or with time slicing, the S2_satellite_delivery_system_descriptor shall be used additionally (see clause 6.2.13.3).

Table 40: Modulation system for satellite

| modulation system | Description |
|-------------------|-------------|
| 0 | DVB-S |
| 1 | DVB-S2 |

modulation_type: This is a 2-bit field. It specifies the modulation scheme used on a satellite delivery system according to table 41.

Table 41: Modulation type for satellite

| modulation type | Description |
|-----------------|-------------------------|
| 00 | Auto |
| 01 | QPSK |
| 10 | 8PSK |
| 11 | 16-QAM (n/a for DVB-S2) |

symbol_rate: The symbol_rate is a 28-bit field giving the 4-bit BCD values specifying 7 characters of the symbol_rate in Msymbol/s where the decimal point occurs after the third character (e.g. 027,4500).

FEC_inner: The FEC_inner is a 4-bit field specifying the inner FEC scheme used according to table 36.

6.2.13.3 S2 satellite delivery system descriptor

When DVB-S2 is used, this descriptor (see table 42) may be used in addition to the satellite delivery system descriptor (see clause 6.2.13.2) to indicate the scrambling sequence used in physical layer scrambling, or the use of multiple input streams, or the use of time slicing.

Table 42: S2 satellite delivery system descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| S2_satellite_delivery_system_descriptor(){ | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| scrambling_sequence_selector | 1 | bslbf |
| multiple_input_stream_flag | 1 | bslbf |
| reserved_zero_future_use | 1 | bslbf |
| not_timeslice_flag | 1 | bslbf |
| reserved_future_use | 2 | bslbf |
| TS_GS_mode | 2 | uimsbf |
| if (scrambling_sequence_selector == 1){ | | |
| reserved_future_use | 6 | bslbf |
| scrambling_sequence_index | 18 | uimsbf |
| } | | |
| if (multiple_input_stream_flag == 1){ | | |
| input_stream_identifier | 8 | uimsbf |
| } | | |
| if (not_timeslice_flag == 0){ | | |
| timeslice_number | 8 | uimsbf |
| } | | |
| } | | |

Semantics for the S2_satellite_delivery_system_descriptor:

scrambling_sequence_selector: A value 0 specifies that the default DVB-S2 physical layer scrambling sequence of index n = 0 is used. A value 1 means that the default scrambling sequence is not used and that the sequence index n will be conveyed by the scrambling_sequence_index field.

multiple_input_stream_flag: The multiple_input_stream_flag is a 1 bit field indicating whether single or multiple transport streams are conveyed. A value "0" indicates that a single transport stream is carried. A value "1" indicates that multiple transport streams are conveyed and that the input_stream_identifier (ISI) field is present.

TS_GS_mode: This 2-bit field indicates which mode the Input Stream is operating in. It shall be coded according to table 43.

Table 43: Coding of the TS GS mode field

| TS_GS_mode | Description |
|------------|---------------------|
| 0 | Generic packetized |
| 1 | GSE |
| 2 | Transport Stream |
| 3 | reserved future use |

NOTE: These values are different from similar assignments in table 3 in EN 302 307-1 [8].

scrambling_sequence_index: This 18 bit field, when present, carries the index of the DVB-S2 physical layer scrambling sequence as defined in clause 5.5.4 of EN 302 307-1 [8].

not_timeslice_flag: This 1-bit field indicates the use of time slicing as defined in Annex M of EN 302 307-1 [8]. When set to 1, it indicates that time slicing is not used. When set to 0, it indicates that time slicing is used.

input_stream_identifier: This 8 bit field carries the DVB-S2 input_stream_identifier (ISI) as defined in clause 5.1.6 of EN 302 307-1 [8].

timeslice_number: When present, this 8-bit field indicates which time slice carries all data for the stream being described.

6.2.13.4 Terrestrial delivery system descriptor

See table 44.

NOTE: The delivery system descriptor for DVB-T2 transmissions is specified in clause 6.4.6.3.

Table 44: Terrestrial delivery system descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| terrestrial_delivery_system_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| centre_frequency | 32 | uimsbf |
| bandwidth | 3 | bslbf |
| priority | 1 | bslbf |
| Time_Slicing_indicator | 1 | bslbf |
| MPE-FEC_indicator | 1 | bslbf |
| reserved_future_use | 2 | bslbf |
| constellation | 2 | bslbf |
| hierarchy_information | 3 | bslbf |
| code_rate-HP_stream | 3 | bslbf |
| code_rate-LP_stream | 3 | bslbf |
| guard_interval | 2 | bslbf |
| transmission_mode | 2 | bslbf |
| other_frequency_flag | 1 | bslbf |
| reserved_future_use | 32 | bslbf |
| } | | |

Semantics for terrestrial delivery system descriptor:

centre_frequency: The centre_frequency is a 32-bit field giving the centre frequency value in multiples of 10 Hz. The coding range is from minimum 10 Hz (0x00000001) up to a maximum of 42 949 672 950 Hz (0xFFFFFFFF).

bandwidth: This is a 3-bit field specifying the bandwidth in use. It shall be coded according to table 45.

Table 45: Signalling format for the bandwidth

| Bandwidth | Bandwidth value |
|------------|-------------------------|
| 000 | 8 MHz |
| 001 | 7 MHz |
| 010 | 6 MHz |
| 011 | 5 MHz |
| 100 to 111 | Reserved for future use |

priority: This 1-bit flag indicates the stream's hierarchical priority. It shall be coded according to table 46. In case the hierarchy_information field is not equal to "000", if priority is set to "1", it indicates that the associated transport stream is a HP stream, and if priority is set to "0", the associated transport stream is a LP stream. In case the hierarchy_information field has the value "000", the priority flag shall be set to "1".

Table 46: Signalling format for the priority

| priority | Description |
|----------|--------------------|
| 1 | HP (high priority) |
| 0 | LP (low priority) |

Time_Slicing_indicator: This 1-bit field indicates the use of the Time Slicing on the associated transport stream. If the Time_Slicing_indicator is set ("1"), Time Slicing is not used. If the Time_Slicing_indicator is cleared ("0"), at least one elementary stream uses Time Slicing.

The Time Slicing is defined in EN 301 192 [4].

MPE-FEC_indicator: This 1-bit field indicates the use of the MPE-FEC on the associated transport stream. If the MPE-FEC_indicator is set ("1"), MPE-FEC is not used. If the MPE-FEC_indicator is cleared ("0"), at least one elementary stream uses MPE-FEC.

The MPE-FEC is defined in EN 301 192 [4].

constellation: This is a 2-bit field. It specifies the constellation pattern used on a terrestrial delivery system according to table 47.

Table 47: Signalling format for the possible constellation patterns

| Constellation | Constellation characteristics |
|---------------|-------------------------------|
| 00 | QPSK |
| 01 | 16-QAM |
| 10 | 64-QAM |
| 11 | reserved for future use |

hierarchy_information: The hierarchy_information specifies whether the transmission is hierarchical and, if so, what the α value is (see table 48). Also, the use of in-depth interleaver is indicated. When the transmission_mode indicates the use of 8k mode, only the native interleaver shall be signalled.

Table 48: Signalling format for the α values and the used interleaver

| Hierarchy_information | α value |
|-----------------------|--|
| 000 | non-hierarchical, native interleaver |
| 001 | $\alpha = 1$, native interleaver |
| 010 | $\alpha = 2$, native interleaver |
| 011 | $\alpha = 4$, native interleaver |
| 100 | non-hierarchical, in-depth interleaver |
| 101 | $\alpha = 1$, in-depth interleaver |
| 110 | $\alpha = 2$, in-depth interleaver |
| 111 | $\alpha = 4$, in-depth interleaver |

code_rate: The code_rate is a 3-bit field specifying the inner FEC scheme used according to table 49. Non-hierarchical channel coding and modulation requires signalling of one code rate. In this case, 3 bits specifying code_rate according

to table 49 are followed by another 3 bits of value "000". Two different code rates may be applied to two different levels of modulation with the aim of achieving hierarchy. Transmission then starts with the code rate for the HP level of the modulation and ends with the one for the LP level.

Table 49: Signalling format for each of the code rates

| code_rate | Description |
|------------|-------------------------|
| 000 | 1/2 |
| 001 | 2/3 |
| 010 | 3/4 |
| 011 | 5/6 |
| 100 | 7/8 |
| 101 to 111 | reserved for future use |

guard_interval: The guard_interval is a 2-bit field specifying the guard interval according to table 50.

Table 50: Signalling format for each of the guard interval values

| guard_interval | Guard interval values |
|----------------|-----------------------|
| 00 | 1/32 |
| 01 | 1/16 |
| 10 | 1/8 |
| 11 | 1/4 |

transmission_mode: This 2-bit field indicates the number of carriers in an OFDM frame according to table 51.

Table 51: Signalling format for transmission mode

| transmission_mode | Description |
|-------------------|-------------------------|
| 00 | 2k mode |
| 01 | 8k mode |
| 10 | 4k mode |
| 11 | reserved for future use |

other_frequency_flag: This 1-bit flag indicates whether other frequencies are in use. The value "0" indicates that no other frequency is in use, "1" indicates that one or more other frequencies are in use.

6.2.14 DSNG descriptor

In Digital Satellite News Gathering (DSNG) transmissions the Transport Stream Description Table (TSDT) shall be present in the bitstream and the TSDT descriptor loop shall contain the DSNG descriptor with the ASCII codes for "CONA" in the text field (see EN 301 210 [5]).

For DSNG applications at least one DSNG descriptor (see table 52) shall be present in the TSDT.

Table 52: DSNG descriptor

| Syntax | Number of bits | Identifier |
|-----------------------|----------------|------------|
| DSNG_descriptor () { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| byte | 8 | uimsbf |
| } | | |
| } | | |

byte: The sequence of bytes in the DSNG descriptor is defined EN 301 210 [5].

6.2.15 Extended event descriptor

The extended event descriptor (see table 53) provides a detailed text description of an event, which may be used in addition to the short event descriptor. More than one extended event descriptor can be associated to allow information

about one event greater in length than 256 bytes to be conveyed. Text information can be structured into two columns, one giving an item description field and the other the item text. A typical application for this structure is to give a cast list, where for example the item description field might be "Producer" and the item field would give the name of the producer.

Table 53: Extended event descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| <code>extended_event_descriptor() {</code> | | |
| <code>descriptor_tag</code> | 8 | uimbsf |
| <code>descriptor_length</code> | 8 | uimbsf |
| <code>descriptor_number</code> | 4 | uimbsf |
| <code>last_descriptor_number</code> | 4 | uimbsf |
| <code>ISO_639_language_code</code> | 24 | bslbf |
| <code>length_of_items</code> | 8 | uimbsf |
| <code>for (i=0;i<N;i++){</code> | | |
| <code>item_description_length</code> | 8 | uimbsf |
| <code>for (j=0;j<N;j++){</code> | | |
| <code>item_description_char</code> | 8 | uimbsf |
| <code>}</code> | | |
| <code>item_length</code> | 8 | uimbsf |
| <code>for (j=0;j<N;j++){</code> | | |
| <code>item_char</code> | 8 | uimbsf |
| <code>}</code> | | |
| <code>}</code> | | |
| <code>text_length</code> | 8 | uimbsf |
| <code>for (i=0;i<N;i++){</code> | | |
| <code>text_char</code> | 8 | uimbsf |
| <code>}</code> | | |
| <code>}</code> | | |

Semantics for the extended event descriptor:

descriptor_number: This 4-bit field gives the number of the descriptor. It is used to associate information which cannot be fitted into a single descriptor. The descriptor_number of the first extended_event_descriptor of an associated set of extended_event_descriptors shall be "0x00". The descriptor_number shall be incremented by 1 with each additional extended_event_descriptor in this section.

last_descriptor_number: This 4-bit field specifies the number of the last extended_event_descriptor (that is, the descriptor with the highest value of descriptor_number) of the associated set of descriptors of which this descriptor is part.

ISO_639_language_code: This 24-bit field identifies the language of the following text fields. The ISO_639_language_code contains a 3-character code as specified by ISO 639-2 [36]. Both ISO 639-2/B and ISO 639-2/T may be used. Each character is coded into 8 bits according to ISO/IEC 8859-1 [17] 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".

length_of_items: This is an 8-bit field specifying the length in bytes of the following items.

item_description_length: This 8-bit field specifies the length in bytes of the item description.

item_description_char: This is an 8-bit field. A string of "item_description_char" fields specify the item description. Text information is coded using the character sets and methods described in annex A.

item_length: This 8-bit field specifies the length in bytes of the item text.

item_char: This is an 8-bit field. A string of "item_char" fields specify the item text. Text information is coded using the character sets and methods described in annex A.

text_length: This 8-bit field specifies the length in bytes of the non itemized extended text.

text_char: This is an 8-bit field. A string of "text_char" fields specify the non itemized extended text. Text information is coded using the character sets and methods described in annex A.

6.2.16 Extension descriptor

The extension descriptor is (see table 54) used to extend the 8-bit namespace of the descriptor_tag field.

Table 54: Extension descriptor

| Syntax | Number of bits | Identifier |
|--------------------------|----------------|------------|
| extension_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| for (i=0;i<N;i++){ | | |
| selector_byte | 8 | bslbf |
| } | | |
| } | | |

Semantics for the extension descriptor:

descriptor_tag_extension: The descriptor tag extension is an 8-bit field which identifies each extended descriptor. The values of descriptor_tag_extension are defined in clause 6.3.

selector_byte: This is an 8-bit field. The sequence of selector_byte fields specifies the selector field. The syntax and semantics of the selector field are defined in clause 6.4.

6.2.17 Frequency list descriptor

The frequency list descriptor (see table 55) may be used in the NIT. It gives the complete list of additional frequencies for a certain multiplex which is transmitted on multiple frequencies.

Table 55: Frequency list descriptor

| Syntax | Number of bits | Identifier |
|-------------------------------|----------------|------------|
| frequency_list_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| reserved_future_use | 6 | bslbf |
| coding_type | 2 | bslbf |
| for (i=0;I<N;i++){ | | |
| centre_frequency | 32 | uimsbf |
| } | | |
| } | | |

Semantics for the frequency list descriptor:

coding_type: This is a 2-bit field that indicates how the frequency is coded and relates to the delivery system used. It has a value indicated in table 56.

Table 56: Coding type values

| Coding_type | Delivery system |
|-------------|-----------------|
| 00 | not defined |
| 01 | satellite |
| 10 | cable |
| 11 | terrestrial |

centre_frequency: This is as defined in the delivery_system_descriptor for the delivery system given by the coding_type.

6.2.18 FTA content management descriptor

6.2.18.0 Semantics and syntax of the FTA content management descriptor

The FTA content management descriptor (see table 57) provides a means of defining the content management policy for an item of content delivered as part of a free-to-air (FTA) DVB Service.

The signalling conveyed by this descriptor has been designed for use in the context of the DVB's Content Protection Copy Management (CPCM) solution. However, the descriptor may also be used in a non-CPCM context.

The FTA flags are intended to signal to a receiving device (not limited to a DVB CPCM implementation) the configuration of internal states that affect the means by which content might be redistributed. It reflects the broadcaster's intention for content usage and shall result in a similar user experience across implementations.

The FTA content usage restrictions defined in this clause apply to 'remote access over the Internet' and 'content scrambling'. They can be summarized as follows:

- a) When remote access is fully enabled (`control_remote_access_over_internet = 00`), content can be viewed, copied, moved and remotely accessed.
- b) When remote access is enabled within a managed domain (`control_remote_access_over_internet = 01`) content can be viewed, copied and moved locally only; and accessed remotely within a managed domain.
- c) If remote access is enabled within a managed domain (`control_remote_access_over_internet = 10`) only after 24 hours of the original broadcast to protect e.g. live transmissions, content can be viewed, copied and moved locally from the time of reception until 24 hours after reception; and then after the 24 hour delay time has expired may additionally be accessed remotely within a managed domain.
- d) When remote access is not permitted (`control_remote_access = 11`) content can be viewed, copied and moved locally only. Some content protection systems may be permitted to expire this prohibition of remote access within a managed domain after a very long time defined by their compliance regimes.

For cases b), c) and d), a managed domain is a distinguishable set of devices belonging to the same household, within which content usage can be controlled. This definition is subject to further refinement by compliance regimes. Likewise, 'local' means within the same immediate vicinity, approximating to the physical extent of a domicile or vehicle. This implies the need for devices to determine whether a device is local to another device prior to applying the related usage rules.

For cases b), c) and d), the transfer of content from/to removable storage medias is by default authorized locally. This removable storage media can be moved to another location and played, even if the playback device belongs to a different managed domain than the one from which the content was transferred to this removable storage media. If content on the removable media storage is bound to the managed domain, content can be further moved and copied within the managed domain. Otherwise, the copy of content shall be bound to the removable storage media and marked 'copy no more' once on the removable storage media.

By default, the signalling does not call for downscaling of content output over analogue interfaces.

When the scrambling after acquisition is enabled (`do_not_scramble = 0`), digital links shall be protected (e.g. DTCP, HDCP).

The FTA signalling information should remain with, and associated to, the content during its entire lifetime as a reminder of the broadcaster's intention.

Implementers are advised to consult possible additional usage and implementation restrictions (e.g. market/regulatory specific), which are not defined in the present document.

Table 57: FTA content management descriptor

| Syntax | Number of Bits | Identifier |
|---------------------------------------|----------------|------------|
| FTA_content_management_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| user_defined | 1 | bslbf |
| reserved_future_use | 3 | bslbf |
| do_not_scramble | 1 | uimsbf |
| control_remote_access_over_internet | 2 | uimsbf |
| do_not_apply_revocation | 1 | uimsbf |
| } | | |

Semantics for the FTA content management descriptor:

descriptor_tag: This 8-bit field shall be set to the value 0x7E.

descriptor_length: This 8-bit field specifies the total number of bytes of the data portion of the descriptor following the byte defining the value of this field.

user_defined: The semantics of this 1-bit field may be defined in specifications outside the scope of the present document. If no semantics are defined, it shall be set to zero.

NOTE: Since this flag appears inside a descriptor dealing with restrictions applicable to the usage of content, misinterpretation by the receiver should be avoided. Broadcasters intending to make use of this bit are thus encouraged to coordinate its use with other broadcasters and manufacturers. For Europe, use of this bit is coordinated by DVB.

do_not_scramble: This is a 1-bit field that indicates whether or not to apply scrambling of the content item for the purposes of content protection.

If do_not_scramble is set to '1' then scrambling shall not be applied for the purposes of content protection. If do_not_scramble is set to '0' then scrambling shall be applied where applicable for content protection.

The specification of 'where applicable' is not defined by the present document.

control_remote_access_over_internet: This is a 2-bit field. It indicates the policy regarding redistribution of the content item over the internet according to table 58.

Table 58: Coding of control_remote_access_over_internet

| Value | Meaning |
|-------|---|
| 00 | Redistribution over the Internet is enabled. |
| 01 | Redistribution over the Internet is enabled but only within a managed domain. |
| 10 | Redistribution over the Internet is enabled but only within a managed domain and after a certain short period of time (e.g. 24 hours). |
| 11 | Redistribution over the Internet is not allowed with the following exception: Redistribution over the Internet within a managed domain is enabled after a specified long (possibly indefinite) period of time. |

In the absence of a managed domain, the values '01' and '10' shall be interpreted as '11'. Hence, since there is no managed domain, this means redistribution over the internet is not allowed.

The specification of 'redistribution over the Internet', 'managed domain', 'certain short period of time' and 'specified long (possibly indefinite) period of time' are not defined by the present document.

do_not_apply_revocation: This 1-bit field indicates whether or not to apply the content revocation process to the content item.

If do_not_apply_revocation is set to '1' then the content revocation process shall not be applied. If do_not_apply_revocation is set to '0' then the content revocation process shall be applied.

The specification of 'content revocation process' is not defined by the present document.

6.2.18.1 Scope of the FTA content management descriptor

The location of a particular instance of the descriptor defines the scope over which the defined content management policy shall apply. This policy shall apply to all items of content within this scope unless overridden at a scope of greater precedence.

The content management policy for an item of content that falls outside of the scope of all explicit signalling is not defined by the present document.

The following scoping rules are listed in order of increasing precedence, each overriding the scope of the previous one:

- To define a content management policy with the scope of all DVB Services within a network, a single instance may be placed in the corresponding first descriptor loop of the NIT sub_table. If a NIT sub_table contains multiple sections then the descriptor shall only appear in the first descriptor loop of one section.
- To define a content management policy with the scope of all DVB Services within a bouquet, a single instance may be placed in the corresponding first descriptor loop of the BAT sub_table. If a BAT sub_table contains multiple sections then the descriptor shall only appear in the first descriptor loop of one section.
- To define a content management policy with the scope of all DVB Services within a transport stream of a particular network, a single instance may be placed in the corresponding transport stream descriptor loop of the NIT. The content management policy for a transport stream overrides the content management policy of the network (if defined).
- To define a content management policy with the scope of all DVB Services within a transport stream of a particular bouquet, a single instance may be placed in the corresponding transport stream descriptor loop of the BAT. The content management policy for a transport stream overrides the content management policy of the bouquet (if defined).
- To define a content management policy for a single DVB Service a single instance may be placed in the corresponding descriptor loop of the SDT. The content management policy for a service overrides the content management policy of the transport stream, bouquet or network (if defined).
- A content management policy for a single event within a DVB Service can only be defined for the present event. A single instance may be placed in the descriptor loop corresponding to the present event in EITp/f. The content management policy for an event overrides the content management policy of the service, transport stream, bouquet or network (if defined).

NOTE: A single instance may also be placed in each of the descriptor loops of an EIT sub_table but this will only be considered as informative for anything other than the present event of EITp/f.

The effect of defining a content management policy in a BAT that conflicts with a definition of equivalent scope in a NIT is not defined by the present document.

6.2.19 Linkage descriptor

6.2.19.0 Semantics and syntax of the linkage descriptor

The linkage descriptor (see table 59) identifies a service that can be presented if the consumer requests for additional information related to a specific entity described by the SI system. The location of the linkage descriptor in the syntax indicates the entity for which additional information is available. For example a linkage descriptor located within the NIT shall point to a service providing additional information on the network, a linkage descriptor in the BAT shall provide a link to a service informing about the bouquet, etc.

A CA replacement service can be identified using the linkage descriptor. This service may be selected automatically by the IRD if the CA denies access to the specific entity described by the SI system.

A service replacement service can also be identified using the linkage_descriptor. This replacement service may be selected automatically by the IRD when the running status of the current service is set to "not_running".

Table 59: Linkage descriptor

| Syntax | Number of bits | Identifier |
|------------------------|----------------|------------|
| linkage_descriptor() { | | |

| | | |
|--|----|--------|
| descriptor_tag | 8 | uimbsf |
| descriptor_length | 8 | uimbsf |
| transport_stream_id | 16 | uimbsf |
| original_network_id | 16 | uimbsf |
| service_id | 16 | uimbsf |
| linkage_type | 8 | uimbsf |
| if (linkage_type == 0x08){ mobile_hand-over_info() } else if (linkage_type == 0x0D){ event_linkage_info() } else if (linkage_type >= 0x0E && linkage_type <= 0x1F) { extended_event_linkage_info() } | | |
| for (i=0;i<N;i++){ private_data_byte } | 8 | bslbf |
| } | | |

Semantics for the linkage descriptor:

transport_stream_id: This is a 16-bit field which identifies the TS containing the information service indicated.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system of the information service indicated.

service_id: This is a 16-bit field which uniquely identifies an information service within a TS. The service_id is the same as the program_number in the corresponding program_map_section. If the linkage_type field has the value 0x04, then the service_id field is not relevant, and shall be set to 0x0000.

linkage_type: This is an 8-bit field specifying the type of linkage e.g. to information (see table 60).

Table 60: Linkage type coding

| linkage_type | Description |
|--|---|
| 0x00 | reserved for future use |
| 0x01 | information service |
| 0x02 | EPG service |
| 0x03 | CA replacement service |
| 0x04 | TS containing complete Network/Bouquet SI |
| 0x05 | service replacement service |
| 0x06 | data broadcast service |
| 0x07 | RCS Map |
| 0x08 | mobile hand-over |
| 0x09 | System Software Update Service (TS 102 006 [11]) |
| 0x0A | TS containing SSU BAT or NIT (TS 102 006 [11]) |
| 0x0B | IP/MAC Notification Service (EN 301 192 [4]) |
| 0x0C | TS containing INT BAT or NIT (EN 301 192 [4]) |
| 0x0D | event linkage (see note) |
| 0x0E to 0x1F | extended event linkage (see note) |
| 0x20 | downloadable font info linkage (DVB BlueBook A174 [56]) |
| 0x21 to 0x7F | reserved for future use |
| 0x80 to 0xFE | user defined |
| 0xFF | reserved for future use |
| NOTE: A linkage_type with a value in the range 0x0D to 0x1F is only valid when the descriptor is carried in the EIT. | |

mobile_hand-over_info(): This field shall be coded according to clause 6.2.19.1.

event_linkage_info(): This field shall be coded according to clause 6.2.19.2.

extended_event_linkage_info(): This field shall be coded according to clause 6.2.19.3.

private_data_byte: This is an 8-bit field, the value of which is privately defined.

6.2.19.1 Mobile hand-over linkage

A service to which a mobile receiver might hand-over to can also be identified using the `mobile_hand-over_info()` structure in the `linkage_descriptor`. This service may be selected automatically by the IRD when the actual service is no longer receivable under its `service_id`. The `hand-over_type` identifies whether the `linkage_descriptor` links to the same service in a different country, to a local variation of the service or an associated service.

The `mobile_hand-over_info()` shall be coded according to table 61.

Table 61: Mobile hand-over info

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| <code>mobile_hand-over_info() {</code> | | |
| <code>hand-over_type</code> | 4 | bslbf |
| <code>reserved_future_use</code> | 3 | bslbf |
| <code>origin_type</code> | 1 | bslbf |
| <code>if (hand-over_type == 0x01</code> <code> hand-over_type == 0x02</code> <code> hand-over_type == 0x03) {</code> | | |
| <code>network_id</code> | 16 | uimsbf |
| <code>}</code> | | |
| <code>if (origin_type == 0x00) {</code> | | |
| <code>initial_service_id</code> | 16 | uimsbf |
| <code>}</code> | | |
| <code>}</code> | | |

Semantics for the mobile hand-over info:

hand-over_type: This is a 4-bit field specifying the type of hand-over (see table 62).

Table 62: Hand-over type coding

| hand-over_type | Description |
|----------------|---|
| 0x00 | reserved for future use |
| 0x01 | DVB hand-over to an identical service in a neighbouring country |
| 0x02 | DVB hand-over to a local variation of the same service |
| 0x03 | DVB hand-over to an associated service |
| 0x04 to 0x0F | reserved for future use |

origin_type: This is a flag specifying in which table the link is originated (see table 63).

Table 63: Origin type coding

| origin_type | Description |
|-------------|-------------|
| 0x00 | NIT |
| 0x01 | SDT |

network_id: This is a 16-bit field which identifies the terrestrial network that supports the service indicated.

initial_service_id: This is a 16-bit field which identifies the service for which the hand-over linkage is valid.

6.2.19.2 Event linkage

Two events can be signalled as equivalent using the `event_linkage_info()` structure in the `linkage_descriptor`. The event being linked to may be a simulcast or may be time offset. The `event_simulcast` flag shall only be set if the target event is higher quality.

The `event_linkage_info()` shall be coded according to table 64.

Table 64: Event linkage info

| Syntax | Number of bits | Identifier |
|-------------------------------------|----------------|------------|
| <code>event_linkage_info() {</code> | | |

| | | |
|-----------------|----|--------|
| target_event_id | 16 | uimsbf |
| target_listed | 1 | bslbf |
| event_simulcast | 1 | bslbf |
| reserved | 6 | bslbf |
| } | | |

Semantics for the event linkage info:

target_event_id: This 16-bit field identifies the event_id of the event (the target event), carried on the service defined by the original_network_id, transport_stream_id and service_id, which is equivalent to the event identified by the location of this descriptor (the source event).

target_listed: This 1-bit field signals whether the service defined by the original_network_id, transport_stream_id and service_id is included in the SDT carried in that Transport Stream. When target_listed is set to 1 (one), the service shall be included in the SDT, otherwise it may not be. In the latter case, the following conditions shall be met:

- the service_type for the service shall be 0x19 (H.264/AVC HD digital television service) if the events are simulcast, otherwise the service_type shall be the same as for the service where the source event is carried;
- EIT_{p/f} information shall be available for the service in that Transport Stream;
- the service shall be running.

event_simulcast: This 1-bit field shall be set to 1 (one) when the target event and the source event are being simulcast. It shall be set to 0 (zero) when the events are offset in time.

6.2.19.3 Extended event linkage

Several events can be signalled as equivalent using the using the extended_event_linkage_info() structure in the linkage_descriptor. The event being linked to may be a simulcast or may be time offset.

The extended_event_linkage_info() shall be coded according to table 65.

Table 65: Extended event linkage info

| Syntax | Number of bits | Identifier |
|---|---|--|
| extended_event_linkage_info() { loop_length for (i=0; i<N; i++) { target_event_id target_listed event_simulcast link_type target_id_type original_network_id_flag service_id_flag if (target_id_type == 3) { user_defined_id } else { if (target_id_type == 1) { target_transport_stream_id } if (original_network_id_flag) { target_original_network_id } if (service_id_flag) { target_service_id } } } } | 8 16 1 1 2 2 1 1 16 16 16 16 | uimsbf uimsbf bslbf bslbf uimsbf uimsbf bslbf bslbf uimsbf uimsbf uimsbf |

Semantics for the extended event linkage info:

loop_length: This 8-bit field indicates the size in bytes of the following loop.

target_event_id: This 16-bit field identifies the event_id of the event (the target event), carried on the service defined by the original_network_id, transport_stream_id and service_id, which is equivalent to the event identified by the location of this descriptor (the source event).

target_listed: This 1-bit field signals whether the service defined by the original_network_id, transport_stream_id and service_id is included in the SDT carried in that Transport Stream. When target_listed is set to 1 (one), the service shall be included in the SDT, otherwise it may not be. In the latter case, the following conditions shall be met:

- the service_type for the service shall be 0x19 (H.264/AVC HD digital television service) if the events are simulcast, otherwise the service_type shall be the same as for the service where the source event is carried;
- EIT_{p/f} information shall be available for the service in that Transport Stream;
- the service shall be running.

event_simulcast: This 1-bit field shall be set to 1 (one) when the target event and the source event are being simulcast. It shall be set to 0 (zero) when the events are offset in time.

link_type: This 2-bit field indicates the type of the target service. It shall be coded according to table 66.

Table 66: Link type coding

| link_type | linkage_type (see note) | Type of target service |
|---------------------|----------------------------|---|
| 0 | 0x0E | SD |
| 1 | 0x0E | HD |
| 2 | 0x0E | frame compatible plano-stereoscopic H.264/AVC |
| 3 | 0x0E | service compatible plano-stereoscopic MVC |
| 0 | 0x0F | UHD |
| 1 | 0x0F | service frame compatible plano-stereoscopic |
| 2 to 3 | 0x0F | reserved for future use |
| 0 to 3 | 0x10 to 0x1F | reserved for future use |
| NOTE: See table 60. | | |

target_id_type: This 2-bit field, together with the original_network_id_flag, and the service_id_flag, identifies the target service or services. A value of zero indicates that the transport_stream_id shall be used to identify a single target service. A value of one indicates that the target_transport_stream_id shall be used instead of the transport_stream_id to identify a single target service. A value of two indicates that the target services can be in one or more transport streams (wildcarded TSid). A value of 3 indicates the target services are matched using a user defined identifier. The target_id_type field shall be coded according to table 67.

Table 67: Target id type coding

| target_id_type | How target service is matched |
|----------------|--|
| 0 | use transport_stream_id |
| 1 | use target_transport_stream_id |
| 2 | match any transport_stream_id (wildcard) |
| 3 | use user_defined_id |

Table 68 shows which identifiers shall be used for matching target services, controlled by the combination of the target_id_type, the original_network_id_flag, and the service_id_flag fields.

Table 68: Target service matching rules

| target_id_type | original_network_id_flag | service_id_flag | Identifiers used for matching the target service: | | | | | |
|-----------------|--------------------------|-----------------|---|---------------------|------------|----------------------------|----------------------------|-------------------|
| | | | transport_stream_id | original_network_id | service_id | target_transport_stream_id | target_original_network_id | target_service_id |
| 0 | 0 | 0 | X | X | X | | | |
| | | 1 | X | X | | | | X |
| | 1 | 0 | X | | X | | X | |
| | | 1 | X | | | | X | X |
| 1 | 0 | 0 | | X | X | X | | |
| | | 1 | | X | | X | | X |
| | 1 | 0 | | | X | X | X | |
| | | 1 | | | | X | X | X |
| 2 (see note) | 0 | 0 | | X | X | | | |
| | | 1 | | X | | | | X |
| | 1 | 0 | | | X | | X | |
| | | 1 | | | | | X | X |
| 3 | n/a | n/a | All services matched with user_defined_id | | | | | |

NOTE: When target_id_type is set to two, neither transport_stream_id, nor target_transport_stream_id are used for matching. Instead, all services with matching remaining identifiers as shown, are considered matches.

original_network_id_flag: This 1-bit flag, which when set to 1 (one) indicates that the target_original_network_id shall be used instead of the original_network_id to determine the target service. See also table 68.

service_id_flag: This 1-bit flag, which when set to 1 (one) indicates that the target_service_id shall be used instead of the service_id to determine the target service. See also table 68.

user_defined_id: If this 16-bit field is used, the linkage descriptor should be in the scope of a private data specifier descriptor, so the receiver can determine the meaning of the user_defined_id field.

target_transport_stream_id: This is a 16-bit field which identifies an alternate TS containing the information service indicated, under the control of the target_id_type, the original_network_id_flag, and the service_id_flag fields (see table 68).

target_original_network_id: This 16-bit field gives the label identifying the network_id of an alternate originating delivery system of the information service indicated, under the control of the target_id_type, the original_network_id_flag, and the service_id_flag fields (see table 68).

target_service_id: This is a 16-bit field which identifies an alternate information service indicated, under the control of the target_id_type, the original_network_id_flag, and the service_id_flag fields (see table 68).

6.2.20 Local time offset descriptor

The local time offset descriptor (see table 69) may be used in the TOT to describe country specific dynamic changes of the local time offset relative to UTC.

Table 69: Local time offset descriptor

| Syntax | Number of bits | Identifier |
|---|----------------|------------|
| <code>local_time_offset_descriptor() {</code> | | |
| <code>descriptor_tag</code> | 8 | uimsbf |
| <code>descriptor_length</code> | 8 | uimsbf |
| <code>for (i=0; i<N; i++) {</code> | | |
| <code>country_code</code> | 24 | bslbf |
| <code>country_region_id</code> | 6 | bslbf |
| <code>reserved</code> | 1 | bslbf |
| <code>local_time_offset_polarity</code> | 1 | bslbf |
| <code>local_time_offset</code> | 16 | bslbf |
| <code>time_of_change</code> | 40 | bslbf |
| <code>next_time_offset</code> | 16 | bslbf |
| <code>}</code> | | |
| <code>}</code> | | |

Semantics for the local time offset descriptor:

country_code: This 24-bit field identifies a country or group of countries. It shall be coded as three characters, where each character is coded into 8-bits according to ISO/IEC 8859-1 [17] and inserted in order into the field. Individual countries shall be identified by using the alpha-3 code as specified in ISO 3166 [35]. A group of countries shall be identified by the string representation of a decimal number in the range 900 to 999. These numbers shall be coded according to TS 101 162 [57]. Country codes for groups of countries shall be limited to those within a single time zone.

EXAMPLE: United Kingdom has 3-character code "GBR", which is coded as:
"0100 0111 0100 0010 0101 0010".

country_region_id: This 6-bit field identifies a zone according to table 70, in the country which is indicated by `country_code`.

This is set to "000000" when there are no different local time zones in the country.

Table 70: Coding of country_region_id

| Country_region_id | Description |
|--------------------|------------------------------------|
| 00 0000 | no time zone extension used |
| 00 0001 | time zone 1 (most easterly region) |
| 00 0010 | time zone 2 |
| | |
| 11 1100 | time zone 60 |
| 11 1101 to 11 1111 | reserved |

local_time_offset_polarity: This 1-bit information indicates the polarity of the following `local_time_offset` and `next_time_offset`. If this bit is set to "0" the polarity is positive and the local time is ahead of UTC. If this bit is set to "1" the polarity is negative and the local time is behind UTC.

local_time_offset: This 16-bit field contains the offset time from UTC at a time when current UTC time is early with respect to `time_of_change`. In conjunction with the `local_time_offset_polarity`, this indicates the time offset in the area which is indicated by the combination of `country_code` and `country_region_id`. These 16 bits are coded as 4 digits in 4-bit BCD in the order hour tens, hour, minute tens, and minutes.

time_of_change: This is a 40-bit field which specifies the date and time in MJD and UTC (see annex C), when the time change takes place. This 40-bit field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 digits in the 4-bit BCD.

next_time_offset: This 16-bit field contains the offset time from UTC at a time when current UTC time is equal to or after time_of_change. In conjunction with the local_time_offset_polarity this indicates the time offset in the area which is indicated by the combination of country_code and country_region_id. These 16-bits are coded as 4-digits in 4-bit BCD in the order hour tens, hour, minute tens and minutes.

6.2.21 Mosaic descriptor

A mosaic component is a collection of different video images to form a coded video component. The information is organized so that each specific information when displayed appears on a small area of a screen.

The mosaic descriptor gives a partitioning of a digital video component into elementary cells, the allocation of elementary cells to logical cells, and gives a link between the content of the logical cell and the corresponding information (e.g. bouquet, service, event, etc.), see table 71.

Table 71: Mosaic descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| mosaic_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| mosaic_entry_point | 1 | bslbf |
| number_of_horizontal_elementary_cells | 3 | uimsbf |
| reserved_future_use | 1 | bslbf |
| number_of_vertical_elementary_cells | 3 | uimsbf |
| for (i=0; i<N; i++) { | | |
| logical_cell_id | 6 | uimsbf |
| reserved_future_use | 7 | bslbf |
| logical_cell_presentation_info | 3 | uimsbf |
| elementary_cell_field_length | 8 | uimsbf |
| for (i=0; j<elementary_cell_field_length; j++) { | | |
| reserved_future_use | 2 | bslbf |
| elementary_cell_id | 6 | uimsbf |
| } | | |
| cell_linkage_info | 8 | uimsbf |
| if (cell_linkage_info ==0x01){ | | |
| bouquet_id | 16 | uimsbf |
| } | | |
| if (cell_linkage_info ==0x02){ | | |
| original_network_id | 16 | uimsbf |
| transport_stream_id | 16 | uimsbf |
| service_id | 16 | uimsbf |
| } | | |
| if (cell_linkage_info ==0x03){ | | |
| original_network_id | 16 | uimsbf |
| transport_stream_id | 16 | uimsbf |
| service_id | 16 | uimsbf |
| } | | |
| if (cell_linkage_info ==0x04){ | | |
| original_network_id | 16 | uimsbf |
| transport_stream_id | 16 | uimsbf |
| service_id | 16 | uimsbf |
| event_id | 16 | uimsbf |
| } | | |
| } | | |
| } | | |

Semantics for the Mosaic Descriptor:

mosaic_entry_point: This is a 1-bit field which when set to a value of "1" indicates that the mosaic is the highest mosaic in a hierarchy. A complete mosaic system could be organized in a tree structure, the flag being set to identify the entry point in the tree.

number_of_horizontal_elementary_cells: This 3-bit field indicates the number of cells of horizontal screen display, see table 72 for coding.

Table 72: Coding of horizontal_elementary_cells

| Value | Meaning |
|-------|-------------|
| 0x00 | one cell |
| 0x01 | two cells |
| 0x02 | three cells |
| 0x03 | four cells |
| 0x04 | five cells |
| 0x05 | six cells |
| 0x06 | seven cells |
| 0x07 | eight cells |

number_of_vertical_elementary_cells: This 3-bit field indicates the number of cells of vertical screen display, see table 73 for coding.

Table 73: Coding of vertical_elementary_cells

| Value | Meaning |
|-------|-------------|
| 0x00 | one cell |
| 0x01 | two cells |
| 0x02 | three cells |
| 0x03 | four cells |
| 0x04 | five cells |
| 0x05 | six cells |
| 0x06 | seven cells |
| 0x07 | eight cells |

logical_cell_id: This 6-bit field is coded in binary form. Different adjacent (see figure 3) elementary cells may be grouped together to form a logical cell. A logical_cell_number is associated to such a group of adjacent elementary_cell_ids. The total number of logical cells shall not exceed the number of elementary cells (maximum = 64). Each elementary cell shall be allocated to one logical cell.

More than one elementary cell may belong to one logical cell.

| | | |
|----------|----------|----------|
| A | B | C |
| D | E | F |
| G | H | I |

NOTE: Cells B, D, H, F are adjacent to cell E; C is not adjacent to A or D; D is not adjacent to H.

Figure 3: Adjacent cells

logical_cell_presentation_info: This 3-bit field identifies the type of presentation for a logical cell. The logical_cell_presentation information allows an identification of presentation styles, which are defined in table 74.

Table 74: Coding of logical_cell_presentation_info

| Value | Meaning |
|--|--------------------------|
| 0x00 | undefined |
| 0x01 | video |
| 0x02 | still picture (see note) |
| 0x03 | graphics/text |
| 0x04 to 0x07 | reserved for future use |
| NOTE: Still picture: A coded still picture consists of a video sequence containing exactly one coded picture which is intra-coded. | |

elementary_cell_field_length: The elementary_cell_field_length is an 8-bit field specifying the number of bytes following this field up to and including the last elementary_cell_id in this logical_cell_id loop.

elementary_cell_id: This 6-bit field indicates in binary form the number of the cell. The value of this field is in the range 0 to N.

NOTE: The elementary cells are implicitly numbered from 0 to N. The value 0 is allocated to the cell of the first row (top left corner). This number is incremented from left to right and from top to bottom in such a way that the number N is allocated to the cell of the last position of the last row (bottom right corner).

cell_linkage_info: This 8-bit field identifies the type of information carried in a logical cell, see table 75 for coding.

Table 75: Coding of cell_linkage_info

| Value | Meaning |
|--------------|-------------------------|
| 0x00 | undefined |
| 0x01 | bouquet related |
| 0x02 | service related |
| 0x03 | other mosaic related |
| 0x04 | event related |
| 0x05 to 0xFF | reserved for future use |

bouquet_id: This is a 16-bit field which serves as a label to identify the bouquet described by the cell.

original_network_id: This 16-bit field is a label (see clause 5.2) which in conjunction with the following fields uniquely identifies a service, event or mosaic.

transport_stream_id: This is a 16-bit field which serves as a label identifying the TS which contains the service, event or mosaic described by the cell.

service_id: This is a 16-bit field which identifies a service within a TS. The service_id is the same as the program_number in the corresponding program_map_section.

The interpretation of this field is context sensitive, dependent on the value of cell_linkage_info:

- when cell_linkage_info = "0x02", this is the service_id of the service described by the cell;
- when cell_linkage_info = "0x03", this is the service_id of the mosaic service described by the cell;
- when cell_linkage_info = "0x04", this is the service_id of the service to which the event described by the cell belongs.

event_id: This is a 16-bit field containing the identification number of the described event.

6.2.22 Multilingual bouquet name descriptor

The multilingual bouquet name descriptor (see table 76) provides the bouquet name in text form in one or more languages.

Table 76: Multilingual bouquet name descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| multilingual_bouquet_name_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| ISO_639_language_code | 24 | bslbf |
| bouquet_name_length | 8 | uimsbf |
| for (j=0; j<N; j++) { | | |
| char | 8 | uimsbf |
| } | | |
| } | | |
| } | | |

Semantics for the multilingual bouquet name descriptor:

ISO_639_language_code: This 24-bit field contains the ISO 639-2 [36] three character language code of the language of the following bouquet name. Both ISO 639-2/B and ISO 639-2/T may be used. Each character is coded into 8 bits according to ISO/IEC 8859-1 [17] 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".

bouquet_name_length: This 8-bit field specifies the length in bytes of the following bouquet name.

char: This is an 8-bit field. A string of char fields specify the name of the bouquet about which the BAT sub_table informs in the language specified. Text information is coded using the character sets and methods described in annex A.

6.2.23 Multilingual component descriptor

The multilingual component descriptor (see table 77) provides a text description of a component in one or more languages. The component is identified by its component tag value.

Table 77: Multilingual component descriptor

| Syntax | Number of bits | Identifier |
|---------------------------------------|----------------|------------|
| multilingual_component_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| component_tag | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| ISO_639_language_code | 24 | bslbf |
| text_description_length | 8 | uimsbf |
| for (j=0; j<N; j++) { | | |
| text_char | 8 | uimsbf |
| } | | |
| } | | |
| } | | |

Semantics for the multilingual component descriptor:

component_tag: This 8-bit field has the same value as the component_tag field in the stream identifier descriptor (if present in the PSI program map section) for the component stream.

ISO_639_language_code: This 24-bit field identifies the language of the following text description of the component. The ISO_639_language_code contains a 3-character code as specified by ISO 639-2 [36]. Both ISO 639-2/B and ISO 639-2/T may be used. Each character is coded into 8 bits according to ISO/IEC 8859-1 [17] 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".

text_description_length: This 8-bit field specifies the length in bytes of the following text description.

text_char: This is an 8-bit field. A string of "text_char" fields specifies a text description of the component stream. Text information is coded using the character sets and methods described in annex A.

6.2.24 Multilingual network name descriptor

The multilingual network name descriptor (see table 78) provides the network name in text form in one or more languages.

Table 78: Multilingual network name descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| multilingual_network_name_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| ISO_639_language_code | 24 | bslbf |
| network_name_length | 8 | uimsbf |

| | | |
|--|---|--------|
| <pre> for (j=0; j<N; j++) { char } } </pre> | 8 | uimbsf |
|--|---|--------|

Semantics for the multilingual network name descriptor:

ISO_639_language_code: This 24-bit field contains the ISO 639-2 [36] three character language code of the language of the following network name. Both ISO 639-2/B and ISO 639-2/T may be used. Each character is coded into 8 bits according to ISO/IEC 8859-1 [17] 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".

network_name_length: This 8-bit field specifies the length in bytes of the following network name.

char: This is an 8-bit field. A string of char fields specify the name of the network about which the NIT informs in the language specified. Text information is coded using the character sets and methods described in annex A.

6.2.25 Multilingual service name descriptor

The multilingual service name descriptor (see table 79) provides the names of the service provider and service in text form in one or more languages.

Table 79: Multilingual service name descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| multilingual_service_name_descriptor() { | | |
| descriptor_tag | 8 | uimbsf |
| descriptor_length | 8 | uimbsf |
| for (i=0; i<N; i++) { | | |
| ISO_639_language_code | 24 | bslbf |
| service_provider_name_length | 8 | uimbsf |
| for (j=0; j<N; j++) { | | |
| char | 8 | uimbsf |
| } | | |
| service_name_length | 8 | uimbsf |
| for (j=0; j<N; j++) { | | |
| char | 8 | uimbsf |
| } | | |
| } | | |
| } | | |

Semantics for the multilingual service name descriptor:

ISO_639_language_code: This 24-bit field contains the ISO 639-2 [36] three character language code of the language of the following text fields. Both ISO 639-2/B and ISO 639-2/T may be used. Each character is coded into 8 bits according to ISO/IEC 8859-1 [17] 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".

service_provider_name_length: This 8-bit field specifies the length in bytes of the following service provider name.

service_name_length: This 8-bit field specifies the length in bytes of the following service name.

char: This is an 8-bit field. A string of char fields specify the name of the service provider or service. Text information is coded using the character sets and methods described in annex A.

6.2.26 Near Video On Demand (NVOD) reference descriptor

This descriptor, in conjunction with the time shifted service and time shifted event descriptors, provides a mechanism for efficiently describing a number of services which carry the same sequence of events, but with the start times offset

from one another. Such a group of time-shifted services is referred to as Near Video On Demand, since a user can at any time access near to the start of an event by selecting the appropriate service of the group.

The NVOD reference descriptor (see table 80) gives a list of the services which together form a NVOD service. Each service is also described in the appropriate SDT sub_table by a time shifted service descriptor, see clause 6.2.44. The time shifted service descriptor associates a time shifted service with a reference_service_id. The reference_service_id is the label under which a full description of the NVOD service is given, but the reference_service_id does not itself correspond to any program_number in the program_map_section.

The time shifted event descriptor is used in the event information for each time shifted service. Instead of duplicating the full information for each event, the time shifted event descriptor points to a reference_event_id in the reference service. The full event information is provided in the event information for the reference service.

The services which make up an NVOD service need not all be carried in the same TS. However, a reference service shall be described in the SI in each TS which carries any services of the NVOD service.

Table 80: NVOD reference descriptor

| Syntax | Number of bits | Identifier |
|-------------------------------|----------------|------------|
| NVOD_reference_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0;i<N;i++) { | | |
| transport_stream_id | 16 | uimsbf |
| original_network_id | 16 | uimsbf |
| service_id | 16 | uimsbf |
| } | | |
| } | | |

Semantics for the NVOD reference descriptor:

transport_stream_id: This is a 16-bit field which identifies the TS.

original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

service_id: This is a 16-bit field which uniquely identifies a service within a TS. The service_id is the same as the program_number in the corresponding program_map_section.

6.2.27 Network name descriptor

The network name descriptor provides the network name in text form (see table 81).

Table 81: Network name descriptor

| Syntax | Number of bits | Identifier |
|-----------------------------|----------------|------------|
| network_name_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0;i<N;i++) { | | |
| char | 8 | uimsbf |
| } | | |
| } | | |

Semantics for the network name descriptor:

char: This is an 8-bit field. A string of char fields specify the name of the delivery system about which the NIT informs. Text information is coded using the character sets and methods described in annex A.

6.2.28 Parental rating descriptor

This descriptor (see table 82) gives a rating based on age and allows for extensions based on other rating criteria.

Table 82: Parental rating descriptor

| Syntax | Number of bits | Identifier |
|--------------------------------|----------------|------------|
| parental_rating_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0;i<N;i++){ | | |
| country_code | 24 | bslbf |
| rating | 8 | uimsbf |
| } | | |
| } | | |

Semantics for the parental rating descriptor:

country_code: This 24-bit field identifies a country or group of countries. It shall be coded as three characters, where each character is coded into 8-bits according to ISO/IEC 8859-1 [17] and inserted in order into the field. Individual countries shall be identified by using the alpha-3 code as specified in ISO 3166 [35]. A group of countries shall be identified by the string representation of a decimal number in the range 900 to 999. These numbers shall be coded according to TS 101 162 [57].

EXAMPLE 1: United Kingdom has 3-character code "GBR", which is coded as:
"0100 0111 0100 0010 0101 0010".

rating: This 8-bit field is coded according to table 83, giving the recommended minimum age in years of the end user.

Table 83: Parental rating descriptor, rating

| Rating | Description |
|--------------|--------------------------------|
| 0x00 | undefined |
| 0x01 to 0x0F | minimum age = rating + 3 years |
| 0x10 to 0xFF | defined by the broadcaster |

EXAMPLE 2: 0x04 implies that end users should be at least 7 years old.

6.2.29 Partial Transport Stream (TS) descriptor

See clause 7.1.2.

6.2.30 PDC descriptor

The PDC-descriptor (see table 84) extends the DVB system with the functionalities of PDC (EN 300 231 [1]). The descriptor carries the Programme Identification Label (PIL) as defined in EN 300 231 [1]. The PIL contains date and time of the first published start time of a certain event.

Table 84: PDC_descriptor

| Syntax | Number of bits | Identifier |
|--------------------------------|----------------|------------|
| PDC_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| reserved_future_use | 4 | bslbf |
| programme_identification_label | 20 | bslbf |
| } | | |

Semantics for the PDC descriptor:

programme_identification_label: This 20-bit field gives the Programme Identification Label (EN 300 231 [1]). The structure of the Programme Identification Label (PIL) is as follows (bit number 1 is the leftmost bit of the string).

| Programme identification label | | | | | | | | | | | | | | | | | | | |
|--------------------------------|---|---|---|---|-------|---|---|---|----|------|----|----|----|----|--------|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| M | | | | L | M | | | L | M | | | | L | M | | | | | L |
| day | | | | | month | | | | | hour | | | | | minute | | | | |

Figure 4: Programme identification label

The values for the day, month, hour (24) and minute are binary coded.

6.2.31 Private data specifier descriptor

This descriptor (see table 85) is used to identify the specifier of any private descriptors or private fields within descriptors.

Table 85: Private data specifier descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| <code>private_data_specifier_descriptor() {</code> | | |
| <code>descriptor_tag</code> | 8 | uimsbf |
| <code>descriptor_length</code> | 8 | uimsbf |
| <code>private_data_specifier</code> | 32 | uimsbf |
| <code>}</code> | | |

Semantics for the private data specifier descriptor:

private_data_specifier: This field shall be coded according to TS 101 162 [57].

6.2.32 Scrambling descriptor

The scrambling descriptor (see table 86) indicates the selected mode of operation for the scrambling system. It is located in the program map section at the program loop level. For further clarifications on the usage of the scrambling descriptor refer to annex E.

Table 86: Scrambling_descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| <code>scrambling_descriptor() {</code> | | |
| <code>descriptor_tag</code> | 8 | uimsbf |
| <code>descriptor_length</code> | 8 | uimsbf |
| <code>scrambling_mode</code> | 8 | uimsbf |
| <code>}</code> | | |

Semantics for the scrambling_descriptor:

scrambling_mode: This 8-bit field identifies the selected mode of the scrambling algorithm (see table 87). The technical details of the scrambling algorithm are available only to bona-fide users upon signature of a Non Disclosure Agreement (NDA) administered by the DVB Common Scrambling Algorithm Custodian.

Table 87: scrambling_mode coding

| scrambling_mode | Description |
|-----------------|---|
| 0x00 | Reserved for future use |
| 0x01 | This value indicates use of DVB-CSA1. It is the default mode and shall be used when the scrambling descriptor is not present in the program map section |
| 0x02 | This value indicates use of DVB-CSA2 |
| 0x03 | This value indicates use of DVB-CSA3 |
| 0x04 to 0x0F | Reserved for future use |
| 0x10 | This value indicates use of DVB-CISSA version 1 |
| 0x11 to 0x1F | Reserved for future DVB-CISSA versions |
| 0x20 to 0x6F | Reserved for future use |
| 0x70 to 0x7F | ATIS defined (see annex J of ATIS-0800006) |
| 0x80 to 0xFE | User defined |
| 0xFF | Reserved for future use |

6.2.33 Service descriptor

The service descriptor (see table 88) provides the names of the service provider and the service in text form together with the service_type.

Note that the use of the HEVC digital television service type (0x1F) and HEVC digital television service type with HDR and/or a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz (0x20) listed in table 89 implies additional semantics and is subject to additional rules (see also note 4 and 5 in table 89).

Table 88: Service descriptor

| Syntax | Number of bits | Identifier |
|------------------------------|----------------|------------|
| service_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| service_type | 8 | uimsbf |
| service_provider_name_length | 8 | uimsbf |
| for (i=0;i<N;i++){ | | |
| char | 8 | uimsbf |
| } | | |
| service_name_length | 8 | uimsbf |
| for (i=0;i<N;i++){ | | |
| char | 8 | uimsbf |
| } | | |
| } | | |

Semantics for the service descriptor:

service_type: This is an 8-bit field specifying the type of the service. The assignment of service_type value for a service is described in annex I. It shall be coded according to table 89.

Table 89: Service type coding

| service_type | Description |
|---|---|
| 0x00 | reserved for future use |
| 0x01 | digital television service (see note 1) |
| 0x02 | digital radio sound service (see note 2) |
| 0x03 | Teletext service |
| 0x04 | NVOD reference service (see note 1) |
| 0x05 | NVOD time-shifted service (see note 1) |
| 0x06 | mosaic service |
| 0x07 | FM radio service |
| 0x08 | DVB SRM service [41] |
| 0x09 | reserved for future use |
| 0x0A | advanced codec digital radio sound service |
| 0x0B | H.264/AVC mosaic service |
| 0x0C | data broadcast service |
| 0x0D | reserved for Common Interface Usage (EN 50221 [31]) |
| 0x0E | RCS Map (see EN 301 790 [7]) |
| 0x0F | RCS FLS (see EN 301 790 [7]) |
| 0x10 | DVB MHP service |
| 0x11 | MPEG-2 HD digital television service |
| 0x12 to 0x15 | reserved for future use |
| 0x16 | H.264/AVC SD digital television service |
| 0x17 | H.264/AVC SD NVOD time-shifted service |
| 0x18 | H.264/AVC SD NVOD reference service |
| 0x19 | H.264/AVC HD digital television service |
| 0x1A | H.264/AVC HD NVOD time-shifted service |
| 0x1B | H.264/AVC HD NVOD reference service |
| 0x1C | H.264/AVC frame compatible plano-stereoscopic HD digital television service (see note 3) |
| 0x1D | H.264/AVC frame compatible plano-stereoscopic HD NVOD time-shifted service (see note 3) |
| 0x1E | H.264/AVC frame compatible plano-stereoscopic HD NVOD reference service (see note 3) |
| 0x1F | HEVC digital television service (see note 4) |
| 0x20 | HEVC UHD digital television service with HDR and/or a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz, or a any combination of HDR and these frame rates (see note 5) |
| 0x21 to 0x7F | reserved for future use |
| 0x80 to 0xFE | user defined |
| 0xFF | reserved for future use |
| NOTE 1: MPEG-2 SD material should use this type. | |
| NOTE 2: MPEG-1 Layer 2 audio material should use this type. | |
| NOTE 3: For information on the use of these values, see clause I.2.3 and TS 101 547-2 [46] (3D Guidelines of Frame Compatible 3D-TV). | |
| NOTE 4: For rules on the use of this value, see clause I.2.5 and TS 101 547-4 [53]. This value should be used for backward compatible HLG10 HDR services, and/or backward compatible high frame rate (HFR) services which are decodable by HEVC_UHDTV_IRD as defined in TS 101 154 [9], see clause I.2.5.2. | |
| NOTE 5: For rules on the use of these values, see clause I.2.6 | |

service_provider_name_length: This 8-bit field specifies the number of bytes that follow the service_provider_name_length field for describing characters of the name of the service provider.

char: This is an 8-bit field. A string of char fields specify the name of the service provider or service. Text information is coded using the character sets and methods described in annex A.

service_name_length: This 8-bit field specifies the number of bytes that follow the service_name_length field for describing characters of the name of the service.

6.2.34 Service availability descriptor

This descriptor (see table 90) may be used in the SDT in a terrestrial network. It provides an identification of the cells in which the service is available or not available.

Table 90: Service availability descriptor

| Syntax | Number of bits | Identifier |
|---|------------------------|--|
| service_availability_descriptor() { descriptor_tag descriptor_length availability_flag reserved for (i=0; i<N; i++) { cell_id } } | 8 8 1 7 16 | uimsbf uimsbf bslbf bslbf uimsbf |

Semantics for the service availability descriptor:

availability_flag: This 1-bit indicator, when set to "1" indicates that the service is available on the cells identified by the cell_ids in the following loop and not available on the other cells. When the bit is set to "0", it indicates that the service is unavailable on the cells identified by the cell_ids in the following loop and available on the other cells.

cell_id: This is a 16-bit field which identifies a cell in the terrestrial network.

6.2.35 Service list descriptor

The service list descriptor (see table 91) provides a means of listing the services by service_id and service type.

Note that the use of the HEVC digital television service type listed in table 89 implies additional semantics and is subject to additional rules (see also note 4 in table 89).

Table 91: Service list descriptor

| Syntax | Number of bits | Identifier |
|---|-------------------|--------------------------------------|
| service_list_descriptor() { descriptor_tag descriptor_length for (i=0; i<N; i++) { service_id service_type } } | 8 8 16 8 | uimsbf uimsbf uimsbf uimsbf |

Semantics for the service list descriptor:

service_id: This is a 16-bit field which uniquely identifies a service within a TS. The service_id is the same as the program_number in the corresponding program_map_section, except that in the case of service_type = 0x04, 0x18 or 0x1B (NVD reference services) the service_id does not have a corresponding program_number.

service_type: This is an 8-bit field specifying the type of the service. The assignment of service_type value for a service is described in annex I. It shall be coded according to table 89.

6.2.36 Service move descriptor

If it is required to move a service from one TS to another, a mechanism is provided which enables an IRD to track the service between TSs by means of a service_move_descriptor. It shall be coded according to table 92.

Table 92: Service move descriptor

| Syntax | Number of bits | Identifier |
|---|--------------------|--------------------------------------|
| service_move_descriptor() { descriptor_tag descriptor_length new_original_network_id new_transport_stream_id } | 8 8 16 16 | uimsbf uimsbf uimsbf uimsbf |

| | | |
|-----------------------------|----|--------|
| <code>new_service_id</code> | 16 | uimsbf |
| } | | |

Semantics for the service move descriptor:

new_original_network_id: This field contains the `original_network_id` of the TS in which the service is found after the move.

new_transport_stream_id: This field contains the `transport_stream_id` of the TS in which the service is found after the move.

new_service_id: This field contains the `service_id` of the service after the move.

6.2.37 Short event descriptor

The short event descriptor provides the name of the event and a short description of the event in text form (table 93).

Table 93: Short event descriptor

| Syntax | Number of bits | Identifier |
|---|----------------|------------|
| <code>short_event_descriptor() {</code> | | |
| <code>descriptor_tag</code> | 8 | uimsbf |
| <code>descriptor_length</code> | 8 | uimsbf |
| <code>ISO_639_language_code</code> | 24 | bslbf |
| <code>event_name_length</code> | 8 | uimsbf |
| for (<code>i=0;i<event_name_length;i++</code>) { | | |
| <code>event_name_char</code> | 8 | uimsbf |
| } | | |
| <code>text_length</code> | 8 | uimsbf |
| for (<code>i=0;i<text_length;i++</code>) { | | |
| <code>text_char</code> | 8 | uimsbf |
| } | | |
| <code>}</code> | | |

Semantics for the short event descriptor:

ISO_639_language_code: This 24-bit field contains the ISO 639-2 [36] three character language code of the language of the following text fields. Both ISO 639-2/B and ISO 639-2/T may be used. Each character is coded into 8 bits according to ISO/IEC 8859-1 [17] 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".

event_name_length: An 8-bit field specifying the length in bytes of the event name.

event_name_char: This is an 8-bit field. A string of "char" fields specifies the event name. Text information is coded using the character sets and methods described in annex A.

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

text_char: This is an 8-bit field. A string of "char" fields specify the text description for the event. Text information is coded using the character sets and methods described in annex A.

6.2.38 Short smoothing buffer descriptor

A `smoothing_buffer_descriptor` is specified in ISO/IEC 13818-1 [15] which enables the bit-rate of a service to be signalled in the PSI.

For use in DVB SI Tables, a more compact and efficient descriptor, the `short_smoothing_buffer_descriptor`, is defined here in table 94.

This descriptor may be included in the EIT Present/Following and EIT Schedule Tables to signal the bit-rate for each event.

The bit-rate is expressed in terms of a smoothing buffer size and output leak rate.

The presence of the descriptor in the EIT Present/Following and EIT Schedule Tables is optional.

The data flows into and from the smoothing buffer are defined as follows:

- bytes of TS packets belonging to the associated service are input to the smoothing buffer at the time defined by equation 2-4 of ISO/IEC 13818-1 [15] (definition of the mathematical byte delivery schedule). The following packets belong to the service:
 - all TS packets of all elementary streams of the service, i.e. all PIDs which are listed as elementary_PIDs in the extended program information part of the PMT section for the service during the time that the event is transmitted;
 - all TS packets of the PID which is identified as the program_map_PID for the service in the PAT at the time that the event is transmitted;
 - all TS packets of the PID which is identified as the PCR_PID in the PMT section for the service at the time that the event is transmitted.
- all bytes that enter the buffer also exit it.

Table 94: Short smoothing buffer descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| <code>short_smoothing_buffer_descriptor() {</code> | | |
| <code>descriptor_tag</code> | 8 | uimbsf |
| <code>descriptor_length</code> | 8 | uimbsf |
| <code>sb_size</code> | 2 | uimbsf |
| <code>sb_leak_rate</code> | 6 | uimbsf |
| for (i=0; i<N; i++) { | | |
| <code>DVB_reserved</code> | 8 | bslbf |
| } | | |
| <code>}</code> | | |

Semantics for the short smoothing buffer descriptor:

sb_size: This 2-bit field indicates the size of the smoothing buffer, and is coded according to table 95.

Table 95: Smoothing buffer size

| Value | Buffer size (bytes) |
|-------|---------------------|
| 0 | DVB_reserved |
| 1 | 1 536 |
| 2 | DVB_reserved |
| 3 | DVB_reserved |

NOTE: Due to implementation constraints, the specified buffer size value considers spare capacity that may be required in a 2 kbyte RAM for packet jitter.

sb_leak_rate: This 6-bit field indicates the value of the leak rate from the buffer, and is coded according to table 96.

Table 96: Smoothing buffer leak rate

| Value | Leak rate (Mbit/s) |
|-------|--------------------|
| 0 | DVB_reserved |
| 1 | 0,0009 |
| 2 | 0,0018 |
| 3 | 0,0036 |
| 4 | 0,0072 |
| 5 | 0,0108 |
| 6 | 0,0144 |
| 7 | 0,0216 |
| 8 | 0,0288 |
| 9 | 0,075 |
| 10 | 0,5 |

| Value | Leak rate (Mbit/s) |
|----------|---|
| 11 | 0,5625 |
| 12 | 0,8437 |
| 13 | 1,0 |
| 14 | 1,1250 |
| 15 | 1,5 |
| 16 | 1,6875 |
| 17 | 2,0 |
| 18 | 2,2500 |
| 19 | 2,5 |
| 20 | 3,0 |
| 21 | 3,3750 |
| 22 | 3,5 |
| 23 | 4,0 |
| 24 | 4,5 |
| 25 | 5,0 |
| 26 | 5,5 |
| 27 | 6,0 |
| 28 | 6,5 |
| 29 | 6,7500 |
| 30 to 32 | $((\text{value}) - 16) \times 0,5$ (7,0 Mbit/s, 7,5 Mbit/s, 8,0 Mbit/s) |
| 33 to 37 | $((\text{value}) - 24)$ (9 Mbit/s, 10 Mbit/s, 11 Mbit/s, 12 Mbit/s, 13 Mbit/s) |
| 38 | 13,5 |
| 39 to 43 | $((\text{value}) - 25)$ (14 Mbit/s, 15 Mbit/s, 16 Mbit/s, 17 Mbit/s, 18 Mbit/s) |
| 44 to 47 | $((\text{value}) - 34) \times 2$ (20 Mbit/s, 22 Mbit/s, 24 Mbit/s, 26 Mbit/s) |
| 48 | 27 |
| 49 to 55 | $((\text{value}) - 35) \times 2$ (28 Mbit/s, 30 Mbit/s, 32 Mbit/s to 40 Mbit/s) |
| 56 | 44 |
| 57 | 48 |
| 58 | 54 |
| 59 | 72 |
| 60 | 108 |
| 61 to 63 | DVB_reserved |

6.2.39 Stream identifier descriptor

The stream identifier descriptor (see table 97) may be used in the PSI PMT to label component streams of a service so that they can be differentiated, e.g. by text descriptions given in component descriptors in the EIT if present. The stream identifier descriptor shall be located within the relevant ES_info descriptor loop of the PMT.

Table 97: Stream identifier descriptor

| Syntax | Number of bits | Identifier |
|---|----------------|------------|
| <code>stream_identifier_descriptor() {</code> | | |
| <code>descriptor_tag</code> | 8 | uimsbf |
| <code>descriptor_length</code> | 8 | uimsbf |
| <code>component_tag</code> | 8 | uimsbf |
| <code>}</code> | | |

Semantics for the stream identifier descriptor:

component_tag: This 8-bit field identifies the component stream for associating it with a description given in a component descriptor. Within a program map section each stream identifier descriptor shall have a different value for this field.

6.2.40 Stuffing descriptor

The stuffing descriptor provides a means of invalidating previously coded descriptors or inserting dummy descriptors for table stuffing (see table 98).

Table 98: Stuffing descriptor

| Syntax | Number of bits | Identifier |
|-------------------------|----------------|------------|
| stuffing_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i= 0;i<N;i++){ | | |
| stuffing_byte | 8 | bslbf |
| } | | |
| } | | |

Semantics for the stuffing descriptor:

stuffing_byte: This is an 8-bit field. Each occurrence of the field may be set to any value. The IRDs may discard the stuffing bytes.

6.2.41 Subtitling descriptor

In the ISO/IEC 13818-1 [15] Program Map Table (PMT) the value of stream_type for any PID carrying DVB subtitle [55] shall be "0x06" (this indicates a PES carrying private data). See table 99.

Table 99: Subtitling descriptor

| Syntax | Number of bits | Identifier |
|---------------------------|----------------|------------|
| subtitling_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i= 0;i<N;i++){ | | |
| ISO_639_language_code | 24 | bslbf |
| subtitling_type | 8 | bslbf |
| composition_page_id | 16 | bslbf |
| ancillary_page_id | 16 | bslbf |
| } | | |
| } | | |

Semantics for the subtitling descriptor:

ISO_639_language_code: This 24-bit field contains the ISO 639-2 [36] three character language code of the language of the subtitle. Both ISO 639-2/B and ISO 639-2/T may be used. Each character is coded into 8 bits according to ISO/IEC 8859-1 [17] 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".

subtitling_type: This 8 bit field provides information on the content of the subtitle and the intended display. The coding of this field shall use the codes defined for component_type when stream_content is 0x03 in table 26 "stream_content and component_type".

composition_page_id: This 16-bit field identifies the composition page. DVB_subtitling_segments signalling this page_id shall be decoded if the previous data in the subtitling descriptor matches the user's selection criteria.

NOTE 1: The composition_page_id is signalled in at least the DVB_subtitling_segments that define the data structure of the subtitle screen; the page_composition_segment and region_composition_segments. It may additionally be signalled in segments containing data on which the composition depends.

ancillary_page_id: This identifies the (optional) ancillary page. DVB_subtitling_segments signalling this page_id shall also be decoded if the previous data in the subtitling descriptor matches the user's selection criteria. The values in the ancillary_page_id and the composition_page_id fields shall be the same if no ancillary page is provided.

NOTE 2: The ancillary_page_id is never signalled in a composition segment. It may be signalled in Colour Look-Up Table (CLUT) definition segments, object segments and any other type of segment.

NOTE 3: (Terminology): A segment that signals a particular page number in its page_id field is said to be "in" that page. The page is said to "contain" that segment.

6.2.42 Telephone descriptor

The telephone descriptor may be used to indicate a telephone number which may be used in conjunction with a modem (PSTN or cable) to exploit narrowband interactive channels. Further information is given in TS 102 201 [i.4].

The telephone descriptor syntax is specified in table 100.

Table 100: Telephone descriptor

| Syntax | Number of bits | Identifier |
|--------------------------------|----------------|------------|
| telephone_descriptor() { | | |
| descriptor_tag | 8 | uimbsbf |
| descriptor_length | 8 | uimbsbf |
| reserved_future_use | 2 | bslbf |
| foreign_availability | 1 | bslbf |
| connection_type | 5 | uimbsbf |
| reserved_future_use | 1 | bslbf |
| country_prefix_length | 2 | uimbsbf |
| international_area_code_length | 3 | uimbsbf |
| operator_code_length | 2 | uimbsbf |
| reserved_future_use | 1 | bslbf |
| national_area_code_length | 3 | uimbsbf |
| core_number_length | 4 | uimbsbf |
| for (i=0;i<N;i++){ | | |
| country_prefix_char | 8 | uimbsbf |
| } | | |
| for (i=0;i<N;i++){ | | |
| international_area_code_char | 8 | uimbsbf |
| } | | |
| for (i=0;i<N;i++){ | | |
| operator_code_char | 8 | uimbsbf |
| } | | |
| for (i=0;i<N;i++){ | | |
| national_area_code_char | 8 | uimbsbf |
| } | | |
| for (i=0;i<N;i++){ | | |
| core_number_char | 8 | uimbsbf |
| } | | |
| } | | |

Semantics for the telephone descriptor:

foreign_availability: This is a 1-bit flag. When set to "1" it indicates that the number described can be called from outside of the country specified by the country_prefix. When set to "0" it indicates that the number can only be called from inside the country specified by the country_prefix.

connection_type: This is a 5-bit field which indicates connection types. One example of the use of the connection type is to inform the IRD that when, if an interaction is initiated, if the connection is not made within 1 minute, then the connection attempt should be aborted.

country_prefix_length: This 2-bit field specifies the number of 8-bit alphanumeric characters in the country prefix.

international_area_code_length: This 3-bit field specifies the number of 8-bit alphanumeric characters in the international area code.

operator_code_length: This 2-bit field specifies the number of 8-bit alphanumeric characters in the operator code.

national_area_code_length: This 3-bit field specifies the number of 8-bit alphanumeric characters in the national area code.

core_number_length: This 4-bit field specifies the number of 8-bit alphanumeric characters in the core number.

country_prefix_char: This 8-bit field which shall be coded in accordance with ISO/IEC 8859-1 [17] gives one alphanumeric character of the country prefix.

international_area_code_char: This 8-bit field which shall be coded in accordance with ISO/IEC 8859-1 [17] gives one alphanumeric character of the international area code.

operator_code_char: This 8-bit field which shall be coded in accordance with ISO/IEC 8859-1 [17] gives one alphanumeric character of the operator code.

national_area_code_char: This 8-bit field which shall be coded in accordance with ISO/IEC 8859-1 [17] gives one alphanumeric character of the national area code.

core_number_char: This 8-bit field which shall be coded in accordance with ISO/IEC 8859-1 [17] gives one alphanumeric character of the core number.

6.2.43 Teletext descriptor

The Teletext descriptor (see table 101) shall be used in the PSI PMT to identify streams which carry EBU Teletext data. The descriptor shall be located within the relevant ES_info descriptor loop of the PMT.

Table 101: Teletext descriptor

| Syntax | Number of bits | Identifier |
|--------------------------|----------------|------------|
| teletext_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0;i<N;i++) { | | |
| ISO_639_language_code | 24 | bslbf |
| teletext_type | 5 | uimsbf |
| teletext_magazine_number | 3 | uimsbf |
| teletext_page_number | 8 | uimsbf |
| } | | |
| } | | |

Semantics for the Teletext descriptor:

ISO_639_language_code: This 24-bit field contains the 3 character ISO 639-2 [36] language code of the language of the teletext. Both ISO 639-2/B and ISO 639-2/T may be used. Each character is coded into 8 bits according to ISO/IEC 8859-1 [17] 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".

teletext_type: This 5-bit field indicates the type of Teletext page indicated. This shall be coded according to table 102.

Table 102: Teletext descriptor, teletext_type

| Teletext_type | Description |
|---------------|--|
| 0x00 | reserved for future use |
| 0x01 | initial Teletext page |
| 0x02 | Teletext subtitle page |
| 0x03 | additional information page |
| 0x04 | programme schedule page |
| 0x05 | Teletext subtitle page for hearing impaired people |
| 0x06 to 0x1F | reserved for future use |

teletext_magazine_number: This is a 3-bit field which identifies the magazine number as defined in EN 300 706 [3].

teletext_page_number: This is an 8-bit field giving two 4-bit hex digits identifying the page number as defined in EN 300 706 [3].

6.2.44 Time shifted event descriptor

The time shifted event descriptor (see table 103) is used in place of the short_event_descriptor to indicate an event which is a time shifted copy of another event.

Table 103: Time shifted event descriptor

| Syntax | Number of bits | Identifier |
|-----------------------------------|----------------|------------|
| time_shifted_event_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| reference_service_id | 16 | uimsbf |
| reference_event_id | 16 | uimsbf |
| } | | |

Semantics for the time shifted event descriptor:

reference_service_id: This 16-bit field identifies the reference service of a NVOD collection of services. The reference service can always be found in this TS. The service_id here does not have a corresponding program_number in the program_map_section.

reference_event_id: This 16-bit field identifies the reference event of which the event described by this descriptor is a time shifted-copy.

6.2.45 Time shifted service descriptor

This descriptor is used in place of the service descriptor to indicate services which are time shifted copies of other services (see table 104).

Table 104: Time shifted service descriptor

| Syntax | Number of bits | Identifier |
|-------------------------------------|----------------|------------|
| time_shifted_service_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| reference_service_id | 16 | uimsbf |
| } | | |

Semantics for the time shifted service descriptor:

reference_service_id: This 16-bit field identifies the reference service of a NVOD collection of services. The reference service can always be found in this TS. The service_id here does not have a corresponding program_number in the program_map_section.

6.2.46 Transport stream descriptor

The transport stream descriptor (see table 105), being transmitted in the TSDDT (see ISO/IEC 13818-1 [15]) only, may be used to indicate the compliance of a transport stream with an MPEG based system, e.g. DVB.

Table 105: Transport stream descriptor

| Syntax | Number of bits | Identifier |
|---------------------------------|----------------|------------|
| transport_stream_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| byte | 8 | uimsbf |
| } | | |
| } | | |

Semantics for the transport stream descriptor:

byte: This is an 8-bit field. For identification of DVB Transport Streams the descriptor_length field shall be set to the value 0x03 indicating three following bytes. The three bytes shall contain the values 0x44, 0x56, 0x42 (ASCII: "DVB").

6.2.47 VBI data descriptor

The VBI data descriptor (see table 106) shall be used in the PSI PMT of a stream which carries VBI data as defined in EN 301 775 [6]. The appropriate ES_info_field of the program map section describing a VBI data stream shall contain one and only one VBI_data_descriptor.

For transmission of multiple VBI data streams in one service a VBI data descriptor in each of the component loops of the program map section is required.

Table 106: VBI_data_descriptor

| Syntax | Number of bits | Identifier |
|--------------------------------|----------------|------------|
| VBI_data_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| data_service_id | 8 | uimsbf |
| data_service_descriptor_length | 8 | uimsbf |
| if (data_service_id==0x01 | | |
| data_service_id==0x02 | | |
| data_service_id==0x04 | | |
| data_service_id==0x05 | | |
| data_service_id==0x06 | | |
| data_service_id==0x07) { | | |
| for (i=0; i<N; i++) { | | |
| reserved | 2 | bslbf |
| field_parity | 1 | bslbf |
| line_offset | 5 | uimsbf |
| } | | |
| } else { | | |
| for (i=0; i<N; i++) { | | |
| reserved | 8 | bslbf |
| } | | |
| } | | |
| } | | |

Semantics for VBI data descriptor:

data_service_id: This 8-bit field identifies a VBI service type contained in the elementary stream to which the VBI_data_descriptor is associated. Its value is defined in table 107. For every VBI service contained in the VBI data stream there shall be one and only one data_service_id coded in the VBI_data_descriptor. However, a VBI service may temporarily not be present in the VBI data stream while its data_service_id still is present in the VBI_data_descriptor. This discrepancy may last no longer than 10 s.

Table 107: data_service_id for VBI_data_descriptor

| data_service_id | Description |
|-----------------|--|
| 0x00 | reserved for future use |
| 0x01 | EBU teletext (Requires additional teletext_descriptor) |
| 0x02 | inverted teletext |
| 0x03 | reserved |
| 0x04 | VPS |
| 0x05 | WSS |
| 0x06 | Closed Captioning |
| 0x07 | monochrome 4:2:2 samples |
| 0x08 to 0xEF | reserved for future use |
| 0xF0 to 0xFF | user defined |

data_service_descriptor_length: This 8-bit field counts the number of bytes immediately following this field used for the description of the service indicated by data_service_id.

The descriptor itself contains one entry (byte) for each VBI line on which the coded data of the associated stream is intended to be presented.

field_parity: This 1-bit flag specifies the field for which the associated coded data is intended to be presented if it is transcoded into the VBI. The value "1" indicates the first (odd) field of a frame, the value "0" indicates the second (even) field of a frame. Within a data service descriptor, first all descriptor entries concerning the first field shall be given (if any), followed by all descriptor entries concerning the second field (if any).

line_offset: This 5-bit field specifies the line number on which the associated coded data is intended to be presented if it is transcoded into the VBI. Within a field, the line_offset numbering shall follow a progressive incremental order. The line_offset parameter follows the definition of the associated data field type (see clauses 2.4 to 2.8 in EN 301 775 [6]). A line_offset referred to as "line number undefined" shall not be used in the VBI_data_descriptor.

The encoder shall ensure that at least all coded lines in the associated elementary stream data will have their counterpart coded in the VBI_data_descriptor (and therefore in the appropriate data service descriptor). It is not allowed to code a specific line_offset and field_parity combination more than once in a single VBI_data_descriptor.

6.2.48 VBI teletext descriptor

The VBI teletext descriptor (see table 108) shall be used in the PSI PMT to identify streams which carry VBI data as well as EBU Teletext data. The descriptor shall be located within the relevant ES_info descriptor loop of the PMT.

Table 108: VBI teletext descriptor

| Syntax | Number of bits | Identifier |
|-----------------------------|----------------|------------|
| VBI_teletext_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| for (i=0;i<N;i++) { | | |
| ISO_639_language_code | 24 | bslbf |
| teletext_type | 5 | uimsbf |
| teletext_magazine_number | 3 | uimsbf |
| teletext_page_number | 8 | uimsbf |
| } | | |
| } | | |

Semantics for VBI teletext descriptor:

The semantics for the VBI teletext descriptor is the same as defined for the teletext descriptor in clause 6.2.42. The only exception is that the VBI teletext descriptor is not to be used to associate stream_type 0x06 with the VBI standard nor the EBU teletext standard. Decoders can only use the languages in this descriptor to select magazines and subtitles.

6.3 Extended descriptor identification and location

All extended descriptors are based on the extension_descriptor (see clause 6.2.16).

Table 109 lists the extended descriptors declared or defined within the present document, giving the descriptor tag extension values and the intended placement within the SI tables. This does not imply that their use in other tables is restricted.

Table 109: Possible locations of extended descriptors

| Descriptor | Tag extension value | NIT | BAT | SDT | EIT | TOT | PMT | SIT (note) |
|--|---------------------|-----|-----|-----|-----|-----|-----|------------|
| image_icon_descriptor | 0x00 | * | * | * | * | - | - | * |
| cpcm_delivery_signalling_descriptor (TS/TR 102 825 [39] and [i.3]) | 0x01 | - | - | * | * | - | - | - |
| CP_descriptor (TS/TR 102 825 [39] and [i.3]) | 0x02 | - | - | - | - | - | * | - |
| CP_identifier_descriptor (TS/TR 102 825 [39] and [i.3]) | 0x03 | * | * | * | * | - | - | - |
| T2_delivery_system_descriptor | 0x04 | * | - | - | - | - | - | - |
| SH_delivery_system_descriptor | 0x05 | * | - | - | - | - | - | - |
| supplementary_audio_descriptor | 0x06 | - | - | - | - | - | * | - |
| network_change_notify_descriptor | 0x07 | * | - | - | - | - | - | - |
| message_descriptor | 0x08 | * | * | * | * | - | - | - |
| target_region_descriptor | 0x09 | * | * | * | - | - | - | - |
| target_region_name_descriptor | 0x0A | * | * | - | - | - | - | - |
| service_relocated_descriptor | 0x0B | - | - | * | - | - | - | - |
| XAIT_PID_descriptor | 0x0C | * | * | - | - | - | - | - |
| C2_delivery_system_descriptor | 0x0D | * | - | - | - | - | - | - |
| DTS-HD_descriptor (Annex G) | 0x0E | - | - | - | - | - | * | - |
| DTS_Neural_descriptor (Annex L) | 0x0F | - | - | - | - | - | * | - |
| video_depth_range_descriptor | 0x10 | - | - | * | * | - | - | - |
| T2MI_descriptor | 0x11 | - | - | - | - | - | * | - |
| reserved for future use | 0x12 | - | - | - | - | - | - | - |
| URI_linkage_descriptor | 0x13 | * | * | * | * | - | * | * |
| CI_ancillary_data_descriptor ([i.9]) | 0x14 | * | * | * | * | - | - | - |
| AC-4_descriptor (Annex D) | 0x15 | - | - | - | - | - | * | - |
| C2_bundle_delivery_system_descriptor | 0x16 | * | - | - | - | - | - | - |
| S2X_satellite_delivery_system_descriptor | 0x17 | * | - | - | - | - | - | - |
| protection_message_descriptor [48] | 0x18 | - | - | - | - | - | * | - |
| audio_preselection_descriptor | 0x19 | - | - | - | - | - | * | - |
| TTML_subtitling_descriptor [56] | 0x20 | - | - | - | - | - | * | - |
| reserved for future use | 0x21 to 0x7F | | | | | | | |
| user defined | 0x80 to 0xFF | | | | | | | |

NOTE: Only found in Partial Transport Streams.

6.4 Extended descriptor coding

6.4.0 Introduction

Syntax and semantics of extended descriptors (in the style of clause 6.2) will be added here as extended descriptors are defined.

6.4.1 Audio preselection descriptor

The audio preselection descriptor (see table 110) provides information about the available audio preselections for one audio programme delivered by one NGA stream (single-stream delivery) or more than one NGA stream (multi-stream delivery). Information from this, and other descriptors, allows the receiver to select and present the appropriate NGA streams to the user.

In case of single-stream delivery, use of the audio preselection descriptor is optional.

The descriptor also provides information to enable linking of multiple NGA streams that contain audio programme components of a single audio programme. In this case of multi-stream delivery, only one audio preselection descriptor shall be present for that audio programme, and it shall be carried in the ES_info descriptor loop of the PSI PMT for the main NGA stream and shall contain all audio preselections that require auxiliary NGA streams. ES_info descriptor loops of PSI PMT for auxiliary NGA streams which are part of a multi-stream delivery shall include a stream identifier descriptor which conveys a component tag value matching one of those listed in the audio preselection descriptor for the main NGA stream.

The audio preselection descriptor is intended be used in the PMT for stream selection and audio preselection selection in the IRD.

The audio preselection descriptor is used in the PMT to indicate the main NGA stream of an audio programme. The descriptor also indirectly identifies all auxiliary NGA streams (if present) via the component_tag field.

The descriptor should list all audio preselections that the broadcaster intends for automatic selection by IRDs. The descriptor need not list all audio preselections available in an NGA stream. It shall list all audio preselections which contain audio programme components delivered via multiple NGA streams.

If NGA streams are transmitted together with non-NGA streams for the same service it is strongly recommended to include in the audio preselection descriptor all audio preselections intended for IRD automatic selection. Audio preselections which are not included in the audio preselection descriptor may not be considered by an IRD for automatic selection, and hence may not be offered to the listener. This is because IRDs are not expected to extract preselection information from the elementary stream when performing an automatic selection.

For further rules on using this descriptor in the PMT, see annex M.2.

Table 110: Audio preselection descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| audio_preselection_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| num_preselections | 5 | uimsbf |
| reserved_zero_future_use | 3 | bslbf |
| for (i=0;i<num_preselections;i++) { | | |
| preselection_id | 5 | uimsbf |
| audio_rendering_indication | 3 | uimsbf |
| audio_description | 1 | bslbf |
| spoken_subtitles | 1 | bslbf |
| dialogue_enhancement | 1 | bslbf |
| interactivity_enabled | 1 | bslbf |
| language_code_present | 1 | bslbf |
| text_label_present | 1 | bslbf |
| multi_stream_info_present | 1 | bslbf |
| future_extension | 1 | bslbf |
| if (language_code_present == 1) { | | |
| ISO_639_language_code | 24 | bslbf |
| } | | |
| if (text_label_present == 1) { | | |
| message_id | 8 | uimsbf |
| } | | |
| if (multi_stream_info_present == 1) { | | |
| num_aux_components | 3 | uimbsf |
| reserved_zero_future_use | 5 | bslbf |
| for (j=0;j<num_aux_components;j++) { | | |
| component_tag | 8 | uimbsf |
| } | | |
| } | | |
| if (future_extension == 1) { | | |
| reserved_zero_future_use | 3 | bslbf |
| future_extension_length | 5 | uimsbf |
| for (i=0;i< future_extension_length;i++) { | | |
| future_extension_byte | 8 | uimsbf |
| } | | |
| } | | |

Semantics for the audio preselection descriptor:

num_preselections: This 5-bit field indicates the number of audio preselections, that are described in this descriptor. The minimum number of num_preselections shall be "1".

preselection_id: This 5-bit field identifies this audio preselection. The value of this field shall match the corresponding field in the elementary stream as described in annex M.3. All the audio programme components corresponding to the first audio preselection in the loop are contained in the main NGA stream.

NOTE: It may be necessary to signal several audio preselections which differ only by the presence of additional features (e.g. accessibility, interactivity), facilitating automatic selection by receivers (e.g. based on user preferences). The audio preselection descriptor may contain a subset of the audio preselections available in the audio elementary stream.

audio_rendering_indication: This 3-bit field contains a hint for a preferred reproduction channel layout. It shall be coded according to table 111.

Table 111: Coding of audio rendering indication

| audio_rendering_indication | Description |
|-----------------------------------|---|
| 0 | no preference given for the reproduction channel layout |
| 1 | preferred reproduction channel layout is stereo |
| 2 | preferred reproduction channel layout is two-dimensional (e.g. 5.1 multi-channel) |
| 3 | preferred reproduction channel layout is three-dimensional |
| 4 | content is pre-rendered for consumption with headphones |
| 5 to 7 | reserved for future use |

audio_description: This 1-bit field indicates, when set to "1" that this audio preselection contains audio description for the visually impaired (i.e. a spoken description of the visual content of the service). When set to "0", it indicates that the audio preselection does not contain audio description for the visually impaired.

spoken_subtitles: This 1-bit field indicates, when set to "1" that this audio preselection contains spoken subtitles. When set to "0", it indicates that the audio preselection does not contain spoken subtitles.

dialogue_enhancement: This 1-bit field indicates, when set to "1" that this audio preselection provides support for dialogue enhancement. When set to "0", it indicates that the audio preselection does not provide support for dialogue enhancement.

NOTE: The dialogue enhancement feature is described in TS 101 154 [9]. The dialogue enhancement feature of an NGA preselection can be used in a flexible way to enable delivery of a wide variety of enhancements, including clean audio for the hearing impaired.

interactivity_enabled: This 1-bit field indicates, when set to "1" that the audio preselection provides additional user interactivity. When set to "0", it indicates that no additional user interactivity is available.

NOTE: This flag may be used to determine the need for initializing a user interface for audio interactivity.

language_code_present: This 1-bit field indicates, when set to "1" that an ISO_639_language_code field for this audio preselection is present. When set to "0", it indicates that no ISO_639_language_code field for this audio preselection is present.

text_label_present: This 1-bit field indicates, when set to "1" that a message_id field is present. The message_descriptor containing the same message_id provides a text label for this audio preselection. When set to "0", it indicates that there is no text label for this audio preselection.

multi_stream_info_present: This 1-bit field indicates, when set to "1" that the audio programme is conveyed over multiple elementary streams within the transport stream. When set to "0", it indicates that this PID conveys a complete audio programme. It shall be set to zero for the first audio preselection in the loop.

future_extension: This 1-bit flag indicates, when set to "1" that the reserved_zero_future_use and the future extension fields are present for this audio preselection. When set to "0", it indicates that the reserved_zero_future_use and the future extension fields are not present for this audio preselection.

ISO_639_language_code: This 24-bit field identifies the language of the audio preselection. If the audio preselection contains multiple languages, the language code shall be set to the primary language of the audio preselection. The ISO_639_language_code contains a 3-character code as specified by ISO639-2 [36]. Both ISO 639-2/B and ISO 639-2/T may be used. Each character shall be coded into 8 bits according to ISO/IEC 8859-1 [17] and inserted in order into the 24-bit field.

message_id: This 8-bit field where present provides the link to a textual description for this audio preselection. The text description is conveyed using a message descriptor with a matching message_id within the SDT of the service. Multiple message descriptors with a matching message_id may be present, if the text description is available in multiple languages.

NOTE: This text label is intended only for user interface display to aid the end user in choosing a specific preselection, and not for any automated actions by the receiver.

num_aux_components: This 3-bit field indicates the number of auxiliary elementary streams (in addition to the main elementary stream) over which the audio data of the audio service is distributed. The value of this field shall not be equal with "0".

component_tag: This 8-bit field has the same value as the component_tag field in the stream identifier descriptor that shall be present in the PSI program map section for the auxiliary NGA stream.

future_extension_length: This 5-bit field gives the total length in bytes of the following loop containing the future extension field.

future_extension_byte: This is an 8-bit field. The sequence of future_extension_byte fields specifies the future extension field. These bytes are reserved for future extensions of the audio_preselection_descriptor.

6.4.2 CI ancillary data descriptor

The CI ancillary data descriptor (see table 112) is used to convey ancillary data used in the construction of Content Identifiers (CI) in companion screen applications, as are defined in [i.9].

Table 112: CI ancillary data descriptor

| Syntax | Number of bits | Identifier |
|----------------------------------|----------------|------------|
| CI_ancillary_data_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| ancillary_data_byte | 8 | uimsbf |
| } | | |
| } | | |

Semantics for the CI ancillary data descriptor:

ancillary_data_byte: This is an 8 bit field. The sequence of ancillary_data_byte fields specifies the ancillary data field. The syntax and semantics of the ancillary data field are defined by the companion screen applications specification [i.9].

6.4.3 CP descriptor

The content protection descriptor (see table 113) is used to specify both system-wide and specific content protection management information. When the CP descriptor is found in the PMT, the CP_PID points to packets containing program related content protection information such as SRMs and content licence related information.

Table 113: CP descriptor

| Syntax | Number of bits | Identifier |
|--------------------------|----------------|------------|
| CP_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| CP_system_id | 16 | uimsbf |
| reserved | 3 | bslbf |
| CP_PID | 13 | uimsbf |
| for (i=0; i<N; i++) { | | |
| private_data_byte | 8 | uimsbf |
| } | | |
| } | | |

Semantics for the CP descriptor:

CP_system_id: This 16-bit field indicates the type of CP system applicable. It shall be coded according to TS 101 162 [57].

CP_PID: This 13-bit field indicates the PID of the Transport Stream packets which shall contain information for the CP systems as specified with the associated CP_system_id.

6.4.4 CP identifier descriptor

The CP identifier descriptor (see table 114) indicates whether a particular bouquet, service or event is associated with a content protection system or carries information relating to a content protection system (e.g. CP system metadata or CP system renewability messages). It identifies the CP system and the type of information by means of the CP_system_id.

Table 114: CP identifier descriptor

| Syntax | Number of bits | Identifier |
|------------------------------|----------------|------------|
| CP_identifier_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| CP_system_id | 16 | uimsbf |
| } | | |
| } | | |

Semantics for the CP identifier descriptor:

CP_system_id: This 16 bit field identifies the CP system and the type of information (e.g. Content Licence, Content metadata, System Renewability Messages, etc.). It shall be coded according to TS 101 162 [57].

6.4.5 CPCM delivery signalling descriptor

The CPCM delivery signalling descriptor conveys Usage State Information (USI) for Content Protection/Copy Management (CPCM) systems. Its syntax and semantics are fully defined and described in part 9 of TS 102 825 [39].

6.4.6 Delivery system descriptors

6.4.6.1 C2 delivery system descriptor

The C2_delivery_system_descriptor (see table 115) shall be used in the TS loop of the Network Information Table (NIT) to describe DVB-C2 transmissions according to [45]. This descriptor maps Transport Streams to data PLPs in data slices of C2 systems. The number of C2_delivery_system_descriptors per NIT shall equal the number of Transport Streams in the network carried over DVB-C2.

Each TS loop of the NIT describing a TS which is conveyed within a C2 system, shall contain either a C2 delivery system descriptor, or one or more C2 bundle delivery system descriptors (see clause 6.4.6.4).

Table 115: C2 delivery system descriptor

| Syntax | Number of bits | Identifier |
|-----------------------------------|----------------|------------|
| C2_delivery_system_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| plp_id | 8 | uimsbf |
| data_slice_id | 8 | uimsbf |
| C2_System_tuning_frequency | 32 | bslbf |
| C2_System_tuning_frequency_type | 2 | uimsbf |
| active_OFDM_symbol_duration | 3 | bslbf |
| guard_interval | 3 | bslbf |
| } | | |

Semantics for the C2 delivery system descriptor:

plp_id: This 8-bit field uniquely identifies a data PLP within the C2 system. The term is defined in [45].

data_slice_id: This 8-bit field uniquely identifies a data slice within the C2 system. The term is defined in [45].

C2_System_tuning_frequency: This 32-bit field indicates the frequency value in steps of 1 Hz that the receiver should tune to. The coding range is from minimum 0x00000001 (1 Hz) up to a maximum of 0xFFFFFFFF (4 294 967 295 Hz). This tuning frequency follows the syntax described in the C2_tuning_frequency_type field.

C2_System_tuning_frequency_type: This 2-bit field indicates the tuning frequency type (see table 116). This differentiation is needed to address all specifics of the C2 transmit signal and the underlying Data Slice structure.

Table 116: C2 tuning frequency type

| C2_tuning_frequency_type | Description |
|--------------------------|--|
| 00 | Data Slice tuning frequency This is the default option for C2 systems. The C2_System_tuning_frequency field conveys the tuning frequency of the data slice to which plp_id refers. The C2_System_tuning_frequency for a particular Data Slice is the sum of the L1 signalling parameters START_FREQUENCY and the DSLICE_TUNE_POS. Note that the Data Slice tuning frequency information in the L1 signalling as well as in the C2_delivery_system_descriptor have to be updated synchronously. |
| 01 | C2 system centre frequency This option is used by C2 headends that are not able to update the Data Slice tuning frequency information in the C2_delivery_system_descriptor and the L1 signalling in a synchronous way. The C2_System_tuning_frequency is the centre frequency of the C2 system, and it is required that a complete Preamble can be received. The receiver needs to evaluate the L1 signalling in the preamble to get knowledge of the final tuning position. |
| 10 | Initial tuning position for a (dependent) Static Data Slice Signalling of this option implies that the Data Slice to be demodulated is a (dependent) Static Slice. In the case of tuning to a (dependent) Static Data Slice, it cannot be guaranteed that the receiver is able to decode the L1 signalling at its final tuning position. Therefore the receiver will first tune to the signalled initial C2_System_tuning_frequency where a complete Preamble is transmitted. This frequency will usually be the C2_System centre frequency, but can be any tuning position where the receiver can reliably decode the L1 signal. The receiver needs to evaluate the L1 signalling in the preamble in order to determine additional parameters (particularly notch positions) as well as the final tuning frequency of the (dependent) Static Data Slice. |
| 11 | reserved for future use |

Active OFDM symbol duration: This 3 bit field indicates the duration of the active OFDM symbol according to table 117.

Table 117: Signalling format for the active OFDM symbol duration

| active_OFDM_symbol_duration | Description |
|-----------------------------|--|
| 000 | 448 μ s (4k FFT mode for 8MHz CATV systems) |
| 001 | 597,33 μ s (4k FFT mode for 6MHz CATV systems) |
| 010 to 111 | reserved for future use |

guard_interval: This 3-bit field indicates the guard interval according to table 118.

Table 118: Signalling format for the guard interval

| guard_interval | Guard interval values |
|----------------|-------------------------|
| 000 | 1/128 |
| 001 | 1/64 |
| 010 to 111 | reserved for future use |

6.4.6.2 SH delivery system descriptor

This descriptor (see table 119) is used to transmit the physical parameters for each DVB-SH [42] signal in the DVB network. This descriptor appears exactly once for each transport stream in each NIT sub_table describing the network. This descriptor is made of static and a dynamic part based on a modulation loop.

Table 119: SH delivery system descriptor

| Syntax | No. of bits | Identifier |
|----------------------------------|-------------|------------|
| SH_delivery_system_descriptor(){ | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| diversity_mode | 4 | bslbf |
| reserved | 4 | bslbf |
| for (i=0; i<N; i++){ | | |
| modulation_type | 1 | bslbf |
| interleaver_presence | 1 | bslbf |
| interleaver_type | 1 | bslbf |
| Reserved | 5 | bslbf |
| if (modulation_type == 0) { | | |
| Polarization | 2 | bslbf |
| roll_off | 2 | bslbf |
| modulation_mode | 2 | bslbf |
| code_rate | 4 | bslbf |
| symbol_rate | 5 | bslbf |
| reserved | 1 | bslbf |
| } else { | | |
| bandwidth | 3 | bslbf |
| priority | 1 | bslbf |
| constellation_and_hierarchy | 3 | bslbf |
| code_rate | 4 | bslbf |
| guard_interval | 2 | bslbf |
| transmission_mode | 2 | bslbf |
| common_frequency | 1 | bslbf |
| } | | |
| if (interleaver_presence == 1) { | | |
| if (interleaver_type == 0) { | | |
| common_multiplier | 6 | uimsbf |
| nof_late_taps | 6 | uimsbf |
| nof_slices | 6 | uimsbf |
| slice_distance | 8 | uimsbf |
| non_late_increments | 6 | uimsbf |
| } else { | | |
| common_multiplier | 6 | uimsbf |
| reserved | 2 | uimsbf |
| } | | |
| } | | |
| } | | |
| } | | |

Semantics for the SH delivery system descriptor:

diversity_mode: This 4-bit field describes the diversity modes and is coded according to table 120.

Table 120: diversity_mode tag

| diversity_mode | paTS | FEC diversity | FEC at phy | FEC at link |
|----------------|-------------------------|---------------|------------|-------------|
| 0000 | no | no | no | no |
| 0001 to 0111 | reserved for future use | | | |
| 1000 | yes | no | no | no |
| 1001 to 1100 | reserved for future use | | | |
| 1101 | yes | yes | no | yes |
| 1110 | yes | yes | yes | no |
| 1111 | yes | yes | yes | yes |

The modulation loop follows and has a number of iterations that depends on the selected SH configuration. For instance N = 1 in SFN, N = 2 in non-SFN.

modulation_type: This 1-bit field indicates which modulation is being used according to table 121.

Table 121: modulation_type

| modulation_type | Description |
|-----------------|-------------|
| 0 | TDM |
| 1 | OFDM |

interleaver_presence: This 1-bit flag indicates the use of an interleaver according to table 122.

Table 122: interleaver_presence flag

| interleaver_presence | Description |
|----------------------|-----------------------------|
| 0 | no interleaver info follows |
| 1 | an interleaver info follows |

interleaver_type: This 1-bit field indicates which interleaver is being used. Two options are possible:

- A complete_interleaver describes the interleaver in the same manner as the TPS or signalling field.
- A short interleaver assumes that the interleaver is a uniform one and therefore gives the common_multiplier only.

polarization: This 2-bit field indicates the polarization of the transmitted signal according to table 123.

Table 123: polarization field

| polarization | Description |
|--------------|---------------------|
| 00 | Linear - horizontal |
| 01 | Linear - vertical |
| 10 | Circular - left |
| 11 | Circular - right |

roll_off: This 2-bit field indicates the chosen roll-off factor according to table 124.

Table 124: roll_off

| roll_off | Description |
|----------|-------------------------|
| 00 | $\alpha = 0.35$ |
| 01 | $\alpha = 0.25$ |
| 10 | $\alpha = 0.15$ |
| 11 | reserved for future use |

modulation_mode: This 2-bit field indicates the chosen modulation according to table 125.

Table 125: modulation_mode

| modulation_mode | Description |
|-----------------|-------------------------|
| 00 | QPSK |
| 01 | 8PSK |
| 10 | 16APSK |
| 11 | reserved for future use |

code_rate: This 4-bit field indicates the chosen code_rate according to table 126.

Table 126: code_rate

| code_rate | Description |
|--------------|-------------------------|
| 0000 | 1/5 standard |
| 0001 | 2/9 standard |
| 0010 | 1/4 standard |
| 0011 | 2/7 standard |
| 0100 | 1/3 standard |
| 0101 | 1/3 complementary |
| 0110 | 2/5 standard |
| 0111 | 2/5 complementary |
| 1000 | 1/2 standard |
| 1001 | 1/2 complementary |
| 1010 | 2/3 standard |
| 1011 | 2/3 complementary |
| 1100 to 1111 | reserved for future use |

symbol_rate: This 5-bit field signals the TDM symbol rate expressed in Msymbols per second according to table 127. In case OFDM modulation is also used, the symbol_rate shall also be compliant with the corresponding symbol rate table for single carrier TDM in [43] where the OFDM parameters (bandwidth and guard interval) and the TDM roll-off determine a TDM symbol rate.

Table 127: TDM symbol_rate

| symbol_rate | Equivalent bandwidth | Equivalent guard interval | Symbol Rate | | |
|------------------|-------------------------|---------------------------|-----------------|-----------------|-----------------|
| | | | roll_off = 0.15 | roll_off = 0.25 | roll_off = 0.35 |
| 00000 | 8 | 1/4 | 34/5 | 32/5 | 29/5 |
| 00001 | 8 | 1/8 | 62/9 | 56/9 | 52/9 |
| 00010 | 8 | 1/16 | 116/17 | 108/17 | 100/17 |
| 00011 | 8 | 1/32 | 224/33 | 208/33 | 64/11 |
| 00100 | 7 | 1/4 | 119/20 | 28/5 | 203/40 |
| 00101 | 7 | 1/8 | 217/36 | 49/9 | 91/18 |
| 00110 | 7 | 1/16 | 203/34 | 189/34 | 175/34 |
| 00111 | 7 | 1/32 | 196/33 | 182/33 | 56/11 |
| 01000 | 6 | 1/4 | 51/10 | 24/5 | 87/20 |
| 01001 | 6 | 1/8 | 31/6 | 14/3 | 13/3 |
| 01010 | 6 | 1/16 | 87/17 | 81/17 | 75/17 |
| 01011 | 6 | 1/32 | 56/11 | 52/11 | 48/11 |
| 01100 | 5 | 1/4 | 17/4 | 4/1 | 29/8 |
| 01101 | 5 | 1/8 | 155/36 | 35/9 | 65/18 |
| 01110 | 5 | 1/16 | 145/34 | 135/34 | 125/34 |
| 01111 | 5 | 1/32 | 140/33 | 130/33 | 40/11 |
| 10000 | 1.7 | 1/4 | 34/25 | 32/25 | 29/25 |
| 10001 | 1.7 | 1/8 | 62/45 | 56/45 | 52/45 |
| 10010 | 1.7 | 1/16 | 116/85 | 108/85 | 20/17 |
| 10011 | 1.7 | 1/32 | 224/165 | 208/165 | 64/55 |
| 010011 to 111111 | reserved for future use | | | | |

bandwidth: This 3-bit field gives the OFDM bandwidth according to table 128.

Table 128: bandwidth tag

| bandwidth | Description |
|------------|-------------|
| 000 | 8 MHz |
| 001 | 7 MHz |
| 010 | 6 MHz |
| 011 | 5 MHz |
| 100 | 1,7 MHz |
| 101 to 111 | Reserved |

priority: This 1-bit field indicates the stream's hierarchical priority according to table 129. Its semantics depend on the chosen constellation_and_hierarchy value.

Table 129: priority

| Constellation_and_hierarchy | priority | Meaning |
|-----------------------------|----------|------------------|
| 000 | 0 | N/A |
| 001 | 1 | No priority mode |
| 010 | 0 | LP |
| 011 | 1 | HP |
| 100 | | |

constellation_and_hierarchy: This 3-bit field indicates the stream constellation and hierarchy according to table 130.

Table 130: constellation_and_hierarchy

| constellation_and_hierarchy | Description |
|-----------------------------|-------------------------------|
| 000 | QPSK |
| 001 | 16-QAM non hierarchical |
| 010 | 16-QAM hierarchical alpha = 1 |
| 011 | 16-QAM hierarchical alpha = 2 |
| 100 | 16-QAM hierarchical alpha = 3 |
| 101 to 111 | reserved for future use |

code_rate: This 4-bit field indicates the current TS code rate according to table 126.

guard_interval: This 2-bit field gives the chosen guard interval according to table 131.

Table 131: guard_interval

| guard_interval | Description |
|----------------|-------------|
| 00 | 1/32 |
| 01 | 1/16 |
| 10 | 1/8 |
| 11 | 1/4 |

transmission_mode: This 2-bit field indicates the transmission mode according to table 132.

Table 132: transmission_mode

| transmission_mode | Description |
|-------------------|-------------|
| 00 | 1k |
| 01 | 2k |
| 10 | 4k |
| 11 | 8k |

common_frequency: This 1-bit field indicates whether the modulation is used over a common frequency. A value of zero denotes that this is not a common frequency. A value of one indicates that this is a common frequency.

common_multiplier: This 6-bit field indicates the length increment in Interleaving Units between two consecutive taps of the physical interleaver belonging to the late tap part. The common_multiplier is also used to compute the actual tap length increment in the non-late category.

nof_late_taps: This 6-bit field indicates the number of taps of the physical time interleaver that belong to the late tap part. Values vary from "0" to "48": "0" signals "no taps belong to the late tap part", "48" signals "48 taps belong to late tap part".

nof_slices: This 6-bit field indicates the number of slices over which the physical time interleaver spans. Minimum value is 1, in which case all taps belong to the late tap part.

slice_distance: This 8-bit field indicates in units of SH frames the distance between two consecutive slices of the physical time interleaver. Values vary from "1" to "63"; if taps are all in the late tap part, this value shall be set to "1".

non_late_increment: This 6-bit field indicates the length increment between two consecutive taps belonging to the same non-late slice of the physical interleaver. The actual length increment in Interleaving Units is computed by multiplying this field with the value of the common_multiplier field.

6.4.6.3 T2 delivery system descriptor

The T2_delivery_system_descriptor (see table 133) shall be used in the TS loop of the Network Information Table to describe DVB-T2 transmissions according to EN 302 755 [40]. This descriptor maps Transport Streams to data Physical Layer Pipes (data PLPs) and T2 systems. For each Transport Stream carried in a DVB-T2 system in the network, there shall be at least one T2_delivery_system_descriptor in the corresponding transport stream loop of the respective NIT sub_table. The descriptor reflects a mapping of a TS - heading the NIT's TS loop - to a data PLP, whereby the T2_system_id field might occur multiple times, via multiple instances of the descriptor, because a particular combination of Transport Stream and data PLP may belong to different T2 systems. Within a network, several different Transport Streams may map to the same combination of T2_system_id and plp_id of the data PLP, for example when the same T2 System is used throughout a network and in different geographical areas different Transport Streams are used and mapped to the same data PLP.

If this descriptor is present in the NIT, the following definitions apply:

- The transport_stream_id and original_network_id announced within the transport stream loop of the NIT identify the Transport Stream that is carried by the PLP/T2 system pair identified with plp_id and T2_system_id in the descriptor.

NOTE 1: In the case of multiple PLPs in a T2 multiplex and in the presence of a common PLP, the corresponding TS is split into a data PLP, being referred to by the plp_id, and a common PLP. A corresponding re-assembly operation of the data PLP and common PLP on the receiver side is performed to recover the TS to be output by the receiver.

Table 133: T2 delivery system descriptor

| Syntax | Number of bits | Identifier |
|-----------------------------------|----------------|------------|
| T2_delivery_system_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| plp_id | 8 | uimsbf |
| T2_system_id | 16 | uimsbf |
| if (descriptor_length > 4){ | | |
| SISO/MISO | 2 | bslbf |
| bandwidth | 4 | bslbf |
| reserved_future_use | 2 | bslbf |
| guard_interval | 3 | bslbf |
| transmission_mode | 3 | bslbf |
| other_frequency_flag | 1 | bslbf |
| tfs_flag | 1 | bslbf |
| for (i=0;i<N,i++){ | | |
| cell_id | 16 | uimsbf |
| if (tfs_flag == 1){ | | |
| frequency_loop_length | 8 | uimsbf |
| for (j=0;j<N;j++){ | | |
| centre_frequency | 32 | uimsbf |
| } | | |
| } | | |
| else{ | | |
| centre_frequency | 32 | uimsbf |
| } | | |

| Syntax | Number of bits | Identifier |
|--------------------------|----------------|------------|
| subcell_info_loop_length | 8 | uimsbf |
| for (k=0;k<N;k++){ | | |
| cell_id_extension | 8 | uimsbf |
| transposer_frequency | 32 | uimsbf |
| } | | |
| } | | |
| } | | |
| } | | |

Semantics for the T2 delivery system descriptor:

plp_id: This 8-bit field identifies uniquely a PLP within the T2 system, identified with the T2_system_id. The term is defined in EN 302 755 [40].

T2_system_id: This 16-bit field identifies uniquely the T2 system within the DVB network (identified by network_id). The term is defined in EN 302 755 [40].

The remaining part of this descriptor, immediately following the T2_system_id field, may be present at most once per T2 system, because the parameters are uniquely applicable to all Transport Streams carried over a particular T2 system. The presence or absence of that part can be derived from the descriptor length field. In the absence of the remaining part this length equals 0x04, otherwise a larger value applies.

SISO/MISO: This 2-bit field indicates the SISO/MISO mode according to table 134.

Table 134: Signalling format for SISO/MISO mode

| SISO/MISO | Description |
|-----------|-------------------------|
| 00 | SISO |
| 01 | MISO |
| 10 | reserved for future use |
| 11 | reserved for future use |

bandwidth: This 4-bit field indicates the bandwidth in use according to table 135.

Table 135: Signalling format for the bandwidth

| Bandwidth | Bandwidth value |
|--------------|-------------------------|
| 0000 | 8 MHz |
| 0001 | 7 MHz |
| 0010 | 6 MHz |
| 0011 | 5 MHz |
| 0100 | 10 MHz |
| 0101 | 1,712 MHz |
| 0110 to 1111 | reserved for future use |

guard_interval: This 3-bit field indicates the guard interval according to table 136.

Table 136: Signalling format for each of the guard interval values

| guard_interval | Guard interval values |
|----------------|-------------------------|
| 000 | 1/32 |
| 001 | 1/16 |
| 010 | 1/8 |
| 011 | 1/4 |
| 100 | 1/128 |
| 101 | 19/128 |
| 110 | 19/256 |
| 111 | reserved for future use |

transmission_mode: This 3-bit field indicates the FFT size of the signals transmitted within the associated cell according to table 137.

Table 137: Signalling format for transmission mode/FFT size

| transmission_mode | Description |
|-------------------|-------------------------|
| 000 | 2k mode |
| 001 | 8k mode |
| 010 | 4k mode |
| 011 | 1k mode |
| 100 | 16k mode |
| 101 | 32k mode |
| 110 to 111 | reserved for future use |

other_frequency_flag: This 1-bit flag indicates whether other frequencies (non-TFS case) or other groups of frequencies (TFS case) are in use. The value 0 (zero) indicates that the set of frequencies (non-TFS case) or the set of groups of frequencies (TFS case) included in the descriptor is complete, whereas the value 1 (one) indicates that the set is incomplete.

tfs_flag: This 1-bit flag indicates whether a TFS arrangement is in place or not. It shall be coded according to table 138.

Table 138: Signalling format for the TFS arrangement

| tfs_flag | Description |
|----------|-----------------------------|
| 0 | No TFS arrangement in place |
| 1 | TFS arrangement in place |

cell_id: This 16-bit field uniquely identifies a cell, as defined in EN 302 755 [40].

NOTE 2: cell_id = 0x0000 indicates that no cell_id is assigned. In this case information for different cells can still be provided by means of multiple instances of the cell loop.

frequency_loop_length: This 8-bit field indicates the total length in bytes of the following loop that enumerates two to six centre frequencies belonging to the TFS arrangement associated with the named cell_id and its parameters. This loop is present only in the case of a TFS arrangement in place, otherwise only a single frequency per cell will be provided with a single instance of the centre_frequency parameter.

centre_frequency: This 32-bit field indicates the frequency value in multiples of 10 Hz. The coding range is from minimum 10 Hz (0x00000001) up to a maximum of 42 949 672 950 Hz (0xFFFFFFFF).

subcell_info_loop_length: This 8-bit field indicates the total length in bytes of the following loop that indicates the frequencies used in subcells.

cell_id_extension: This 8-bit field is used to identify a sub-cell within a cell.

transposer_frequency: This 32-bit field indicates the centre frequency that is used by a transposer in the sub-cell indicated. It is encoded in the same way as the centre_frequency field.

6.4.6.4 C2 bundle delivery system descriptor

The C2_bundle_delivery_system_descriptor (see table 139) shall be used in the TS loop of the Network Information Table (NIT) to describe DVB-C2 transmissions according to [45] using channel bundling. This descriptor maps Transport Streams bundled within multiple data PLPs, data slices, and C2 systems. The C2_bundle_delivery_system_descriptor shall list the given parameters of all bundled PLPs which are required to reassemble the TS.

NOTE: Bundled PLPs may be transmitted via several C2 systems.

Each TS loop of the NIT describing a TS which is conveyed within a C2 system, shall contain either a C2 delivery system descriptor (see clause 6.4.6.1), or one or more C2 bundle delivery system descriptors. When multiple C2 bundle delivery system descriptors are present in the same TS loop of the NIT describing a TS, then the bundle is defined by the combination of the PLPs listed in all those descriptors.

Table 139: C2 bundle delivery system descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| C2_bundle_delivery_system_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| for (i=0;i<N;i++) { | | |
| plp_id | 8 | uimsbf |
| data_slice_id | 8 | uimsbf |
| C2_system_tuning_frequency | 32 | bslbf |
| C2_system_tuning_frequency_type | 2 | uimsbf |
| active_OFDM_symbol_duration | 3 | bslbf |
| guard_interval | 3 | bslbf |
| master_channel | 1 | uimsbf |
| reserved_zero_future_use | 7 | bslbf |
| } | | |
| } | | |

Semantics of the C2 bundle delivery system descriptor:

plp_id: This 8-bit field shall be coded according to clause 6.4.6.1.

data_slice_id: This 8-bit field shall be coded according to clause 6.4.6.1.

C2_System_tuning_frequency: This 32-bit field shall be coded according to clause 6.4.6.1.

C2_System_tuning_frequency_type: This 2-bit field shall be coded according to clause 6.4.6.1.

active_OFDM_symbol_duration: This 3 bit field shall be coded according to clause 6.4.6.1.

guard_interval: This 3-bit field shall be coded according to clause 6.4.6.1.

master_channel: This 1-bit field indicates whether the associated PLP is using the master clock for the generation of the ISSY time stamps. If the associated PLP uses the master clock for ISSY generation, this field shall be set to '1'. Otherwise this field shall be set to '0'.

6.4.6.5 S2X satellite delivery system descriptor

When DVB-S2X modulation as defined in EN 302 307-2 [54] is used, the S2X delivery system descriptor is the only delivery system descriptor needed. It shall be coded according to table 140. It describes the modulation parameters used for S2X transmissions. When channel bonding is used, it also describes the modulation parameters of the channels that are bonded with the master channel. The PSI/SI tables as are required by the present document, may need to be carried in some or all of the bonded channels. To support network discovery by receivers, all bonded channels shall convey at least a NIT for the actual network. The distribution and replication of the further PSI/SI tables for a TS across all the bonded channels is subject to operational decisions by the network operator.

Table 140: S2X satellite S2Xdelivery system descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| S2X_satellite_delivery_system_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| receiver_profiles | 5 | bslbf |
| reserved_zero_future_use | 3 | bslbf |
| S2X_mode | 2 | uimsbf |
| scrambling_sequence_selector | 1 | bslbf |
| reserved_zero_future_use | 3 | bslbf |
| TS_GS_S2X_mode | 2 | bslbf |
| if (scrambling_sequence_selector == 1) { | | |
| reserved_zero_future_use | 6 | bslbf |
| scrambling_sequence_index | 18 | uimsbf |
| } | | |
| frequency (see note) | 32 | bslbf |

| | | |
|--|----|--------|
| orbital_position (see note) | 16 | bslbf |
| west_east_flag (see note) | 1 | bslbf |
| polarization (see note) | 2 | bslbf |
| multiple_input_stream_flag (see note) | 1 | bslbf |
| reserved_zero_future_use | 1 | bslbf |
| roll_off (see note) | 3 | bslbf |
| reserved_zero_future_use | 4 | bslbf |
| symbol_rate (see note) | 28 | bslbf |
| if (multiple_input_stream_flag == 1) { input_stream_identifier (see note) | 8 | uimsbf |
| } | | |
| if (S2X_mode==2) { timeslice_number | 8 | uimsbf |
| } | | |
| if (S2X_mode==3) { reserved_zero_future_use | 7 | bslbf |
| num_channel_bonds_minus_one | 1 | uimsbf |
| for (i=0;i<N;i++) { frequency | 32 | bslbf |
| orbital_position | 16 | bslbf |
| west_east_flag | 1 | bslbf |
| polarization | 2 | bslbf |
| bonded_channel_multiple_input_stream_flag | 1 | bslbf |
| reserved_zero_future_use | 1 | bslbf |
| roll_off | 3 | bslbf |
| reserved_zero_future_use | 4 | bslbf |
| symbol_rate | 28 | bslbf |
| if (bonded_channel_multiple_input_stream_flag == 1) { input_stream_identifier | 8 | uimsbf |
| } | | |
| } | | |
| for (i=0;i<N;i++) { reserved_future_use | 8 | bslbf |
| } | | |

NOTE: When channel bonding is used, these parameters describe the master channel.

Semantics for the S2X delivery system descriptor:

receiver_profiles: This 5-bit field indicates which receiver profiles are targeted by the stream. It shall be coded according to table 141. When no particular receiver profile is targeted, all bits shall be set to zero.

Table 141: Receiver profiles

| receiver_profiles (Bit number) | Description |
|--|-----------------------|
| b ₀ (see note) | Broadcast services |
| b ₁ | Interactive services |
| b ₂ | DSNG |
| b ₃ | Professional services |
| b ₄ | VL-SNR |
| NOTE: This bit is transmitted last (see clause 5.1.6). | |

S2X_mode: This 2-bit field indicates in which DVB-S2X mode the stream is operated. It shall be coded according to table 142.

Table 142: S2X mode

| S2X_mode | Description |
|----------|-------------------------|
| 0 | reserved for future use |
| 1 | S2X |
| 2 | S2X + time slicing |
| 3 | S2X + channel bonding |

scrambling_sequence_selector: This 1-bit field shall be coded according to clause 6.2.13.3.

TS_GS_S2X_mode: This 2-bit field indicates in which mode the stream is operated. It shall be coded according to table 143.

Table 143: TS/GS S2X mode

| TS_GS_S2X_mode | Description |
|----------------|--------------------------|
| 0 | generic packetized |
| 1 | GSE |
| 2 | GSE high efficiency mode |
| 3 | Transport Stream |

NOTE: These values are compatible with the coding of the TS/GS field in the BBFrame header of DVB-S2X (see clause 5.1.6 of EN 302 307-2 [54]).

scrambling_sequence_index: This 18-bit field shall be coded according to clause 6.2.13.3.

frequency: This 32-bit field shall be coded according to clause 6.2.13.2.

orbital_position: This 16-bit field shall be coded according to clause 6.2.13.2.

west_east_flag: This 1-bit field shall be coded according to clause 6.2.13.2.

polarization: This 2-bit field shall be coded according to clause 6.2.13.2.

multiple_input_stream_flag: This 1-bit field shall be coded according to clause 6.2.13.3.

roll_off: This 3-bit field indicates the roll-off factor used in DVB-S2X. It shall be coded according to table 144.

Table 144: S2X roll off

| roll_off | Description |
|----------|-------------------------|
| 000 | $\alpha = 0,35$ |
| 001 | $\alpha = 0,25$ |
| 010 | $\alpha = 0,20$ |
| 011 | reserved for future use |
| 100 | $\alpha = 0,15$ |
| 101 | $\alpha = 0,10$ |
| 110 | $\alpha = 0,05$ |
| 111 | reserved for future use |

symbol_rate: This 28-bit field shall be coded according to clause 6.2.13.2.

input_stream_identifier: This 1-bit field shall be coded according to clause 6.2.13.3.

timeslice_number: This 8-bit field shall be coded according to clause 6.2.13.3.

num_channel_bonds_minus_one: This 1-bit field indicates the number of descriptions of channels bonded with the master channel that follow. If one channel is bonded with the master channel, this field shall be set to zero, and one instance of the description loop shall follow. If two channels are bonded with the master channel, this field shall be set to one, and two instances of the description loop shall follow.

NOTE: The total number of bonded channels, including the master channel, is hence the value of this field plus two.

bonded_channel_multiple_input_stream_flag: This 1-bit field shall be coded as the multiple_input_stream_flag field defined in clause 6.2.13.3 to describe each non-master bonded channel.

6.4.7 Image icon descriptor

The image icon descriptor (see table 145) carries inline icon data or a URL that identifies the location of an icon file. It is intended to be used to carry or reference icons for two main purposes depending on where the icon is located:

- 1) When found in the Related Content Table (see TS 102 323 [12], clause 10) it shall be displayed as a call-to-action.
- 2) When found in the NIT, BAT, SDT, EIT or SIT it may be used within a receiver's native user interface and position information shall be omitted.

The resolution and size of the image may be specified within the image encoding itself and/or by the icon_type field. This will be determined depending on local profile.

Table 145: Image Icon descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| image_icon_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| descriptor_number | 4 | uimsbf |
| last_descriptor_number | 4 | uimsbf |
| reserved_future_use | 5 | uimsbf |
| icon_id | 3 | uimsbf |
| if (descriptor_number == 0x00) { | | |
| icon_transport_mode | 2 | uimsbf |
| position_flag | 1 | bslbf |
| if (position_flag == 0x01) { | | |
| coordinate_system | 3 | uimsbf |
| reserved_future_use | 2 | bslbf |
| icon_horizontal_origin | 12 | uimsbf |
| icon_vertical_origin | 12 | uimsbf |
| } | | |
| else { | | |
| reserved_future_use | 5 | bslbf |
| } | | |
| icon_type_length | 8 | uimsbf |
| for (i=0; i< icon_type_length; i++) { | | |
| icon_type_char | 8 | uimsbf |
| } | | |
| if (icon_transport_mode == 0x00) { | | |
| icon_data_length | 8 | uimsbf |
| for (j=0; j<icon_data_length; j++) { | | |
| icon_data_byte | 8 | uimsbf |
| } | | |
| } | | |
| else if (icon_transport_mode == 0x01) { | | |
| url_length | 8 | uimsbf |
| for (k=0; k< url_length; k++) { | | |
| url_char | 8 | uimsbf |
| } | | |
| } | | |
| } | | |
| else { | | |
| icon_data_length | 8 | uimsbf |
| for (m=0; m<icon_data_length; m++) { | | |
| icon_data_byte | 8 | uimsbf |
| } | | |
| } | | |
| } | | |

Semantics for the image icon descriptor:

descriptor_number: This 4-bit field gives the number of the descriptor. It is used to associate information which cannot be fitted into a single descriptor. The descriptor_number of the first image_icon_descriptor of an associated set of image_icon_descriptors shall be "0x00". The descriptor_number shall be incremented by 1 with each additional image_icon_descriptor with the same icon_id in this sub_table.

last_descriptor_number: This 4-bit field specifies the number of the last image_icon_descriptor (that is, the descriptor with the highest value of descriptor_number) of the associated set of descriptors with the same icon_id of which this descriptor is part.

icon_id: This is a 3-bit field which identifies this icon as distinct from other icons delivered in the same descriptor loop. The scope of the id is restricted to the current descriptor loop. The icon_id value of '000' shall not be used in this descriptor (see TS 102 323 [12]).

icon_transport_mode: This field indicates the mode of delivery of the icon as defined in table 146.

Table 146: Icon transport mode

| Value | Meaning |
|-------------|---|
| 0x00 | The icon is delivered in the icon_data_bytes |
| 0x01 | The location of the icon file is identified by URL carried in the url_char sequence of bytes. |
| 0x02 - 0x03 | DVB reserved |

position_flag: This one bit field indicates whether on-screen position information is provided for the icon image. If this field is set to '1' then icon_horizontal_origin and icon_vertical_origin shall be used to position the icon on-screen. If this field is set to '0' the position is not specified.

NOTE 1: If video scaling has occurred prior to the compositing of the video and graphics plane then the intended relationship between the icon position and underlying video may not be maintained.

coordinate_system: This field specifies the coordinate system on which the icon position is based. The value of this field is defined in table 147.

Table 147: Coordinate system

| Value | Meaning |
|--------------|------------------------------------|
| 0x00 | The coordinate system is 720x576 |
| 0x01 | The coordinate system is 1280x720 |
| 0x02 | The coordinate system is 1920x1080 |
| 0x03 to 0x06 | DVB reserved |
| 0x07 | User Private |

NOTE 2: If the coordinate system specified does not match the video resolution or display resolution then scaling of the icon position will be required. The mechanism for this is outside the scope of the present document.

icon_horizontal_origin: The horizontal pixel position on the screen to be used as icon origin (top-left). When this field is set to zero the left most column of pixels of the icon shall be positioned in the left most pixel column of the display.

icon_vertical_origin: The vertical pixel position on the screen to be used as icon origin (top-left). When this field is set to zero the upper most row of pixels of the icon shall be positioned in the upper most pixel row of the display.

icon_type_length: The length of the icon type string. If the type is not specified then this value shall be '0'.

icon_type_char: A sequence of bytes carrying a string describing the image type. This shall be a MIME image media subtype as described in RFC 2045 [44]. The top level media type "image" may be omitted. If the image types PNG or JPEG are used then they shall conform to the restrictions defined in ES 201 812 [38], clause 15.

NOTE 3: It is expected that receivers will implement at least the PNG and JPEG image formats conforming to the restrictions defined in ES 201 812 [38], clause 15. Broadcasters are thus encouraged to restrict themselves to these formats.

EXAMPLE: For a PNG image the icon_type_char field would contain either image/png or, for short, /png.

url_length: The length in bytes of the URL.

url_char: A sequence of bytes carrying a URL which describes the location of an icon file. This field shall be encoded according to clause 6.2 of TS 102 323 [12]. The URL may be a DVB locator referencing a file in an object carousel, as specified in clause 6.4 of TS 102 323 [12] or a reference to an IP-based resource.

icon_data_length: The length in bytes of the icon data carried in the remainder of the descriptor.

6.4.8 Message descriptor

This descriptor (see table 148) allows broadcasters to provide receivers with a textual message which the receiver may display to the user at appropriate times (not defined here).

Table 148: Message descriptor

| Syntax | Number of bits | Identifier |
|--------------------------|----------------|------------|
| message_descriptor () { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| message_id | 8 | uimsbf |
| ISO_639_language_code | 24 | bslbf |
| for (j=0; j<N; j++) { | | |
| text_char | 8 | uimsbf |
| } | | |
| } | | |

Semantics for the message descriptor:

message_id: This 8-bit field uniquely identifies the message within the scope of the sub_table in which it appears. Note that multiple descriptors with the same message_id shall not be broadcast unless they have differing ISO_639_language_code values. The value of this field shall be in the range of 1 to 255.

ISO_639_language_code: This 24-bit field contains the ISO 639-2 three character language code of the language of the textual message. Both ISO 639-2/B and ISO 639-2/T may be used. Each character is coded into 8-bits according to ISO/IEC 8859-1 [17] and inserted in order into the 24-bit field.

text_char: This is an 8-bit field containing a string of characters specifying the text to be displayed. Textual information is coded according to annex A.

6.4.9 Network change notify descriptor

This descriptor (see table 149) allows broadcasters to signal network change events to receivers. A network change event is a single, clearly identifiable change in the network configuration, e.g. transmission parameters and/or available services, which may require action on the part of receivers.

Network changes may be confined to a part of the network, as indicated by the use of the cell_id field (used in DVB-T and DVB-T2 systems) or may apply to the whole network, as defined by the NIT.

Where used, the network_change_notify descriptor shall signal at all times the current list of scheduled network change events. The absence of a network_change_notify descriptor shall be used to indicate that there are no scheduled network change events.

Table 149: Network_change_notify descriptor

| Syntax | Number of bits | Identifier |
|---------------------------------------|----------------|------------|
| network_change_notify_descriptor () { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| cell_id | 16 | uimsbf |
| loop_length | 8 | uimsbf |
| for (j=0; j<N; j++) { | | |
| network_change_id | 8 | uimsbf |
| network_change_version | 8 | uimsbf |
| start_time_of_change | 40 | bslbf |
| change_duration | 24 | uimsbf |
| receiver_category | 3 | uimsbf |
| invariant_ts_present | 1 | uimsbf |
| change_type | 4 | uimsbf |
| message_id | 8 | uimsbf |
| if (invariant_ts_present == 1) { | | |

| | | |
|-------------------|----|--------|
| invariant_ts_tsid | 16 | uimsbf |
| invariant_ts_onid | 16 | uimsbf |
| } | | |
| } | | |
| } | | |

Semantics for the network change notify descriptor:

cell_id: This 16-bit field uniquely identifies a cell within a DVB-T or DVB-T2 network (as defined by network_id). A cell_id of 0x0000 shall be used to signal a change affecting all cell_ids. Where multiple changes exist for a given cell_id, they shall be signalled in one instance of the descriptor. For other delivery systems, this field is not used and the network change applies to the network defined by the NIT carrying this descriptor.

loop_length: This 8-bit field specifies the length in bytes of the following items.

network_change_id: This 8-bit field is a unique identifier for the network change event signalled within this cell. No inferences should be made from the incremental or other nature of the value of network_change_id.

network_change_version: This 8-bit field signals the version of the change. It shall be changed if any parameter within this loop is changed. Values shall be assigned incrementally until 0xFF. If further network_change_versions are required, a new network_change_id shall be issued.

start_time_of_change: This 40-bit field indicates the time at which the network changes are planned to start in Universal Time, Co-ordinated (UTC) and Modified Julian Date (MJD) (see annex C). This field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 digits in 4-bit BCD.

change_duration: This 24-bit field indicates the planned duration of the network change in hours, minutes and seconds coded as 6 digits in 4-bit BCD. Durations signalled within this field shall be limited to a maximum of 11:59:59.

EXAMPLE: 01:45:30 is coded as "0x014530".

receiver_category: This 3-bit field indicates the category of receivers affected by the change being signalled according to table 150.

Table 150: Receiver category

| receiver_category | Description |
|-------------------|---|
| 0x0 | All receivers |
| 0x1 | DVB-T2 or DVB-S2 or DVB-C2 capable receivers only |
| 0x2 to 0x7 | reserved for future use |

invariant_ts_present: If this 1-bit field is set to '1', an invariant transport stream is being signalled. If set to '0', all multiplexes with this cell_id (for DVB-T or DVB-T2 systems) or within the network (for other delivery systems) should be considered as subject to change. An invariant transport stream is one which uses the same transmission parameters before and after the network changes (but not necessarily during the change).

change_type: This 4-bit field specifies the type of change that will take place, as defined in table 151. Minor changes are defined as those changes which can be detected by a receiver by comparison of the old and new SI. Major changes are defined as those which could require a receiver to tune or scan away from the current multiplex.

The "default" category shall be used when another category does not adequately describe the current scenario, or when multiple categories would describe the current scenario. The "message only" category shall be used when there are no changes to the network but the broadcaster wishes to provide a message to be displayed by the receivers. The "coverage change" category shall be used when power and/or modulation parameter changes may change the coverage of a transmitter. It shall also be used when a cell or transmitter is being added or removed since this can also change the coverage. A "coverage change" may result in the receiver no longer receiving signals from the most appropriate cell or transmitter.

Table 151: Network change type

| change_type | Description |
|-------------|---|
| 0x0 | Message only |
| 0x1 | Minor - default |
| 0x2 | Minor - multiplex removed |
| 0x3 | Minor - service changed |
| 0x4 to 0x7 | reserved for future use for other minor changes |
| 0x8 | Major - default |
| 0x9 | Major - multiplex frequency changed |
| 0xA | Major - multiplex coverage changed |
| 0xB | Major - multiplex added |
| 0xC to 0xF | reserved for future use for other major changes |

message_id: This 8-bit field is used to link to a message in the message descriptor carried in the same NIT. A message_id of 0x00 shall be used to signal that there is no message associated with this network change. Multiple network changes may be linked to a single message_id.

invariant_ts_tsid: This 16-bit field contains the transport_stream_id of the invariant transport stream.

invariant_ts_onid: This 16-bit field contains the original_network_id of the invariant transport stream.

6.4.10 Service relocated descriptor

If a service has moved from one TS to another, a mechanism is provided which enables an IRD to track the service at its new location (for example between TSs) by means of a service_relocated_descriptor (see table 152).

NOTE: A new location for a service first of all implies new identifiers are being used. It does not necessarily imply that it has moved to a different physical location.

Table 152: Service relocated descriptor

| Syntax | Number of bits | Identifier |
|----------------------------------|----------------|------------|
| service_relocated_descriptor() { | | |
| descriptor_tag | 8 | uimbsf |
| descriptor_length | 8 | uimbsf |
| descriptor_tag_extension | 8 | uimbsf |
| old_original_network_id | 16 | uimbsf |
| old_transport_stream_id | 16 | uimbsf |
| old_service_id | 16 | uimbsf |
| } | | |

Semantics for the service relocated descriptor:

old_original_network_id: This field contains the original_network_id of the TS in which the service was found before the relocation.

old_transport_stream_id: This field contains the transport_stream_id of the TS in which the service was found before the relocation.

old_service_id: This field contains the service_id of the service before the relocation.

6.4.11 Supplementary audio descriptor

The supplementary audio descriptor (see table 153) provides additional information about the associated audio component. Information from this and other descriptors allows the receiver to present the appropriate stream, or mix of streams, to the user. It shall be carried in the ES loop of the PSI PMT for each audio stream carrying supplementary services. It may also be carried in this position for audio streams carrying main audio. The use of the descriptor is described in more detail in annex J.

The mix_type field in this descriptor can be used to indicate if the audio stream is a “complete and independent” stream, or a dependent stream which is not complete and independent and is intended to be mixed with a separate audio stream by the receiver, this is known as “receiver-mix audio”, see clause J.2. The mix_type is also used to indicate when an

SAOC-DE parametric data stream is embedded into an existing audio stream or carried as a separate component, see clause J.5.

The `editorial_classification` field can be used to indicate the editorial use of the audio stream. This can be used to indicate which streams are Main programme audio that are targeted at the majority of viewers. The `editorial_classification` field can also be used to indicate the use of additional audio streams such as those for audiences with visual and/or hearing impairments. Examples include spoken subtitles, audio description or speech audio without music and sounds effects.

For receivers that don't have the capability to mix a dependent stream with another audio stream, a broadcaster can pre-mix the audio and broadcast it as a "complete and independent" additional audio stream, this is known as "broadcast-mix audio", see clause J.3.

The supplementary audio descriptor can override the ISO 639 language code defined in the other descriptors in the same ES loop of the PMT, such as the `ISO_639_language_descriptor` [15]. This is commonly used where the language code in the other descriptors (such as the `ISO_639_language_descriptor`) is set to aid legacy receivers. In such cases the more accurate language code is conveyed in the supplementary audio descriptor.

This descriptor shall be carried in the ES loop of the PSI PMT for each audio stream carrying a dependent stream. It may also be carried in this position for any audio stream. If a service contains a dependent audio stream this descriptor should be present in the ES loop of the PSI PMT carrying the audio streams with which the dependent stream could be mixed. The use of the descriptor is described in more detail in annex J.

NOTE: This descriptor cannot currently be used to describe audio which contains multiple audio sub-streams on a single PID.

For further rules on using this descriptor in the PMT, see annex M.2.

Table 153: Supplementary audio descriptor

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| <code>supplementary_audio_descriptor() {</code> | | |
| <code>descriptor_tag</code> | 8 | uimbsf |
| <code>descriptor_length</code> | 8 | uimbsf |
| <code>descriptor_tag_extension</code> | 8 | uimbsf |
| <code>mix_type</code> | 1 | uimbsf |
| <code>editorial_classification</code> | 5 | uimbsf |
| <code>reserved_future_use</code> | 1 | uimbsf |
| <code>language_code_present</code> | 1 | uimbsf |
| if (<code>language_code_present == 1</code>) { | | |
| <code>ISO_639_language_code</code> | 24 | bslbf |
| } | | |
| for (<code>i=0;i<N;i++</code>) { | | |
| <code>private_data_byte</code> | 8 | uimbsf |
| } | | |
| <code>}</code> | | |

Semantics for the supplementary audio descriptor:

mix_type: This 1-bit field indicates whether the audio stream is a complete and independent stream or whether it is a dependent stream intended to be mixed with a separate complete and independent audio stream according to TS 101 154 [9] annex G. It shall be coded according to table 154.

Table 154: mix_type coding

| mix_type | Description |
|--|---|
| 0 | Audio stream is a dependent stream and is intended to be mixed or combined with a separate complete and independent audio stream by the receiver. |
| 1 | Audio stream is a complete and independent stream. |
| NOTE: Restrictions on valid combinations of audio_type, mix_type, and editorial_classification are given in annex J.4. | |

editorial_classification: This 5-bit field indicates the editorial intention of the audio stream according to table 155.

Table 155: editorial_classification coding

| editorial_classification | Description |
|---|--|
| 0x00 | Main audio (contains all of the main audio components and can be presented on its own or mixed with a supplementary audio stream). This classification shall not be used for broadcast-mix audio (see clause J.3) e.g. audio streams that are premixed with visual impaired or hearing impaired audio. |
| 0x01 | Audio description for the visually impaired (contains a spoken description of the visual content of the service). |
| 0x02 | Clean audio for the hearing impaired. |
| 0x03 | Spoken subtitles for the visually impaired. |
| 0x04 | Dependent parametric data stream (not embedded) |
| 0x05 to 0x16 | Reserved for future use. |
| 0x17 | Unspecific supplementary audio for the general audience |
| 0x18 to 0x1F | User defined. |
| NOTE: Restrictions on valid combinations of audio_type, mix_type, and editorial_classification are given in clause J.4. | |

language_code_present: When this field is set to 1 (one), it indicates that the ISO_639_language_code field is present and this ISO 639 language code shall override other ISO 639 language code contained in other descriptors in the same ES loop of the PMT. When set to 0 (zero), no language code field follows, and the language of the audio stream shall be indicated by another descriptor in the same ES loop of the PMT.

ISO_639_language_code: If present, this 24 bit field identifies the language, as defined by ISO 639, of the audio stream and overrides the value of any other ISO_639_language_code contained in other descriptors in the same ES loop.

6.4.12 Target region descriptor

The target_region_descriptor (see table 156) identifies a set of target regions. The set of target regions identified by the descriptor is the union of target region codes, identified by each loop in the descriptor, additionally qualified by the country_code. If the loop is empty, then this descriptor identifies a single country, by means of the country_code.

The target_region_descriptor is a scoping descriptor (see clause 6.5). The location of a particular instance of the target_region_descriptor defines the scope of the descriptor.

The scoping property is the set of target regions it identifies. The set of target regions identified at a particular scope is the union of the sets of target regions identified by all target region descriptors present at that scope in the same table.

The target_region_descriptor may appear:

- c) in the first loop of the NIT for all services within a network,
- d) in the TS loop of the NIT for all services within a transport stream of a network,
- e) in the first loop of the BAT for all services within a bouquet,
- f) in the TS loop of the BAT for all services within a transport stream of a bouquet,
- g) in the SDT for a single service within a transport stream.

Its use in other tables is not defined. This descriptor may appear multiple times in any descriptor loop of a table. The set of target regions defined in a descriptor loop of a table, is the union of the target regions defined by each target region descriptor.

The hierarchical level of the target region identified by a single loop of the target_region_descriptor depends on the value of the region_depth field.

- If the region_depth is 0, then the target_region_descriptor loop identifies a single country.
- If the value of the region_depth is 1, then the target_region_descriptor loop identifies a single primary region within a country.
- If the value of the region_depth is 2, then the target_region_descriptor loop identifies a single secondary region, within a primary region within a country.
- If the value of the region_depth is 3, the target_region_descriptor loop identifies a single tertiary region, within a secondary region within a primary region within a country.

Table 156: target_region_descriptor

| Syntax | Number of bits | Identifier |
|------------------------------|----------------|------------|
| target_region_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| country_code | 24 | bslbf |
| for (i=0;i<N;i++){ | | |
| reserved | 5 | bslbf |
| country_code_flag | 1 | bslbf |
| region_depth | 2 | bslbf |
| if (country_code_flag==1) { | | |
| country_code | 24 | bslbf |
| } | | |
| if (region_depth>=1) { | | |
| primary_region_code | 8 | bslbf |
| if (region_depth>=2) { | | |
| secondary_region_code | 8 | bslbf |
| if (region_depth==3) { | | |
| tertiary_region_code | 16 | bslbf |
| } | | |
| } | | |
| } | | |
| } | | |
| } | | |

Semantics for the target region descriptor:

country_code: This 24-bit field identifies a country or group of countries. It shall be coded as three characters, where each character is coded into 8-bits according to ISO/IEC 8859-1 [17] and inserted in order into the field. Individual countries shall be identified by using the alpha-3 code as specified in ISO 3166 [35]. A group of countries shall be identified by the string representation of a decimal number in the range 900 to 999. These numbers shall be coded according to TS 101 162 [57].

EXAMPLE: United Kingdom has 3-character code "GBR", which is coded as:
 "0100 0111 0100 0010 0101 0010".

The country code can be signalled in two possible locations: the location outside the loop specifies the initial country code which applies until overridden by a country_code defined inside the loop. A country_code defined inside the loop applies to all following loops in the descriptor until overridden by another country_code defined inside a subsequent loop.

country_code_flag: This 1-bit field if set to one indicates a country_code is defined inside the loop. The country code applies to all following target regions in the descriptor until another country code is defined inside a subsequent loop.

region_depth: This 2-bit field identifies the hierarchical depth of the target region described by this descriptor loop. The semantics of the region_depth field are:

- If the region_depth is 0 then the target region is the entire country identified by the current value of country_code.
- If the region_depth is 1 then the target region is a primary region, identified by the primary_region_code within the identified country.
- If the region_depth is 2 then the target region is a secondary region, identified by the combination of the secondary_region_code and primary_region_code within the identified country.
- If the region_depth is 3 then the target region is a tertiary region, identified by the combination of the tertiary_region_code, secondary_region_code and primary_region_code within the identified country.

primary_region_code: This 8-bit field identifies the target primary region within a country.

secondary_region_code: This 8-bit field identifies the target secondary region within a primary region within a country.

tertiary_region_code: This 16-bit field identifies the target tertiary region within a secondary region within a primary region within a country.

6.4.13 Target region name descriptor

A target region is a geographical area containing the intended audience of a broadcast. A target region can be either an entire country, or a geographical area within a country. A country may be subdivided into a set of primary regions and each primary region may be subdivided into a set of secondary regions and, finally, each secondary region may be subdivided into a set of tertiary regions. In general, target regions subdivide a country into a hierarchy of primary, secondary and tertiary regions.

The target region name descriptor (see table 157) assigns a name to a target region. The target region name may be represented in multiple languages. For a particular target region within a country, only one name per language code shall be signalled.

This descriptor may only be present in the first loop of the BAT or in the first loop of the NIT. Multiple instances of this descriptor are allowed. If the name of a target region is defined in both the BAT and the NIT, then the name of that region in the BAT takes precedence.

All target regions, identified by a target region descriptor in a NIT, BAT or SDT, shall be named by a target region name descriptor, defined in a NIT or a BAT. The hierarchical structure created may have variable depth. Broadcasters shall ensure an appropriate name is transmitted at each populated level and at all levels above them in the hierarchy.

Table 157: target_region_name_descriptor

| Syntax | Number of bits | Identifier |
|----------------------------------|----------------|------------|
| target_region_name_descriptor(){ | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| country_code | 24 | bslbf |
| ISO_639_language_code | 24 | bslbf |
| for (i=0;i<N;i++){ | | |
| region_depth | 2 | bslbf |
| region_name_length | 6 | bslbf |
| for (j=0;j<M;j++){ | | |
| text_char | 8 | uimsbf |
| } | | |
| primary_region_code | 8 | bslbf |
| if (region_depth >=2){ | | |
| secondary_region_code | 8 | bslbf |
| if (region_depth==3){ | | |
| tertiary_region_code | 16 | bslbf |
| } | | |
| } | | |
| } | | |

| | | |
|---|--|--|
| } | | |
|---|--|--|

Semantics for the target region name descriptor:

country_code: This 24-bit field identifies a country or group of countries. It shall be coded as three characters, where each character is coded into 8-bits according to ISO/IEC 8859-1 [17] and inserted in order into the field. Individual countries shall be identified by using the alpha-3 code as specified in ISO 3166 [35]. A group of countries shall be identified by the string representation of a decimal number in the range 900 to 999. These numbers shall be coded according to TS 101 162 [57].

EXAMPLE 1: United Kingdom has 3-character code "GBR", which is coded as:
"0100 0111 0100 0010 0101 0010".

ISO_639_language_code: This 24-bit field identifies the language of the component text description which may be contained in this descriptor. The ISO_639_language_code contains a 3-character code as specified by ISO 639-2 [36]. Both ISO 639-2/B and ISO 639-2/T may be used. Each character is coded into 8 bits according to ISO/IEC 8859-1 [17] and inserted in order into the 24-bit field.

EXAMPLE 2: French has 3-character code "fre", which is coded as:
"0110 0110 0111 0010 0110 0101"

region_depth: This 2-bit field identifies the hierarchical depth of the target region described by this descriptor loop. The value 0 is reserved. The semantics of the region_depth field are:

- If the region_depth is 1 then the target region is a primary region, identified by the primary_region_code within the identified country.
- If the region_depth is 2 then the target region is a secondary region, identified by the combination of the secondary_region_code and primary_region_code within the identified country.
- If the region_depth is 3 then the target region is a tertiary region, identified by the combination of the tertiary_region_code, secondary_region_code and primary_region_code within the identified country.

region_name_length: This 6-bit field specifies the length in bytes of the string that provides the target region name.

text_char: This is an 8-bit field. A string of text_char fields specifies the target region name. Text information shall be coded using the character sets and methods described in annex A.

primary_region_code: This 8-bit field identifies the target primary region within a country.

secondary_region_code: This 8-bit field identifies the target secondary region within a primary region within a country.

tertiary_region_code: This 16-bit field identifies the target tertiary region within a secondary region within a primary region within a country.

6.4.14 T2-MI descriptor

The T2-MI descriptor (see table 158) may be used in the PSI PMT to identify each PID carrying a single stream of T2-MI [49] within a DVB transport stream.

Table 158: T2MI_descriptor

| Syntax | Number of bits | Identifier |
|----------------------------|----------------|------------|
| T2MI_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| reserved_zero_future_use | 5 | bslbf |
| t2mi_stream_id | 3 | uimsbf |
| reserved_zero_future_use | 5 | bslbf |
| num_t2mi_streams_minus_one | 3 | uimsbf |
| reserved_zero_future_use | 7 | bslbf |
| pcr_iscr_common_clock_flag | 1 | bslbf |
| for (i=0; i<N; i++) { | | |

| | | |
|--------------------------|---|-------|
| reserved_zero_future_use | 8 | bslbf |
| } | | |
| } | | |

Semantics for the T2-MI descriptor:

t2mi_stream_id: Identifies the different T2-MI streams in a unique way. This field shall have the same value as the t2mi_stream_id field in the header of the T2-MI packets [49].

num_t2mi_streams_minus_one: The total number of T2-MI streams required to generate the complete DVB-T2 signal. This shall have the same value for all T2-MI streams that make up a composite signal.

EXAMPLE 1: Where only one T2-MI stream is required by a modulator to generate a complete signal, the field has a value of '0'.

EXAMPLE 2: Where a T2 composite signal is being generated that requires two T2-MI streams to make the complete signal, the field has a value of '1'.

pcr_iscr_common_clock_flag: A value of '1' indicates that the PCRs signalled in the PMT of the service carrying the T2-MI and the ISCRs (Input Stream Time References) in the T2 Baseband Frame packets are generated from a common clock source (see annex G of [49]).

reserved_future_use: These bits are reserved for future use, and they shall be set to value of '0'.

6.4.15 URI linkage descriptor

The URI linkage descriptor (see table 159) identifies a resource obtainable via an IP network. The usage of the resource identified in this way will depend upon both the location of this descriptor and the linkage type signalled in the descriptor.

Table 159: URI_linkage_descriptor

| Syntax | Number of bits | Identifier |
|---------------------------------|----------------|------------|
| URI_linkage_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| uri_linkage_type | 8 | uimsbf |
| uri_length | 8 | uimsbf |
| for (i=0;i<uri_length;i++) { | | |
| uri_char | 8 | bslbf |
| } | | |
| if ((uri_linkage_type == 0x00) | | |
| (uri_linkage_type == 0x01)) { | | |
| min_polling_interval | 16 | uimsbf |
| } | | |
| for (i=0;i<N;i++) { | | |
| private_data_byte | 8 | bslbf |
| } | | |
| } | | |

Semantics for the URI linkage descriptor:

uri_linkage_type: This is an 8-bit field specifying the type of URI linkage e.g. to information. It shall be coded according to ETSI TS 101 162 [57].

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

uri_char: This is an 8-bit field. A string of uri_char fields specifies a URI string as defined in RFC 3986 [51].

min_polling_interval: This is a 16-bit field. The value of this field shall be the minimum time in intervals of two seconds between requests made by the receiver to this URI. A value of zero means that no minimum interval is implied. In this case receivers should choose a sensible default (for example a random value) or use a value defined by means outside of the scope of this document (for example a value specified by local profile or platform).

EXAMPLE: An interval of 10 minutes would be encoded as 300 (0x012C) since 300 intervals of 2 seconds each sum up to 600 seconds, which is equal to 10 minutes.

EXAMPLE: An interval of 24 hours would be encoded as 43 200 (0xA8C0) since 43 200 intervals of two seconds each sum up to 86 400 seconds, which equals 24 hours.

EXAMPLE: When the maximum value of 65 535 (0xFFFF) is encoded, the minimum interval is 131 070 seconds, which equals 36 hours, 24 minutes, and 30 seconds.

private_data_byte: This is an 8-bit field, the value of which is privately defined.

6.4.16 Video depth range descriptor

6.4.16.0 Semantics and syntax of the video depth range descriptor

To assist receivers in optimizing the placement of graphics, like text or icons, they wish to display on top of plano-stereoscopic 3D video, the video_depth_range_descriptor (see table 160) indicates the intended depth range of the 3D video.

The video_depth_range_descriptor is a scoping descriptor (see clause 6.5). The location of a particular instance of the video_depth_range_descriptor defines the scope of the descriptor. If this descriptor occurs multiple times in a descriptor loop, then a particular range_type shall be present in only one instance of the video_depth_range_descriptor.

Table 160: video_depth_range_descriptor

| Syntax | Number of bits | Identifier |
|----------------------------------|----------------|------------|
| video_depth_range_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| range_type | 8 | uimsbf |
| range_length | 8 | uimsbf |
| if (range_type == 0) { | | |
| production_disparity_hint_info() | | |
| } else if (range_type == 1) { | | |
| /* empty */ | | |
| } else { | | |
| for (i=0; i<range_length; i++) { | | |
| range_selector_byte | 8 | bslbf |
| } | | |
| } | | |
| } | | |
| } | | |

Semantics for the video depth range descriptor:

range_type: This 8-bit field indicates the type of depth range described in this descriptor. It shall be coded according to table 161. A particular range_type shall occur in only one loop of this descriptor.

Table 161: range_type coding

| range_type | Description |
|--------------|------------------------------------|
| 0x00 | production disparity hint |
| 0x01 | multi-region disparity SEI present |
| 0x02 to 0xFF | reserved for future use |

If the range_type is set to 0x01, it indicates the presence of multi-region disparity SEI data in the video elementary stream as defined in annex B.11 of TS 101 154 [9]. In this case, range_length shall be set to zero.

range_length: This 8-bit field indicates the number of bytes following this field up to, but not including either the next range_type field or the end of the descriptor.

production_disparity_hint_info(): This field shall be coded according to clause 6.4.16.1.

range_selector_byte: For the purpose of application selection, the range_selector_bytes are used. The definition of the range_selector_bytes of the video_depth_range_descriptor will depend on the range_type.

6.4.16.1 Production disparity hint

When used, the disparity_hint_info() conveys the recommended maximum and minimum disparity values as per the broadcaster's production guidelines used in the creation of the plano-stereoscopic 3DTV content, or accordingly modified by the broadcaster previous to transmission. Most of the content is expected to be most of the time within these disparity values, which shall be coded according to table 162. Disparity refers to the difference between the horizontal positions of a pixel in the left and right view that represents the same point in space. A positive value, right minus left, refers to a point away from the viewer and a negative value is one closer to the viewer.

Table 162: production_disparity_hint_info

| Syntax | Number of bits | Identifier |
|---|----------------|--------------------|
| production_disparity_hint_info() { video_max_disparity_hint video_min_disparity_hint } | 12 12 | tcimsbf tcimsbf |

Semantics for the production disparity hint info:

video_max_disparity_hint: This 12-bit field identifies the intended largest disparity according to the current production guidelines, which corresponds to an object at infinity, away from the viewer. If infinity disparity is unknown, then the disparity of the "furthest away object" should be given. This disparity is relative to a screen with a horizontal size of 11 520 pixels.

video_min_disparity_hint: This 12-bit field identifies the intended smallest disparity according to the current production guidelines, which corresponds to an object closest by the viewer. This disparity is relative to a screen with a horizontal size of 11 520 pixels.

6.5 Scoping rules for scoping descriptors

The information conveyed in some descriptors has a wider scope than the context in which the descriptor appears. Such descriptors are called scoping descriptors and the rules for their use are set forth in this clause.

The location of a particular instance of a Scoping Descriptor defines the scope over which the scoping properties shall apply. These properties shall apply to all entities (e.g. services or events) within this scope unless overridden at a scope of greater precedence.

The Scoping Descriptor that holds for entities that fall outside of the scope of all explicit signalling is not defined by the present document.

In increasing precedence order, the descriptor loops where a Scoping Descriptor may appear, if allowed by the type of scoping descriptor, are:

- 1) NIT first loop (lowest precedence)
- 2) BAT first loop
- 3) NIT TS loop
- 4) BAT TS loop
- 5) SDT
- 6) EIT (highest precedence)

7 Storage Media Interoperability (SMI) measures

7.0 Introduction

IEC 61883 [32] describes methods for delivering TS over the IEEE 1394.1 [33] to receivers. One likely source for this data is a digital storage device.

In certain cases TSs can be "incomplete", thus not conforming to the normal broadcast specifications. These "partial" TSs represent a subset of the data streams in the original TS. They may also be "discontinuous" - that is there may be changes in the TS or the subset of the TS presented and there may be temporal discontinuities. This clause on Storage Media Interoperability (SMI) describes the SI and PSI required in the delivered data in these cases.

7.1 SMI tables

7.1.0 General principles

The SMI tables are encoded using the private section syntax defined in ISO/IEC 13818-1 [15]. The SIT may be up to 4 096 bytes long.

The bitstream presented at a digital interface shall either be a "complete" TS conforming to TS 101 154 [9] and with SI conforming to the present document or it shall be "partial" TS. In the latter case the SI and PSI shall conform to the following clauses.

A "partial" TS shall not carry any SI tables other than the Selection Information Table (SIT) and Discontinuity Information Table (DIT) described in clause 7.1.1 and 7.1.2 respectively. The PSI shall be restricted to the PAT and PMT instances required to correctly describe the streams within the "partial" TS.

The presence of the SIT in a bitstream identifies the bitstream as a "partial" TS coming from a digital interface. In this case the receiver should not expect the SI information required in a broadcast TS and should instead rely on that carried by the SIT.

The SIT contains a summary of all relevant SI information contained in the broadcast stream. The DIT shall be inserted at transition points where SI information is discontinuous. The use of the SIT and DIT is restricted to partial TSs, they shall not be used in broadcasts.

7.1.1 Discontinuity Information Table (DIT)

The DIT (see table 163) is to be inserted at transition points at which SI information may be discontinuous.

Table 163: Discontinuity information section

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| <code>discontinuity_information_section() {</code> | | |
| <code>table_id</code> | 8 | uimsbf |
| <code>section_syntax_indicator</code> | 1 | bslbf |
| <code>reserved_future_use</code> | 1 | bslbf |
| <code>reserved</code> | 2 | bslbf |
| <code>section_length</code> | 12 | uimsbf |
| <code>transition_flag</code> | 1 | uimsbf |
| <code>reserved_future_use</code> | 7 | bslbf |
| <code>}</code> | | |

Semantics for the selection information section:

table_id: See table 2.

section_syntax_indicator: The `section_syntax_indicator` is a 1 bit field which shall be set to "0".

section_length: This is a 12 bit field, which is set to 0x001.

transition_flag: This 1 bit flag indicates the kind of transition in the TS. When the bit is set to "1", it indicates that the transition is due to a change of the originating source. The change of the originating source can be a change of

originating TS and/or a change of the position in the TS (e.g. in case of time-shift). When the bit is set to "0", it indicates that the transition is due to a change of the selection only, i.e. while staying within the same originating TS at the same position.

7.1.2 Selection Information Table (SIT)

The SIT describes the services and events carried by the "partial" TS. SIT sections shall be coded according to table 164.

Table 164: Selection information section

| Syntax | Number of bits | Identifier |
|-----------------------------------|----------------|------------|
| selection_information_section() { | | |
| table_id | 8 | uimsbf |
| section_syntax_indicator | 1 | bslbf |
| DVB_reserved_future_use | 1 | bslbf |
| ISO_reserved | 2 | bslbf |
| section_length | 12 | uimsbf |
| DVB_reserved_future_use | 16 | uimsbf |
| ISO_reserved | 2 | bslbf |
| version_number | 5 | uimsbf |
| current_next_indicator | 1 | bslbf |
| section_number | 8 | uimsbf |
| last_section_number | 8 | uimsbf |
| DVB_reserved_for_future_use | 4 | uimsbf |
| transmission_info_loop_length | 12 | bslbf |
| for (i = 0; i < N; i++) { | | |
| descriptor() | | |
| } | | |
| for (i=0; i < N; i++) { | | |
| service_id | 16 | uimsbf |
| DVB_reserved_future_use | 1 | uimsbf |
| running_status | 3 | bslbf |
| service_loop_length | 12 | bslbf |
| for (j=0; j < N; j++) { | | |
| descriptor() | | |
| } | | |
| } | | |
| CRC_32 | 32 | rpchof |
| } | | |

Semantics for the selection information section:

table_id: See table 2.

section_syntax_indicator: The section_syntax_indicator is a 1 bit field which shall be set to "1".

section_length: This is a 12-bit field, the first two bits of which shall be "00". 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 so that the entire section has a maximum length of 4 096 bytes.

version_number: This 5-bit field is the version number of the table. The version_number shall be incremented by 1 when a change in the information carried within the 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 table. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable table.

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

section_number: This 8 bit field gives the number of the section. The section_number shall be 0x00.

last_section_number: This 8 bit field specifies the number of the last section. The last_section_number shall be 0x00.

transmission_info_loop_length: This 12 bit field gives the total length in bytes of the following descriptor loop describing the transmission parameters of the partial TS.

service_id: This is a 16 bit field which serves as a label to identify this service from any other service within a TS. The service_id is the same as the program_number in the corresponding program_map_section.

running_status: This 3 bit field indicates the running status of the event in the original stream. This is the running status of the original present event. If no present event exists in the original stream the status is considered as "not running". The meaning of the running_status value is as defined in TS 101 211 [i.1].

service_loop_length: This 12 bit field gives the total length in bytes of the following descriptor loop containing SI related information on the service and event contained in the partial TS.

CRC_32: This is a 32 bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex A of ISO/IEC 13181-1 [15] after processing the entire section.

7.2 SMI descriptors

7.2.0 Introduction

This section contains syntax and semantics for descriptors exclusively found in partial TSs.

7.2.1 Partial Transport Stream (TS) descriptor

The transmission information descriptor loop of the SIT contains all the information required for controlling and managing the play-out and copying of partial TSs. The partial transport stream descriptor (see table 165) is proposed to describe this information.

Table 165: Partial Transport Stream (TS) descriptor

| Syntax | Number of bits | Identifier |
|---|----------------|------------|
| partial_transport_stream_descriptor() { | | |
| descriptor_tag | 8 | bslbf |
| descriptor_length | 8 | uimsbf |
| DVB_reserved_future_use | 2 | bslbf |
| peak_rate | 22 | uimsbf |
| DVB_reserved_future_use | 2 | bslbf |
| minimum_overall_smoothing_rate | 22 | uimsbf |
| DVB_reserved_future_use | 2 | bslbf |
| maximum_overall_smoothing_buffer | 14 | uimsbf |
| } | | |

Semantics for the partial TS descriptor:

peak_rate: The maximum momentary transport packet rate (i.e. 188 bytes divided by the time interval between start times of two succeeding TS packets). At least an upper bound for this peak_rate should be given. This 22-bit field is coded as a positive integer in units of 400 bit/s.

minimum_overall_smoothing_rate: Minimum smoothing buffer leak rate for the overall TS (all packets are covered). This 22-bit field is coded as a positive integer in units of 400 bit/s. The value 0x3FFFFFF is used to indicate that the minimum smoothing rate is undefined.

maximum_overall_smoothing_buffer: Maximum smoothing buffer size for the overall TS (all packets are covered). This 14-bit field is coded as a positive integer in units of 1 byte. The value 0x3FFF is used to indicate that the maximum smoothing buffer size is undefined.

Annex A (normative): Coding of text characters

A.0 General principles

Text items can optionally include information to select a wide range of character tables as indicated below.

If no character selection information is given in a text item, then the default character coding table (table 00 - Latin alphabet) of figure A.1 is assumed.

A.1 Control codes

For one-byte character tables, the codes in the range 0x80 to 0x9F are assigned to control functions as shown in table A.1.

Table A.1: Single byte control codes

| Control code | UTF-8 Encoded Control code | Description |
|--------------|----------------------------|-------------------------|
| 0x80 to 0x85 | 0xC2 0x80 to 0xC2 0x85 | reserved for future use |
| 0x86 | 0xC2 0x86 | character emphasis on |
| 0x87 | 0xC2 0x87 | character emphasis off |
| 0x88 to 0x89 | 0xC2 0x88 to 0xC2 0x89 | reserved for future use |
| 0x8A | 0xC2 0x8A | CR/LF |
| 0x8B to 0x9F | 0xC2 0x8B to 0xC2 0x9F | user defined |

For two-byte character tables, the codes in the range 0xE080 to 0xE09F are assigned to control functions as shown in table A.2.

Table A.2: DVB codes within private use area of ISO/IEC 10646 [14]

| Control code | UTF-8 Encoded Control code | Description |
|------------------|----------------------------------|-------------------------|
| 0xE080 to 0xE085 | 0xEE 0x82 0x80 to 0xEE 0x82 0x85 | reserved for future use |
| 0xE086 | 0xEE 0x82 0x86 | character emphasis on |
| 0xE087 | 0xEE 0x82 0x87 | character emphasis off |
| 0xE088 to 0xE089 | 0xEE 0x82 0x88 to 0xEE 0x82 0x89 | reserved for future use |
| 0xE08A | 0xEE 0x82 0x8A | CR/LF |
| 0xE08B to 0xE09F | 0xEE 0x82 0x8B to 0xEE 0x82 0x9F | reserved for future use |

A.2 Selection of character table

Text fields can optionally start with non-spacing, non-displayed data which specifies the alternative character table to be used for the remainder of the text item.

If the first byte of the text field has a value in the range "0x20" to "0xFF" then this and all subsequent bytes in the text item are coded using the default character coding table (table 00 - Latin alphabet) of figure A.1.

The selection of character table is indicated in table A.3.

Table A.3: Character coding tables

| First byte value | Character code table | Table description | Reproduced in figure |
|------------------|----------------------|-------------------------|----------------------|
| 0x01 | ISO/IEC 8859-5 [21] | Latin/Cyrillic alphabet | A.2 |
| 0x02 | ISO/IEC 8859-6 [22] | Latin/Arabic alphabet | A.3 |
| 0x03 | ISO/IEC 8859-7 [23] | Latin/Greek alphabet | A.4 |
| 0x04 | ISO/IEC 8859-8 [24] | Latin/Hebrew alphabet | A.5 |
| 0x05 | ISO/IEC 8859-9 [25] | Latin alphabet No. 5 | A.6 |
| 0x06 | ISO/IEC 8859-10 [26] | Latin alphabet No. 6 | A.7 |

| First byte value | Character code table | Table description | Reproduced in figure |
|---|--------------------------------------|---|----------------------|
| 0x07 | ISO/IEC 8859-11 [27] | Latin/Thai (draft only) | A.8 |
| 0x08 | reserved for future use (see note) | | |
| 0x09 | ISO/IEC 8859-13 [28] | Latin alphabet No. 7 | A.9 |
| 0x0A | ISO/IEC 8859-14 [29] | Latin alphabet No. 8 (Celtic) | A.10 |
| 0x0B | ISO/IEC 8859-15 [30] | Latin alphabet No. 9 | A.11 |
| 0x0C to 0x0F | reserved for future use | | |
| 0x10 | ISO/IEC 8859 | See table A.4 | |
| 0x11 | ISO/IEC 10646 [14] | Basic Multilingual Plane (BMP) | |
| 0x12 | KSX1001-2004 [37] | Korean Character Set | |
| 0x13 | GB-2312-1980 [50] | Simplified Chinese Character | |
| 0x14 | Big5 subset of ISO/IEC 10646 [14] | Traditional Chinese | |
| 0x15 | UTF-8 encoding of ISO/IEC 10646 [14] | Basic Multilingual Plane (BMP) | |
| 0x16 to 0x1E | reserved for future use | | |
| 0x1F | Described by encoding_type_id | Described by 8 bit encoding_type_id conveyed in second byte of the string | |
| NOTE: The first byte value 0x08 was previously reserved for ISO/IEC 8859-12 which should have covered the Devanagiri script. ISO/IEC however never published part 12 of the ISO/IEC 8859 series (see http://en.wikipedia.org/wiki/ISO/IEC_8859-12). Applications of the current document for the scripts used by South-Asian languages should use the BMP of ISO/IEC 10646 [14], where appropriate glyphs are provided. | | | |

If the first byte of the text field has value "0x1F" then the following byte carries an 8-bit value (uimbsf) containing the encoding_type_id. This value indicates the encoding scheme of the string. It shall be coded according to TS 101 162 [57].

If the first byte of the text field has a value "0x10" then the following two bytes carry a 16-bit value (uimbsf) N to indicate that the remaining data of the text field is coded using the character code table specified in table A.4.

Table A.4: Character Coding Tables for first byte 0x10

| First byte value | Second byte value | Third Byte Value | Selected character code table | Table description | Reproduced in figure |
|------------------|-------------------|------------------|-------------------------------|-------------------------------|----------------------|
| 0x10 | 0x00 | 0x00 | reserved for future use | | |
| 0x10 | 0x00 | 0x01 | ISO/IEC 8859-1 [17] | West European | |
| 0x10 | 0x00 | 0x02 | ISO/IEC 8859-2 [18] | East European | |
| 0x10 | 0x00 | 0x03 | ISO/IEC 8859-3 [19] | South European | |
| 0x10 | 0x00 | 0x04 | ISO/IEC 8859-4 [20] | North and North-East European | |
| 0x10 | 0x00 | 0x05 | ISO/IEC 8859-5 [21] | Latin/Cyrillic | A.2 |
| 0x10 | 0x00 | 0x06 | ISO/IEC 8859-6 [22] | Latin/Arabic | A.3 |
| 0x10 | 0x00 | 0x07 | ISO/IEC 8859-7 [23] | Latin/Greek | A.4 |
| 0x10 | 0x00 | 0x08 | ISO/IEC 8859-8 [24] | Latin/Hebrew | A.5 |
| 0x10 | 0x00 | 0x09 | ISO/IEC 8859-9 [25] | West European & Turkish | A.6 |
| 0x10 | 0x00 | 0x0A | ISO/IEC 8859-10 [26] | North European | A.7 |
| 0x10 | 0x00 | 0x0B | ISO/IEC 8859-11 [27] | Thai | A.8 |
| 0x10 | 0x00 | 0x0C | | Reserved for future use | |
| 0x10 | 0x00 | 0x0D | ISO/IEC 8859-13 [28] | Baltic | A.9 |
| 0x10 | 0x00 | 0x0E | ISO/IEC 8859-14 [29] | Celtic | A.10 |
| 0x10 | 0x00 | 0x0F | ISO/IEC 8859-15 [30] | West European | A.11 |
| 0x10 | 0x00 | 0x10 to 0xFF | reserved for future use | | |
| 0x10 | 0x01 to 0xFF | 0x00 to 0xFF | reserved for future use | | |

The figures A.1 through A.11 below use the following conventions:

- Colour coding is used to indicate the general glyph category:

| Colour | Description |
|--------------|---|
| light orange | letters of the Latin alphabet which are compatible with 7-bit US-ASCII encoding |
| light red | numbers of the Latin alphabet which are compatible with 7-bit US-ASCII encoding |
| light blue | marks, punctuation, symbols, and separators |
| light pink | non-spacing symbols (diacritical marks) |
| light green | region-specific alphabet symbols |

- Non-printing characters use the following mnemonics:

| Mnemonic | Description |
|----------|--------------------|
| SPC | space |
| NBSP | no-break space |
| SHY | soft hyphen |
| LRM | left-to-right mark |
| RLM | right-to-left mark |

| | | Second Nibble | | | | | | | | | | | | | | | |
|---|----|---------------|-----------|-----------|-----------|------------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-------------|
| | | -0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |
| F i r s t N i b b l e | 0- | | | | | | | | | | | | | | | | |
| | 1- | | | | | | | | | | | | | | | | |
| | 2- | SPC 0020 | ! 0021 | " 0022 | # 0023 | \$ 0024 | % 0025 | & 0026 | ' 0027 | (0028 |) 0029 | * 002A | + 002B | , 002C | - 002D | . 002E | / 002F |
| | 3- | 0 0030 | 1 0031 | 2 0032 | 3 0033 | 4 0034 | 5 0035 | 6 0036 | 7 0037 | 8 0038 | 9 0039 | : 003A | ; 003B | < 003C | = 003D | > 003E | ? 003F |
| | 4- | @ 0040 | A 0041 | B 0042 | C 0043 | D 0044 | E 0045 | F 0046 | G 0047 | H 0048 | I 0049 | J 004A | K 004B | L 004C | M 004D | N 004E | O 004F |
| | 5- | P 0050 | Q 0051 | R 0052 | S 0053 | T 0054 | U 0055 | V 0056 | W 0057 | X 0058 | Y 0059 | Z 005A | [005B | \ 005C |] 005D | ^ 005E | _ 005F |
| | 6- | ` 0060 | a 0061 | b 0062 | c 0063 | d 0064 | e 0065 | f 0066 | g 0067 | h 0068 | i 0069 | j 006A | k 006B | l 006C | m 006D | n 006E | o 006F |
| | 7- | p 0070 | q 0071 | r 0072 | s 0073 | t 0074 | u 0075 | v 0076 | w 0077 | x 0078 | y 0079 | z 007A | { 007B | 007C | } 007D | ~ 007E | |
| | 8- | | | | | | | | | | | | | | | | |
| | 9- | | | | | | | | | | | | | | | | |
| | A- | NBSP 00A0 | ¡ 00A1 | ¢ 00A2 | £ 00A3 | € 20AC | ¥ 00A5 | | § 00A7 | ¤ 00A4 | ' 2018 | " 201C | « 00AB | ← 2190 | ↑ 2191 | → 2192 | ↓ 2193 |
| | B- | ° 00B0 | ± 00B1 | ² 00B2 | ³ 00B3 | × 00D7 | μ 00B5 | ¶ 00B6 | · 00B7 | ÷ 00F7 | ' 2019 | " 201D | » 00BB | ¼ 00BC | ½ 00BD | ¾ 00BE | ¿ 00BF |
| | C- | | ³ 0300 | ² 0301 | ¹ 0302 | ˆ 0303 | ˜ 0304 | ˘ 0306 | ˙ 0307 | ˚ 0308 | | ˆ 030A | ˜ 0327 | | ˘ 030B | ˙ 0328 | ˚ 030C |
| | D- | — 2015 | ¹ 00B9 | ® 00AE | © 00A9 | ™ 2122 | ♪ 266A | ¬ 00AC | ¡ 00A6 | | | | | ⅛ 215B | ⅜ 215C | ⅝ 215D | ⅞ 215E |
| | E- | Ω 2126 | Æ 00C6 | Ð 0110 | ª 00AA | ¨ 0126 | | ¡ 0132 | £ 013F | Ł 0141 | Ø 00D8 | Œ 0152 | ° 00BA | Ɔ 00DE | Ʀ 0166 | Ŋ 014A | ʰ 0149 |
| | F- | κ 0138 | æ 00E6 | đ 0111 | ð 00F0 | ħ 0127 | ı 0131 | ij 0133 | ı 0140 | ı 0142 | ø 00F8 | œ 0153 | ß 00DF | Ɔ 00FE | Ʀ 0167 | Ŋ 014B | SHY 00AD |
| | | 0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |

NOTE: This table is a superset of ISO/IEC 6937 [16] with addition of the Euro symbol (U+20AC) in position 0xA4.

Figure A.1: Character code table 00 - Latin alphabet with Unicode equivalents

| | | Second Nibble | | | | | | | | | | | | | | | |
|---|----|---------------|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-------------|-----------|-----------|
| | | -0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |
| F i r s t N i b b l e | 0- | | | | | | | | | | | | | | | | |
| | 1- | | | | | | | | | | | | | | | | |
| | 2- | SPC 0020 | ! 0021 | " 0022 | # 0023 | \$ 0024 | % 0025 | & 0026 | ' 0027 | (0028 |) 0029 | * 002A | + 002B | , 002C | - 002D | . 002E | / 002F |
| | 3- | 0 0030 | 1 0031 | 2 0032 | 3 0033 | 4 0034 | 5 0035 | 6 0036 | 7 0037 | 8 0038 | 9 0039 | : 003A | ; 003B | < 003C | = 003D | > 003E | ? 003F |
| | 4- | @ 0040 | A 0041 | B 0042 | C 0043 | D 0044 | E 0045 | F 0046 | G 0047 | H 0048 | I 0049 | J 004A | K 004B | L 004C | M 004D | N 004E | O 004F |
| | 5- | P 0050 | Q 0051 | R 0052 | S 0053 | T 0054 | U 0055 | V 0056 | W 0057 | X 0058 | Y 0059 | Z 005A | [005B | \ 005C |] 005D | ^ 005E | _ 005F |
| | 6- | ` 0060 | a 0061 | b 0062 | c 0063 | d 0064 | e 0065 | f 0066 | g 0067 | h 0068 | i 0069 | j 006A | k 006B | l 006C | m 006D | n 006E | o 006F |
| | 7- | p 0070 | q 0071 | r 0072 | s 0073 | t 0074 | u 0075 | v 0076 | w 0077 | x 0078 | y 0079 | z 007A | { 007B | 007C | } 007D | ~ 007E | |
| | 8- | | | | | | | | | | | | | | | | |
| | 9- | | | | | | | | | | | | | | | | |
| | A- | NBSP 00A0 | Ё 0401 | Ђ 0402 | Ѓ 0403 | Є 0404 | Ѕ 0405 | І 0406 | Ї 0407 | Ј 0408 | Љ 0409 | Њ 040A | Ћ 040B | Ќ 040C | SHY 00AD | Ў 040E | Џ 040F |
| | B- | А 0410 | Б 0411 | В 0412 | Г 0413 | Д 0414 | Е 0415 | Ж 0416 | З 0417 | И 0418 | Й 0419 | К 041A | Л 041B | М 041C | Н 041D | О 041E | П 041F |
| | C- | Р 0420 | С 0421 | Т 0422 | У 0423 | Ф 0424 | Х 0425 | Ц 0426 | Ч 0427 | Ш 0428 | Щ 0429 | Ъ 042A | Ы 042B | Ь 042C | Э 042D | Ю 042E | Я 042F |
| | D- | а 0430 | б 0431 | в 0432 | г 0433 | д 0434 | е 0435 | ж 0436 | з 0437 | и 0438 | й 0439 | к 043A | л 043B | м 043C | н 043D | о 043E | п 043F |
| | E- | р 0440 | с 0441 | т 0442 | у 0443 | ф 0444 | х 0445 | ц 0446 | ч 0447 | ш 0448 | щ 0449 | ъ 044A | ы 044B | ь 044C | э 044D | ю 044E | я 044F |
| | F- | Ѕ 2116 | ё 0451 | ђ 0452 | ѓ 0453 | є 0454 | ѕ 0455 | і 0456 | ї 0457 | ј 0458 | љ 0459 | њ 045A | ћ 045B | ќ 045C | § 00A7 | ў 045E | џ 045F |
| | | 0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |

NOTE 1: For the Ruthenian language, the characters in code positions 0xA5 (S) and 0xF5 s are replaced by І and Ї, respectively.

NOTE 2: This is compatible with ISO/IEC 8859-5 [21].

Figure A.2: Character code table 01 - Latin/Cyrillic alphabet with Unicode equivalents

| | | Second Nibble | | | | | | | | | | | | | | | |
|---|----|---------------|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-------------|-----------|-----------|
| | | -0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |
| F i r s t N i b b l e | 0- | | | | | | | | | | | | | | | | |
| | 1- | | | | | | | | | | | | | | | | |
| | 2- | SPC 0020 | ! 0021 | * 0022 | # 0023 | \$ 0024 | % 0025 | & 0026 | ' 0027 | (0028 |) 0029 | * 002A | + 002B | , 002C | - 002D | . 002E | / 002F |
| | 3- | 0 0030 | 1 0031 | 2 0032 | 3 0033 | 4 0034 | 5 0035 | 6 0036 | 7 0037 | 8 0038 | 9 0039 | : 003A | ; 003B | < 003C | = 003D | > 003E | ? 003F |
| | 4- | @ 0040 | A 0041 | B 0042 | C 0043 | D 0044 | E 0045 | F 0046 | G 0047 | H 0048 | I 0049 | J 004A | K 004B | L 004C | M 004D | N 004E | O 004F |
| | 5- | P 0050 | Q 0051 | R 0052 | S 0053 | T 0054 | U 0055 | V 0056 | W 0057 | X 0058 | Y 0059 | Z 005A | [005B | \ 005C |] 005D | ^ 005E | _ 005F |
| | 6- | ` 0060 | a 0061 | b 0062 | c 0063 | d 0064 | e 0065 | f 0066 | g 0067 | h 0068 | i 0069 | j 006A | k 006B | l 006C | m 006D | n 006E | o 006F |
| | 7- | p 0070 | q 0071 | r 0072 | s 0073 | t 0074 | u 0075 | v 0076 | w 0077 | x 0078 | y 0079 | z 007A | { 007B | 007C | } 007D | ~ 007E | |
| | 8- | | | | | | | | | | | | | | | | |
| | 9- | | | | | | | | | | | | | | | | |
| | A- | NBSP 00A0 | | | | □ 00A4 | | | | | | | | · 060C | SHY 00AD | | |
| | B- | | | | | | | | | | | | · 061B | | | | · 061F |
| | C- | | ء 0621 | آ 0622 | أ 0623 | ؤ 0624 | إ 0625 | ئ 0626 | ا 0627 | ب 0628 | ة 0629 | ت 062A | ث 062B | ج 062C | ح 062D | خ 062E | د 062F |
| | D- | ذ 0630 | ر 0631 | ز 0632 | س 0633 | ش 0634 | ص 0635 | ض 0636 | ط 0637 | ظ 0638 | ع 0639 | غ 063A | | | | | |
| | E- | ـ 0640 | ف 0641 | ق 0642 | ك 0643 | ل 0644 | م 0645 | ن 0646 | ه 0647 | و 0648 | ى 0649 | ي 064A | ٴ 064B | ٴ 064C | ٴ 064D | ٴ 064E | ٴ 064F |
| | F- | ٴ 0650 | ٴ 0651 | ٴ 0652 | | | | | | | | | | | | | |
| | | 0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |

NOTE: This is compatible with ISO/IEC 8859-6 [22].

Figure A.3: Character code table 02 - Latin/Arabic alphabet with Unicode equivalents

| | | Second Nibble | | | | | | | | | | | | | | | |
|---|----|---------------|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-------------|-----------|-----------|
| | | -0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |
| F i r s t N i b b l e | 0- | | | | | | | | | | | | | | | | |
| | 1- | | | | | | | | | | | | | | | | |
| | 2- | SPC 0020 | ! 0021 | " 0022 | # 0023 | \$ 0024 | % 0025 | & 0026 | ' 0027 | (0028 |) 0029 | * 002A | + 002B | , 002C | - 002D | . 002E | / 002F |
| | 3- | 0 0030 | 1 0031 | 2 0032 | 3 0033 | 4 0034 | 5 0035 | 6 0036 | 7 0037 | 8 0038 | 9 0039 | : 003A | ; 003B | < 003C | = 003D | > 003E | ? 003F |
| | 4- | @ 0040 | A 0041 | B 0042 | C 0043 | D 0044 | E 0045 | F 0046 | G 0047 | H 0048 | I 0049 | J 004A | K 004B | L 004C | M 004D | N 004E | O 004F |
| | 5- | P 0050 | Q 0051 | R 0052 | S 0053 | T 0054 | U 0055 | V 0056 | W 0057 | X 0058 | Y 0059 | Z 005A | [005B | \ 005C |] 005D | ^ 005E | _ 005F |
| | 6- | ` 0060 | a 0061 | b 0062 | c 0063 | d 0064 | e 0065 | f 0066 | g 0067 | h 0068 | i 0069 | j 006A | k 006B | l 006C | m 006D | n 006E | o 006F |
| | 7- | p 0070 | q 0071 | r 0072 | s 0073 | t 0074 | u 0075 | v 0076 | w 0077 | x 0078 | y 0079 | z 007A | { 007B | 007C | } 007D | ~ 007E | |
| | 8- | | | | | | | | | | | | | | | | |
| | 9- | | | | | | | | | | | | | | | | |
| | A- | NBSP 00A0 | ‘ 2018 | ’ 2019 | £ 00A3 | € 20AC | ƒ 20AF | ¡ 00A6 | § 00A7 | “ 00AB | ” 00A9 | © 00A9 | « 037A | » 00AB | SHY 00AC | | — 2015 |
| | B- | ° 00B0 | ± 00B1 | ² 00B2 | ³ 00B3 | ´ 0384 | µ 0385 | À 0386 | Á 0387 | Â 0388 | Ã 0389 | Ä 038A | Å 00BB | » 038C | ½ 00BD | ¾ 038E | Ω 038F |
| | C- | İ 0390 | À 0391 | Á 0392 | Â 0393 | Ã 0394 | Ä 0395 | Å 0396 | Æ 0397 | Ç 0398 | È 0399 | É 039A | Ê 039B | Ë 039C | Ì 039D | Í 039E | Î 039F |
| | D- | Π 03A0 | Ρ 03A1 | | Σ 03A3 | Τ 03A4 | Υ 03A5 | Φ 03A6 | Χ 03A7 | Ψ 03A8 | Ω 03A9 | Ϊ 03AA | Ϋ 03AB | ά 03AC | έ 03AD | ή 03AE | ί 03AF |
| | E- | Ú 03B0 | α 03B1 | β 03B2 | γ 03B3 | δ 03B4 | ε 03B5 | ζ 03B6 | η 03B7 | θ 03B8 | ι 03B9 | κ 03BA | λ 03BB | μ 03BC | ν 03BD | ξ 03BE | ο 03BF |
| | F- | π 03C0 | ρ 03C1 | ς 03C2 | σ 03C3 | τ 03C4 | υ 03C5 | φ 03C6 | χ 03C7 | ψ 03C8 | ω 03C9 | ϊ 03CA | ϋ 03CB | ό 03CC | ύ 03CD | ώ 03CE | |
| | | 0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |

NOTE: This is compatible with ISO/IEC 8859-7 [23].

Figure A.4: Character code table 03 - Latin/Greek alphabet with Unicode equivalents

| | | Second Nibble | | | | | | | | | | | | | | | |
|---|----|---------------|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-------------|-------------|-----------|
| | | -0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |
| F i r s t N i b b l e | 0- | | | | | | | | | | | | | | | | |
| | 1- | | | | | | | | | | | | | | | | |
| | 2- | SPC 0020 | ! 0021 | " 0022 | # 0023 | \$ 0024 | % 0025 | & 0026 | ' 0027 | (0028 |) 0029 | * 002A | + 002B | , 002C | - 002D | . 002E | / 002F |
| | 3- | 0 0030 | 1 0031 | 2 0032 | 3 0033 | 4 0034 | 5 0035 | 6 0036 | 7 0037 | 8 0038 | 9 0039 | : 003A | ; 003B | < 003C | = 003D | > 003E | ? 003F |
| | 4- | @ 0040 | A 0041 | B 0042 | C 0043 | D 0044 | E 0045 | F 0046 | G 0047 | H 0048 | I 0049 | J 004A | K 004B | L 004C | M 004D | N 004E | O 004F |
| | 5- | P 0050 | Q 0051 | R 0052 | S 0053 | T 0054 | U 0055 | V 0056 | W 0057 | X 0058 | Y 0059 | Z 005A | [005B | \ 005C |] 005D | ^ 005E | _ 005F |
| | 6- | ` 0060 | a 0061 | b 0062 | c 0063 | d 0064 | e 0065 | f 0066 | g 0067 | h 0068 | i 0069 | j 006A | k 006B | l 006C | m 006D | n 006E | o 006F |
| | 7- | p 0070 | q 0071 | r 0072 | s 0073 | t 0074 | u 0075 | v 0076 | w 0077 | x 0078 | y 0079 | z 007A | { 007B | 007C | } 007D | ~ 007E | |
| | 8- | | | | | | | | | | | | | | | | |
| | 9- | | | | | | | | | | | | | | | | |
| | A- | NBSP 00A0 | | ¢ 00A2 | £ 00A3 | ¤ 00A4 | ¥ 00A5 | ¦ 00A6 | § 00A7 | ¨ 00A8 | © 00A9 | × 00AD | « 00AB | ¬ 00AC | SHY 00AD | ® 00AE | ™ 00AF |
| | B- | ° 00B0 | ± 00B1 | ² 00B2 | ³ 00B3 | ´ 00B4 | µ 00B5 | ¶ 00B6 | · 00B7 | ¸ 00B8 | ¹ 00B9 | ÷ 00BF | » 00BB | ¼ 00BC | ½ 00BD | ¾ 00BE | |
| | C- | | | | | | | | | | | | | | | | |
| | D- | | | | | | | | | | | | | | | | ™ 2017 |
| | E- | א 05D0 | ב 05D1 | ג 05D2 | ד 05D3 | ה 05D4 | ו 05D5 | ז 05D6 | ח 05D7 | ט 05D8 | י 05D9 | ך 05DA | כ 05DB | ל 05DC | ם 05DD | נ 05DE | ן 05DF |
| | F- | ס 05E0 | ע 05E1 | פ 05E2 | ף 05E3 | צ 05E4 | ץ 05E5 | ק 05E6 | ר 05E7 | ש 05E8 | ת 05E9 | | | | LRM 200E | RLM 200F | |
| | | 0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |

NOTE: This is compatible with ISO/IEC 8859-8 [24].

Figure A.5: Character code table 04 - Latin/Hebrew alphabet with Unicode equivalents

| | | Second Nibble | | | | | | | | | | | | | | | |
|---|----|---------------|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-------------|-----------|-----------|
| | | -0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |
| F i r s t N i b b l e | 0- | | | | | | | | | | | | | | | | |
| | 1- | | | | | | | | | | | | | | | | |
| | 2- | SPC 0020 | ! 0021 | “ 0022 | # 0023 | \$ 0024 | % 0025 | & 0026 | ' 0027 | (0028 |) 0029 | * 002A | + 002B | , 002C | - 002D | . 002E | / 002F |
| | 3- | 0 0030 | 1 0031 | 2 0032 | 3 0033 | 4 0034 | 5 0035 | 6 0036 | 7 0037 | 8 0038 | 9 0039 | : 003A | ; 003B | < 003C | = 003D | > 003E | ? 003F |
| | 4- | @ 0040 | A 0041 | B 0042 | C 0043 | D 0044 | E 0045 | F 0046 | G 0047 | H 0048 | I 0049 | J 004A | K 004B | L 004C | M 004D | N 004E | O 004F |
| | 5- | P 0050 | Q 0051 | R 0052 | S 0053 | T 0054 | U 0055 | V 0056 | W 0057 | X 0058 | Y 0059 | Z 005A | [005B | \ 005C |] 005D | ^ 005E | _ 005F |
| | 6- | ` 0060 | a 0061 | b 0062 | c 0063 | d 0064 | e 0065 | f 0066 | g 0067 | h 0068 | i 0069 | j 006A | k 006B | l 006C | m 006D | n 006E | o 006F |
| | 7- | p 0070 | q 0071 | r 0072 | s 0073 | t 0074 | u 0075 | v 0076 | w 0077 | x 0078 | y 0079 | z 007A | { 007B | 007C | } 007D | ~ 007E | |
| | 8- | | | | | | | | | | | | | | | | |
| | 9- | | | | | | | | | | | | | | | | |
| | A- | NBSP 00A0 | ı 00A1 | ¢ 00A2 | £ 00A3 | ¤ 00A4 | ¥ 00A5 | ¦ 00A6 | § 00A7 | ¨ 00A8 | © 00A9 | ª 00AA | « 00AB | ¬ 00AC | SHY 00AD | ® 00AE | — 00AF |
| | B- | ° 00B0 | ± 00B1 | ² 00B2 | ³ 00B3 | ´ 00B4 | µ 00B5 | ¶ 00B6 | · 00B7 | ¸ 00B8 | ¹ 00B9 | º 00BA | » 00BB | ¼ 00BC | ½ 00BD | ¾ 00BE | ¿ 00BF |
| | C- | À 00C0 | Á 00C1 | Â 00C2 | Ã 00C3 | Ä 00C4 | Å 00C5 | Æ 00C6 | Ç 00C7 | È 00C8 | É 00C9 | Ê 00CA | Ë 00CB | Ì 00CC | Í 00CD | Î 00CE | Ï 00CF |
| | D- | Ğ 011E | Ñ 00D1 | Ò 00D2 | Ó 00D3 | Ô 00D4 | Õ 00D5 | Ö 00D6 | × 00D7 | Ø 00D8 | Ù 00D9 | Ú 00DA | Û 00DB | Ü 00DC | İ 0130 | Ş 015E | ß 00DF |
| | E- | à 00E0 | á 00E1 | â 00E2 | ã 00E3 | ä 00E4 | å 00E5 | æ 00E6 | ç 00E7 | è 00E8 | é 00E9 | ê 00EA | ë 00EB | ì 00EC | í 00ED | î 00EE | ï 00EF |
| | F- | ğ 011F | ñ 00F1 | ò 00F2 | ó 00F3 | ô 00F4 | õ 00F5 | ö 00F6 | ÷ 00F7 | ø 00F8 | ù 00F9 | ú 00FA | û 00FB | ü 00FC | ı 0131 | ş 015F | ÿ 00FF |
| | | 0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |

NOTE: This is compatible with ISO/IEC 8859-9 [25].

Figure A.6: Character code table 05 - Latin/Turkish alphabet with Unicode equivalents

| | | Second Nibble | | | | | | | | | | | | | | | |
|---|----|---------------|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-------------|-----------|-----------|
| | | -0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |
| F i r s t N i b b l e | 0- | | | | | | | | | | | | | | | | |
| | 1- | | | | | | | | | | | | | | | | |
| | 2- | SPC 0020 | ! 0021 | " 0022 | # 0023 | \$ 0024 | % 0025 | & 0026 | ' 0027 | (0028 |) 0029 | * 002A | + 002B | , 002C | - 002D | . 002E | / 002F |
| | 3- | 0 0030 | 1 0031 | 2 0032 | 3 0033 | 4 0034 | 5 0035 | 6 0036 | 7 0037 | 8 0038 | 9 0039 | : 003A | ; 003B | < 003C | = 003D | > 003E | ? 003F |
| | 4- | @ 0040 | A 0041 | B 0042 | C 0043 | D 0044 | E 0045 | F 0046 | G 0047 | H 0048 | I 0049 | J 004A | K 004B | L 004C | M 004D | N 004E | O 004F |
| | 5- | P 0050 | Q 0051 | R 0052 | S 0053 | T 0054 | U 0055 | V 0056 | W 0057 | X 0058 | Y 0059 | Z 005A | [005B | \ 005C |] 005D | ^ 005E | _ 005F |
| | 6- | ` 0060 | a 0061 | b 0062 | c 0063 | d 0064 | e 0065 | f 0066 | g 0067 | h 0068 | i 0069 | j 006A | k 006B | l 006C | m 006D | n 006E | o 006F |
| | 7- | p 0070 | q 0071 | r 0072 | s 0073 | t 0074 | u 0075 | v 0076 | w 0077 | x 0078 | y 0079 | z 007A | { 007B | 007C | } 007D | ~ 007E | |
| | 8- | | | | | | | | | | | | | | | | |
| | 9- | | | | | | | | | | | | | | | | |
| | A- | NBSP 00A0 | À 0104 | Á 0112 | Â 0122 | Ã 012A | Ä 0128 | Å 0136 | Š 00A7 | Ł 013B | Đ 0110 | Š 0160 | Ŧ 0166 | Ž 017D | SHY 00AD | Ů 016A | Ň 014A |
| | B- | ° 00B0 | à 0105 | á 0113 | â 0123 | ã 012B | ä 0129 | å 0137 | · 00B7 | ł 013C | đ 0111 | š 0161 | ŧ 0167 | ž 017E | — 2015 | ů 016B | ň 014B |
| | C- | Ā 0100 | Ā 00C1 | Ā 00C2 | Ā 00C3 | Ā 00C4 | Ā 00C5 | Æ 00C6 | Ĳ 012E | Č 010C | É 00C9 | Ę 0118 | Ě 00CB | È 0116 | Í 00CD | Î 00CE | Ī 00CF |
| | D- | Ð 00D0 | Ñ 0145 | Ŏ 014C | Ŏ 00D3 | Ŏ 00D4 | Ŏ 00D5 | Ŏ 00D6 | Ů 0168 | Ø 00D8 | Ų 0172 | Ú 00DA | Û 00DB | Ü 00DC | Ý 00DD | Þ 00DE | ß 00DF |
| | E- | ā 0101 | á 00E1 | â 00E2 | ă 00E3 | ä 00E4 | å 00E5 | æ 00E6 | ĵ 012F | č 010D | é 00E9 | ę 0119 | ě 00EB | è 0117 | í 00ED | î 00EE | ï 00EF |
| | F- | ò 00F0 | ñ 0146 | ō 014D | ó 00F3 | ô 00F4 | õ 00F5 | ö 00F6 | ū 0169 | ø 00F8 | ų 0173 | ú 00FA | û 00FB | ü 00FC | ý 00FD | þ 00FE | κ 0138 |
| | | 0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |

NOTE: This is compatible with ISO/IEC 8859-10 [26].

Figure A.7: Character code table 06 - Latin/Nordic alphabet with Unicode equivalents

| | | Second Nibble | | | | | | | | | | | | | | | |
|---|----|---------------|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | -0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |
| F i r s t N i b b l e | 0- | | | | | | | | | | | | | | | | |
| | 1- | | | | | | | | | | | | | | | | |
| | 2- | SPC 0020 | ! 0021 | " 0022 | # 0023 | \$ 0024 | % 0025 | & 0026 | ' 0027 | (0028 |) 0029 | * 002A | + 002B | , 002C | - 002D | . 002E | / 002F |
| | 3- | 0 0030 | 1 0031 | 2 0032 | 3 0033 | 4 0034 | 5 0035 | 6 0036 | 7 0037 | 8 0038 | 9 0039 | : 003A | ; 003B | < 003C | = 003D | > 003E | ? 003F |
| | 4- | @ 0040 | A 0041 | B 0042 | C 0043 | D 0044 | E 0045 | F 0046 | G 0047 | H 0048 | I 0049 | J 004A | K 004B | L 004C | M 004D | N 004E | O 004F |
| | 5- | P 0050 | Q 0051 | R 0052 | S 0053 | T 0054 | U 0055 | V 0056 | W 0057 | X 0058 | Y 0059 | Z 005A | [005B | \ 005C |] 005D | ^ 005E | _ 005F |
| | 6- | ` 0060 | a 0061 | b 0062 | c 0063 | d 0064 | e 0065 | f 0066 | g 0067 | h 0068 | i 0069 | j 006A | k 006B | l 006C | m 006D | n 006E | o 006F |
| | 7- | p 0070 | q 0071 | r 0072 | s 0073 | t 0074 | u 0075 | v 0076 | w 0077 | x 0078 | y 0079 | z 007A | { 007B | 007C | } 007D | ~ 007E | |
| | 8- | | | | | | | | | | | | | | | | |
| | 9- | | | | | | | | | | | | | | | | |
| | A- | NBSP 00A0 | ก 0E01 | ข 0E02 | ช 0E03 | ค 0E04 | ด 0E05 | พ 0E06 | ง 0E07 | จ 0E08 | ฉ 0E09 | ช 0E0A | ซ 0E0B | ณ 0E0C | ญ 0E0D | อ 0E0E | โ 0E0F |
| | B- | ฐ 0E10 | ฑ 0E11 | ฒ 0E12 | ณ 0E13 | ต 0E14 | ด 0E15 | ถ 0E16 | ท 0E17 | ธ 0E18 | น 0E19 | บ 0E1A | ป 0E1B | ผ 0E1C | ฝ 0E1D | พ 0E1E | ฟ 0E1F |
| | C- | ภ 0E20 | ม 0E21 | ย 0E22 | ร 0E23 | ฤ 0E24 | ล 0E25 | ภ 0E26 | ว 0E27 | ศ 0E28 | ษ 0E29 | ส 0E2A | ห 0E2B | พ 0E2C | อ 0E2D | ธ 0E2E | า 0E2F |
| | D- | ะ 0E30 | ั 0E31 | า 0E32 | ำ 0E33 | เ 0E34 | อ 0E35 | โ 0E36 | ั 0E37 | อ 0E38 | อ 0E39 | อ 0E3A | | | | | ฿ 0E3F |
| | E- | เ 0E40 | แ 0E41 | โ 0E42 | ไ 0E43 | ใ 0E44 | ง 0E45 | ง 0E46 | อ 0E47 | อ 0E48 | อ 0E49 | อ 0E4A | อ 0E4B | อ 0E4C | อ 0E4D | อ 0E4E | อ 0E4F |
| | F- | อ 0E50 | อ 0E51 | อ 0E52 | อ 0E53 | อ 0E54 | อ 0E55 | อ 0E56 | อ 0E57 | อ 0E58 | อ 0E59 | อ 0E5A | อ 0E5B | | | | |
| | | 0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |

NOTE: This is compatible with ISO/IEC 8859-11 [27].

Figure A.8: Character code table 07 - Latin/Thai alphabet with Unicode equivalents

| | | Second Nibble | | | | | | | | | | | | | | | |
|---|----|---------------|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-------------|-----------|-----------|
| | | -0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |
| F i r s t N i b b l e | 0- | | | | | | | | | | | | | | | | |
| | 1- | | | | | | | | | | | | | | | | |
| | 2- | SPC 0020 | ! 0021 | “ 0022 | # 0023 | \$ 0024 | % 0025 | & 0026 | ‘ 0027 | (0028 |) 0029 | * 002A | + 002B | , 002C | - 002D | . 002E | / 002F |
| | 3- | 0 0030 | 1 0031 | 2 0032 | 3 0033 | 4 0034 | 5 0035 | 6 0036 | 7 0037 | 8 0038 | 9 0039 | : 003A | ; 003B | < 003C | = 003D | > 003E | ? 003F |
| | 4- | @ 0040 | A 0041 | B 0042 | C 0043 | D 0044 | E 0045 | F 0046 | G 0047 | H 0048 | I 0049 | J 004A | K 004B | L 004C | M 004D | N 004E | O 004F |
| | 5- | P 0050 | Q 0051 | R 0052 | S 0053 | T 0054 | U 0055 | V 0056 | W 0057 | X 0058 | Y 0059 | Z 005A | [005B | \ 005C |] 005D | ^ 005E | _ 005F |
| | 6- | ` 0060 | a 0061 | b 0062 | c 0063 | d 0064 | e 0065 | f 0066 | g 0067 | h 0068 | i 0069 | j 006A | k 006B | l 006C | m 006D | n 006E | o 006F |
| | 7- | p 0070 | q 0071 | r 0072 | s 0073 | t 0074 | u 0075 | v 0076 | w 0077 | x 0078 | y 0079 | z 007A | { 007B | 007C | } 007D | ~ 007E | |
| | 8- | | | | | | | | | | | | | | | | |
| | 9- | | | | | | | | | | | | | | | | |
| | A- | NBSP 00A0 | ” 201D | ¢ 00A2 | £ 00A3 | ¤ 00A4 | ” 201E | ¡ 00A6 | § 00A7 | Ø 00D8 | © 00A9 | Ŕ 0156 | « 00AB | ¬ 00AC | SHY 00AD | ® 00AE | Æ 00C6 |
| | B- | ° 00B0 | ± 00B1 | ² 00B2 | ³ 00B3 | “ 201C | µ 00B5 | ¶ 00B6 | · 00B7 | ø 00F8 | ¹ 00B9 | ı 0157 | » 00BB | ¼ 00BC | ½ 00BD | ¾ 00BE | æ 00E6 |
| | C- | Ą 0104 | Į 012E | Ą 0100 | Ć 0106 | Ä 00C4 | Å 00C5 | Ę 0118 | Ě 0112 | Č 010C | É 00C9 | Ž 0179 | Ě 0116 | Ğ 0122 | Ķ 0136 | Ī 012A | Ļ 013B |
| | D- | Š 0160 | Ň 0143 | Ň 0145 | Ó 00D3 | Ō 014C | Ŏ 00D5 | Ö 00D6 | × 00D7 | Ů 0172 | Ł 0141 | Ś 015A | Ū 016A | Ü 00DC | Ž 017B | Ž 017D | ß 00DF |
| | E- | ą 0105 | į 012F | ā 0101 | ć 0107 | ä 00E4 | å 00E5 | ę 0119 | ě 0113 | č 010D | é 00E9 | ž 017A | ê 0117 | ğ 0123 | ķ 0137 | ī 012B | ļ 013C |
| | F- | š 0161 | ň 0144 | ņ 0146 | ó 00F3 | ō 014D | ö 00F5 | ÷ 00F6 | ů 00F7 | ł 0173 | ś 0142 | ŭ 015B | ū 016B | ü 00FC | ž 017C | ž 017E | ’ 2019 |
| | | 0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |

NOTE: This is compatible with ISO/IEC 8859-13 [28].

Figure A.9: Character code table 09 - Latin/Baltic Rim alphabet with Unicode equivalents

| | | Second Nibble | | | | | | | | | | | | | | | |
|---|----|---------------|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-------------|-----------|-----------|
| | | -0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |
| F i r s t N i b b l e | 0- | | | | | | | | | | | | | | | | |
| | 1- | | | | | | | | | | | | | | | | |
| | 2- | SPC 0020 | ! 0021 | " 0022 | # 0023 | \$ 0024 | % 0025 | & 0026 | ' 0027 | (0028 |) 0029 | * 002A | + 002B | , 002C | - 002D | . 002E | / 002F |
| | 3- | 0 0030 | 1 0031 | 2 0032 | 3 0033 | 4 0034 | 5 0035 | 6 0036 | 7 0037 | 8 0038 | 9 0039 | : 003A | ; 003B | < 003C | = 003D | > 003E | ? 003F |
| | 4- | @ 0040 | A 0041 | B 0042 | C 0043 | D 0044 | E 0045 | F 0046 | G 0047 | H 0048 | I 0049 | J 004A | K 004B | L 004C | M 004D | N 004E | O 004F |
| | 5- | P 0050 | Q 0051 | R 0052 | S 0053 | T 0054 | U 0055 | V 0056 | W 0057 | X 0058 | Y 0059 | Z 005A | [005B | \ 005C |] 005D | ^ 005E | _ 005F |
| | 6- | ` 0060 | a 0061 | b 0062 | c 0063 | d 0064 | e 0065 | f 0066 | g 0067 | h 0068 | i 0069 | j 006A | k 006B | l 006C | m 006D | n 006E | o 006F |
| | 7- | p 0070 | q 0071 | r 0072 | s 0073 | t 0074 | u 0075 | v 0076 | w 0077 | x 0078 | y 0079 | z 007A | { 007B | 007C | } 007D | ~ 007E | |
| | 8- | | | | | | | | | | | | | | | | |
| | 9- | | | | | | | | | | | | | | | | |
| | A- | NBSP 00A0 | Â 1E02 | â 1E03 | £ 00A3 | Č 010A | č 010B | Ď 1E0A | š 00A7 | Ŵ 1E80 | © 00A9 | ŵ 1E82 | đ 1E0B | Ÿ 1EF2 | SHY 00AD | ® 00AE | Ÿ 0178 |
| | B- | Ě 1E1E | ĭ 1E1F | Ĝ 0120 | ĝ 0121 | Ĭ 1E40 | ĭ 1E41 | Ŧ 00B6 | Ĥ 1E56 | ŵ 1E81 | þ 1E57 | ŵ 1E83 | Š 1E60 | ŷ 1EF3 | Ŵ 1E84 | ŵ 1E85 | š 1E61 |
| | C- | À 00C0 | Á 00C1 | Â 00C2 | Ã 00C3 | Ä 00C4 | Å 00C5 | Æ 00C6 | Ç 00C7 | È 00C8 | É 00C9 | Ê 00CA | Ë 00CB | Ì 00CC | Í 00CD | Î 00CE | Ï 00CF |
| | D- | Ŵ 0174 | Ņ 00D1 | Ō 00D2 | Ó 00D3 | Ô 00D4 | Õ 00D5 | Ö 00D6 | Ŧ 1E6A | Ø 00D8 | Ù 00D9 | Ú 00DA | Û 00DB | Ü 00DC | Ý 00DD | Ŷ 0176 | ß 00DF |
| | E- | à 00E0 | á 00E1 | â 00E2 | ã 00E3 | ä 00E4 | å 00E5 | æ 00E6 | ç 00E7 | è 00E8 | é 00E9 | ê 00EA | ë 00EB | ì 00EC | í 00ED | î 00EE | ï 00EF |
| | F- | ŵ 0175 | ñ 00F1 | ò 00F2 | ó 00F3 | ô 00F4 | õ 00F5 | ö 00F6 | ŧ 1E6B | ø 00F8 | ù 00F9 | ú 00FA | û 00FB | ü 00FC | ý 00FD | ÿ 0177 | |
| | | 0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |

NOTE: This is compatible with ISO/IEC 8859-14 [29].

Figure A.10: Character code table 0A - Latin/Celtic alphabet with Unicode equivalents

| | | Second Nibble | | | | | | | | | | | | | | | |
|---|----|---------------|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-------------|-----------|-----------|
| | | -0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |
| F i r s t N i b b l e | 0- | | | | | | | | | | | | | | | | |
| | 1- | | | | | | | | | | | | | | | | |
| | 2- | SPC 0020 | ! 0021 | " 0022 | # 0023 | \$ 0024 | % 0025 | & 0026 | ' 0027 | (0028 |) 0029 | * 002A | + 002B | , 002C | - 002D | . 002E | / 002F |
| | 3- | 0 0030 | 1 0031 | 2 0032 | 3 0033 | 4 0034 | 5 0035 | 6 0036 | 7 0037 | 8 0038 | 9 0039 | : 003A | ; 003B | < 003C | = 003D | > 003E | ? 003F |
| | 4- | @ 0040 | A 0041 | B 0042 | C 0043 | D 0044 | E 0045 | F 0046 | G 0047 | H 0048 | I 0049 | J 004A | K 004B | L 004C | M 004D | N 004E | O 004F |
| | 5- | P 0050 | Q 0051 | R 0052 | S 0053 | T 0054 | U 0055 | V 0056 | W 0057 | X 0058 | Y 0059 | Z 005A | [005B | \ 005C |] 005D | ^ 005E | _ 005F |
| | 6- | ` 0060 | a 0061 | b 0062 | c 0063 | d 0064 | e 0065 | f 0066 | g 0067 | h 0068 | i 0069 | j 006A | k 006B | l 006C | m 006D | n 006E | o 006F |
| | 7- | p 0070 | q 0071 | r 0072 | s 0073 | t 0074 | u 0075 | v 0076 | w 0077 | x 0078 | y 0079 | z 007A | { 007B | 007C | } 007D | ~ 007E | |
| | 8- | | | | | | | | | | | | | | | | |
| | 9- | | | | | | | | | | | | | | | | |
| | A- | NBSP 00A0 | ı 00A1 | ¢ 00A2 | £ 00A3 | € 20AC | ¥ 00A5 | Š 0160 | § 00A7 | š 0161 | © 00A9 | ª 00AA | « 00AB | ¬ 00AC | SHY 00AD | ® 00AE | ˆ 00AF |
| | B- | ° 00B0 | ± 00B1 | ² 00B2 | ³ 00B3 | Ž 017D | µ 00B5 | ¶ 00B6 | · 00B7 | ž 017E | ¹ 00B9 | º 00BA | » 00BB | Œ 0152 | œ 0153 | Ÿ 0178 | ¿ 00BF |
| | C- | À 00C0 | Á 00C1 | Â 00C2 | Ã 00C3 | Ä 00C4 | Å 00C5 | Æ 00C6 | Ç 00C7 | È 00C8 | É 00C9 | Ê 00CA | Ë 00CB | Ì 00CC | Í 00CD | Î 00CE | Ï 00CF |
| | D- | Ð 00D0 | Ñ 00D1 | Ò 00D2 | Ó 00D3 | Ô 00D4 | Õ 00D5 | Ö 00D6 | × 00D7 | Ø 00D8 | Ù 00D9 | Ú 00DA | Û 00DB | Ü 00DC | Ý 00DD | Þ 00DE | ß 00DF |
| | E- | à 00E0 | á 00E1 | â 00E2 | ã 00E3 | ä 00E4 | å 00E5 | æ 00E6 | ç 00E7 | è 00E8 | é 00E9 | ê 00EA | ë 00EB | ì 00EC | í 00ED | î 00EE | ï 00EF |
| | F- | ð 00F0 | ñ 00F1 | ò 00F2 | ó 00F3 | ô 00F4 | õ 00F5 | ö 00F6 | ÷ 00F7 | ø 00F8 | ù 00F9 | ú 00FA | û 00FB | ü 00FC | ý 00FD | þ 00FE | ÿ 00FF |
| | | 0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -A | -B | -C | -D | -E | -F |

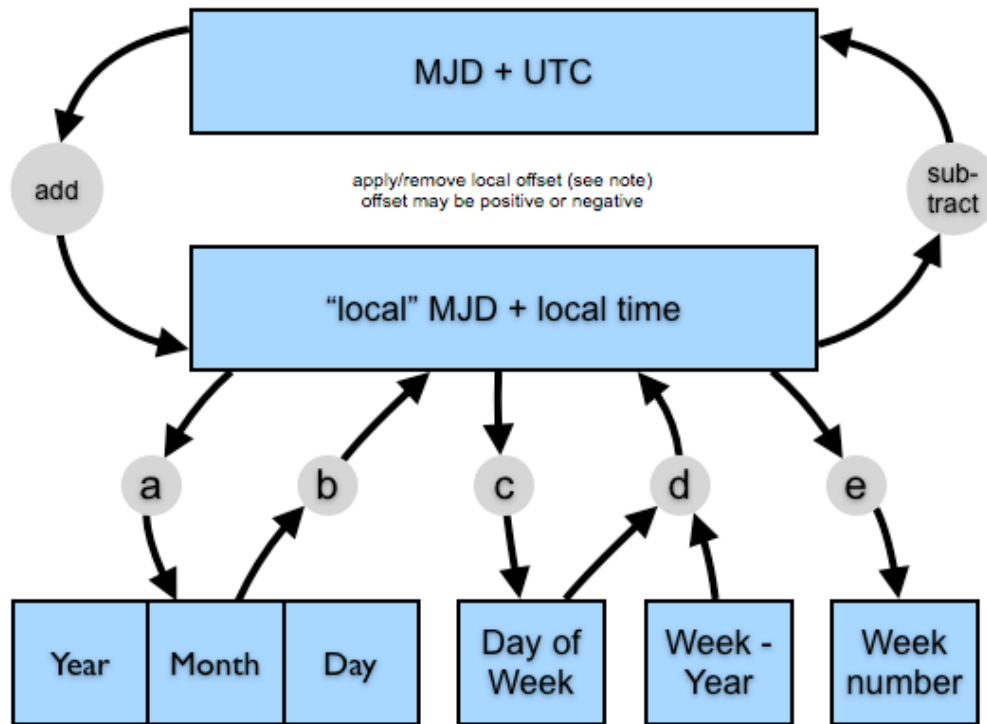
NOTE: This is compatible with ISO/IEC 8859-15 [30].

Figure A.11: Character code table 0B - Latin/Western European alphabet with Unicode equivalents

Annex B:
Void

Annex C (informative): Conversion between time and date conventions

The types of conversion which may be required are summarized in figure C.1.



NOTE: Offsets are positive for Longitudes East of Greenwich and negative for Longitudes West of Greenwich.

Figure C.1: Conversion routes between Modified Julian Date (MJD) and Co-ordinated Universal Time (UTC)

The conversion between MJD + UTC and the "local" MJD + local time is simply a matter of adding or subtracting the local offset. This process may, of course, involve a "carry" or "borrow" from the UTC affecting the MJD. The other five conversion routes shown on the diagram are detailed in the formulas below:

Symbols used:

| | |
|-----------------|---|
| D | Day of month from 1 to 31 |
| int | Integer part, ignoring remainder |
| K, L, M', W, Y' | Intermediate variables |
| M | Month from January (= 1) to December (= 12) |
| MJD | Modified Julian Date |
| mod 7 | Remainder (0-6) after dividing integer by 7 |
| UTC | Universal Time, Co-ordinated |
| WD | Day of week from Monday (= 1) to Sunday (= 7) |
| WN | Week number according to ISO 8601 [34] |
| WY | "Week number" Year from 1900 |
| x | Multiplication |
| Y | Year from 1900 (e.g. for 2003, Y = 103) |

- a) To find Y, M, D from MJD
- $$Y' = \text{int} [(\text{MJD} - 15\,078,2) / 365,25]$$
- $$M' = \text{int} \{ [\text{MJD} - 14\,956,1 - \text{int} (Y' \times 365,25)] / 30,6001 \}$$
- $$D = \text{MJD} - 14\,956 - \text{int} (Y' \times 365,25) - \text{int} (M' \times 30,6001)$$
- If $M' = 14$ or $M' = 15$, then $K = 1$; else $K = 0$
- $$Y = Y' + K$$
- $$M = M' - 1 - K \times 12$$

- b) To find MJD from Y, M, D
 If $M = 1$ or $M = 2$, then $L = 1$; else $L = 0$
 $MJD = 14\,956 + D + \text{int}[(Y - L) \times 365,25] + \text{int}[(M + 1 + L \times 12) \times 30,6001]$
- c) To find WD from MJD
 $WD = [(MJD + 2) \bmod 7] + 1$
- d) To find MJD from WY, WN, WD
 $MJD = 15\,012 + WD + 7 \times \{ WN + \text{int}[(WY \times 1\,461 / 28) + 0,41] \}$
- e) To find WY, WN from MJD
 $W = \text{int}[(MJD / 7) - 2\,144,64]$
 $WY = \text{int}[(W \times 28 / 1\,461) - 0,0079]$
 $WN = W - \text{int}[(WY \times 1\,461 / 28) + 0,41]$

EXAMPLE:

| | | | | | |
|-----|---|---------------|----|---|------------|
| MJD | = | 45 218 | W | = | 4 315 |
| Y | = | (19)82 | WY | = | (19)82 |
| M | = | 9 (September) | N | = | 36 |
| D | = | 6 | WD | = | 1 (Monday) |

NOTE: These formulas are applicable between the inclusive dates 1900 March 1 to 2100 February 28.

Annex D (normative):

Service information implementation of AC-3, Enhanced AC-3, and AC-4 audio in DVB systems

D.0 Introduction

This annex describes the implementation and implementation guidelines for DVB Service Information for conveying AC-3, Enhanced AC-3, and AC-4 audio elementary streams within a DVB transport stream.

The transmission of AC-3, Enhanced AC-3, and AC-4 audio streams is optional in DVB systems, as specified in TS 101 154 [9]. However, if AC-3, Enhanced AC-3, or AC-4 audio streams are transmitted in a DVB system the specifications in this annex shall be followed.

D.1 AC-3 and Enhanced AC-3 component types

Table D.1 shows the assignment of component_type values in the component_descriptor in the case that the stream_content value is set to 0x04, indicating the reference to an AC-3 or Enhanced AC-3 stream. If values of the component_type field other than those shown in table D.1 are conveyed, the component_type field may be ignored.

Table D.1: AC-3 and Enhanced AC-3 component_type value assignments

| component_type bits | Description |
|--|--|
| b ₇ (MSB) | Enhanced AC-3 flag (see table D.2) |
| b ₆ | Full service flag (see table D.3) |
| b ₅ to b ₃ | Service type flags (see table D.4) |
| b ₂ to b ₀ (see note) | Number of channels flags (see table D.5) |
| NOTE: This bit is transmitted last (see clause 5.1.6). | |

Table D.2: Enhanced AC-3 flag

| Enhanced AC-3 flag b ₇ | Description |
|--------------------------------------|-------------------------|
| 0 | Stream is AC-3 |
| 1 | Stream is Enhanced AC-3 |

Table D.3: Full service flag

| full service flag b ₆ | Description |
|-------------------------------------|---|
| 0 | Decoded audio stream is an associated service intended to be combined with another decoded audio stream before presentation to the listener |
| 1 | Decoded audio stream is a full service (suitable for decoding and presentation to the listener) |

Table D.4: Service type flags

| service type flags | | | Description | Restrictions (see note 1) | |
|--------------------|----------------|----------------|------------------------|-------------------------------------|--|
| b ₅ | b ₄ | b ₃ | | full service flag (b ₆) | number of channels flags (b ₂ to b ₀ , see note 2) |
| 0 | 0 | 0 | Complete Main (CM) | set to 1 | |
| 0 | 0 | 1 | Music and Effects (ME) | set to 0 | |
| 0 | 1 | 0 | Visually Impaired (VI) | | |
| 0 | 1 | 1 | Hearing Impaired (HI) | | |
| 1 | 0 | 0 | Dialogue (D) | set to 0 | |
| 1 | 0 | 1 | Commentary (C) | | set to 000 |
| 1 | 1 | 0 | Emergency (E) | set to 1 | set to 000 |
| 1 | 1 | 1 | Voiceover (VO) | set to 0 | set to 000 |
| 1 | 1 | 1 | Karaoke | set to 1 | set to 010, 011 or 100 |

NOTE 1: The values of the service type flags are only valid if the conditions identified in the restrictions column are satisfied.

NOTE 2: This bit is transmitted last (see clause 5.1.6).

Table D.5: Number of channels flags

| number of channels flags | | | Description | Restrictions (see note 1) Enhanced AC-3 flag (b ₇) |
|--------------------------|----------------|--------------------------------|--|---|
| b ₂ | b ₁ | b ₀ (see note 2) | | |
| 0 | 0 | 0 | Mono | |
| 0 | 0 | 1 | 1+1 Mode | |
| 0 | 1 | 0 | 2 channel (stereo) | |
| 0 | 1 | 1 | 2 channel Surround encoded (stereo) | |
| 1 | 0 | 0 | Multichannel audio (> 2 channels) | |
| 1 | 0 | 1 | Multichannel audio (> 5.1 channels) | set to 1 |
| 1 | 1 | 0 | Elementary stream contains multiple programmes carried in independent substreams | set to 1 |
| 1 | 1 | 1 | reserved for future use | |

NOTE 1: The values of the number of channels flags are only valid if the conditions identified in the restrictions column are satisfied.

NOTE 2: This bit is transmitted last (see clause 5.1.6).

D.2 AC-3 descriptor

The AC-3_descriptor identifies an AC-3 audio elementary stream that has been coded in accordance with clause 6.2 of TS 101 154 [9]. The intended purpose is to provide configuration information for the IRD.

The descriptor shall be included in a program map section at most once in each relevant ES_info descriptor loop which describes an elementary stream carrying AC-3 audio coded in accordance with T clause 6.2 of TS 101 154 [9].

The descriptor tag provides a unique identification of the presence of the AC-3 elementary stream. Other optional fields in the descriptor may be used to provide identification of the component type mode of the AC-3 audio coded in the stream (component_type field) and indicate if the stream is a main AC-3 audio service (main field) or an associated AC-3 service (asvc field).

The descriptor has a minimum length of one byte, but may be longer depending upon the state of the flags and the additional info loop.

D.3 AC-3 descriptor syntax and semantics

The AC-3 descriptor (see table D.6D.6) is used in the PSI PMT to identify streams which carry AC-3 audio. The descriptor shall be included in a program map section at most once in each relevant ES_info descriptor loop.

Table D.6: AC-3 descriptor syntax

| Syntax | Number of bits | Identifier |
|---------------------|----------------|------------|
| AC-3_descriptor() { | | |

| | | |
|---|---|--------|
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| component_type_flag | 1 | bslbf |
| bsid_flag | 1 | bslbf |
| mainid_flag | 1 | bslbf |
| asvc_flag | 1 | bslbf |
| reserved_flags | 4 | bslbf |
| if (component_type_flag == 1){ component_type } | 8 | uimsbf |
| if (bsid_flag == 1){ bsid } | 8 | uimsbf |
| if (mainid_flag == 1){ mainid } | 8 | uimsbf |
| if (asvc_flag == 1){ asvc } | 8 | uimsbf |
| for(i=0;i<N;i++){ additional_info_byte } | 8 | uimsbf |
| } | | |

Semantics for the AC-3 descriptor:

descriptor_tag: The descriptor tag is an 8-bit field which identifies each descriptor. The value assigned to the AC-3 descriptor_tag is 0x6A (see table 12).

descriptor_length: This 8-bit field specifies the total number of bytes of the data portion of the descriptor following the byte defining the value of this field. The AC-3 descriptor has a minimum length of one byte but may be longer depending on the use of the optional flags and the additional_info_loop.

component_type_flag: This 1-bit field is mandatory. It should be set to "1" to include the optional component_type field in the descriptor.

bsid_flag: This 1-bit field is mandatory. It should be set to "1" to include the optional bsid field in the descriptor.

mainid_flag: This 1-bit field is mandatory. It should be set to "1" to include the optional mainid field in the descriptor.

asvc_flag: This 1-bit field is mandatory. It should be set to "1" to include the optional asvc field in the descriptor.

reserved flags: These 1-bit fields are reserved for future use. They should always be set to "0".

component_type: This optional 8-bit field indicates the type of audio carried in the AC-3 elementary stream. It is set to the same value as the component type field of the component descriptor (see table D.2).

bsid: This optional 8-bit field indicates the AC-3 coding version. The three MSBs should always be set to "0". The five LSBs are set to the same value as the bsid field in the AC-3 elementary stream, i.e. "01000" (= 8) or "00110" (= 6) in the current version of AC-3.

mainid: This optional 8-bit field identifies a main audio service and contains a number in the range 0 to 7 which identifies a main audio service. Each main service should be tagged with a unique number. This value is used as an identifier to link associated services with particular main services.

asvc: This 8-bit field is optional. Each bit (0 to 7) identifies with which main services this associated service is associated. The left most bit, bit 7, indicates whether this associated service may be reproduced along with main service number 7. If the bit has a value of 1, the service is associated with main service number 7. If the bit has a value of 0, the service is not associated with main service number 7.

additional_info_byte: These optional bytes are reserved for future use.

D.4 Enhanced_AC-3 descriptor

The Enhanced_AC-3_descriptor identifies an Enhanced AC-3 audio elementary stream that has been coded in accordance with clause 6.2 of TS 101 154 [9]. The intended purpose is to provide configuration information for the IRD.

The descriptor shall be included in a program map section at most once in each relevant ES_info descriptor loop which describes an elementary stream carrying Enhanced AC-3 audio coded in accordance with clause 6.2 of TS 101 154 [9].

The descriptor tag provides a unique identification of the presence of the Enhanced AC-3 elementary stream. Other optional fields in the descriptor may be used to provide identification of the component type mode of the Enhanced AC-3 audio coded in the stream (component_type field) and indicate if the stream is a main Enhanced AC-3 audio service (mainid field) or an associated Enhanced AC-3 service (asvc field), or contains independent substreams (substream field).

The descriptor has a minimum length of one byte, but may be longer depending upon the state of the flags and the additional info loop.

D.5 Enhanced_AC-3 descriptor syntax and semantics

The Enhanced AC-3 descriptor (see table D.7) is used in the PSI PMT to identify streams which carry Enhanced AC-3 audio. The descriptor shall be included in a program map section at most once in each relevant ES_info descriptor loop.

Table D.7: Enhanced AC-3 descriptor syntax

| Syntax | Number of Bits | Identifier |
|--------------------------------|----------------|------------|
| enhanced_ac-3_descriptor(){ | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| component_type_flag | 1 | bslbf |
| bsid_flag | 1 | bslbf |
| mainid_flag | 1 | bslbf |
| asvc_flag | 1 | bslbf |
| mixinfoexists | 1 | bslbf |
| substream1_flag | 1 | bslbf |
| substream2_flag | 1 | bslbf |
| substream3_flag | 1 | bslbf |
| if (component_type_flag == 1){ | 8 | uimsbf |
| component_type | | |
| } | | |
| if (bsid_flag == 1){ | 8 | uimsbf |
| bsid | | |
| } | | |
| if (mainid_flag == 1){ | 8 | uimsbf |
| mainid | | |
| } | | |
| if (asvc_flag == 1){ | 8 | bslbf |
| asvc | | |
| } | | |
| if (substream1_flag == 1){ | 8 | uimsbf |
| substream1 | | |
| } | | |
| if (substream2_flag == 1){ | 8 | uimsbf |
| substream2 | | |
| } | | |
| if (substream3_flag == 1){ | 8 | uimsbf |
| substream3 | | |
| } | | |
| for (i=0;i<N;i++){ | 8 | bslbf |
| additional_info_byte | | |
| } | | |
| } | | |

Semantics for the Enhanced AC-3 descriptor:

descriptor_tag: The descriptor tag is an 8-bit field which identifies each descriptor. The value assigned to the Enhanced AC-3 descriptor_tag is 0x7A (see table 12).

descriptor_length: This 8-bit field specifies the total number of bytes of the data portion of the descriptor following the byte defining the value of this field. The Enhanced AC-3 descriptor has a minimum length of one byte but may be longer depending on the use of the optional flags and the additional_info_loop.

component_type_flag: This 1-bit field is mandatory. It should be set to "1" to include the optional component_type field in the descriptor.

bsid_flag: This 1-bit field is mandatory. It should be set to "1" to include the optional bsid field in the descriptor.

mainid_flag: This 1-bit field is mandatory. It should be set to "1" to include the optional mainid field in the descriptor.

asvc_flag: This 1-bit field is mandatory. It should be set to "1" to include the optional asvc field in the descriptor.

mixinfoexists: This 1-bit field is mandatory for Enhanced AC-3 streams. If set to "1" the Enhanced AC-3 stream contains metadata in independent substream 0 to control mixing with another AC-3 or Enhanced AC-3 stream.

substream1_flag: This 1-bit field is mandatory. It should be set to "1" to include the optional substream1 field in the descriptor. This flag should be set to "1" when the Enhanced AC-3 stream contains an additional programme carried in independent substream 1.

substream2_flag: This 1-bit field is mandatory. It should be set to "1" to include the optional substream2 field in the descriptor. This flag should be set to "1" when the Enhanced AC-3 stream contains an additional programme carried in independent substream 2.

substream3_flag: This 1-bit field is mandatory. It should be set to "1" to include the optional substream3 field in the descriptor. This flag should be set to "1" when the Enhanced AC-3 stream contains an additional programme carried in independent substream 3.

component_type: This optional 8-bit field indicates the type of audio carried in independent substream 0 of the Enhanced AC-3 elementary stream. It is set to the same value as the component type field of the component descriptor (see table D.1).

bsid: This optional 8-bit field indicates the Enhanced AC-3 coding version. The three MSBs should always be set to "0". The five LSBs are set to the same value as the bsid field in the Enhanced AC-3 elementary stream, "10000" (= 16) in the current version of Enhanced AC-3.

mainid: This optional 8-bit field identifies a main audio service and contains a number in the range 0 to 7 which identifies a main audio service. Each main service should be tagged with a unique number. This value is used as an identifier to link associated services with particular main services.

asvc: This 8-bit field is optional. Each bit (0 to 7) identifies with which main services this associated service is associated. The left most bit, bit 7, indicates whether this associated service may be reproduced along with main service number 7. If the bit has a value of 1, the service is associated with main service number 7. If the bit has a value of 0, the service is not associated with main service number 7.

substream1: This optional 8-bit field indicates the type of audio carried in independent substream 1 of the Enhanced AC-3 elementary stream. The value assignments of each bit are indicated in table D.8.

substream2: This optional 8-bit field indicates the type of audio carried in independent substream 2 of the Enhanced AC-3 elementary stream. The value assignments of each bit are indicated in table D.8.

substream3: This optional 8-bit field indicates the type of audio carried in independent substream 3 of the Enhanced AC-3 elementary stream. The value assignments of each bit are indicated in table D.8.

additional_info_byte: These optional bytes are reserved for future use.

Table D.8: Substream1-3 field bit value assignments

| substream1-3 bits | Description |
|--|---|
| b ₇ (MSB) | Mixing metadata flag (see table D.9) |
| b ₆ | Full service flag (see table D.3) |
| b ₅ to b ₃ | Service type flags (see table D.4) |
| b ₂ to b ₀ (see note) | Number of channels flags (see table D.10) |
| NOTE: This bit is transmitted last (see clause 5.1.6). | |

If values of the substream1, substream2 or substream3 fields other than those shown in table D.8 are conveyed, the respective substream field may be ignored.

Table D.9: Mixing metadata flag

| mixing metadata flag b ₇ | Description |
|--|---|
| 0 | No mixing metadata present in substream |
| 1 | Mixing metadata present in substream |

Table D.10: Number of channels flags

| number of channels flags | | | Description |
|--|----------------|---------------------------|-------------------------------------|
| b ₂ | b ₁ | b ₀ (see note) | |
| 0 | 0 | 0 | Mono |
| 0 | 0 | 1 | 1+1 Mode |
| 0 | 1 | 0 | 2 channel (stereo) |
| 0 | 1 | 1 | 2 channel Surround encoded (stereo) |
| 1 | 0 | 0 | Multichannel audio (> 2 channels) |
| 1 | 0 | 1 | Multichannel audio (> 5.1 channels) |
| 1 | 1 | 0 | reserved for future use |
| 1 | 1 | 1 | reserved for future use |
| NOTE: This bit is transmitted last (see clause 5.1.6). | | | |

D.6 AC-4 descriptor

The AC-4 descriptor identifies an AC-4 audio elementary stream that has been coded in accordance with clauses 6.6 or 6.7 of TS 101 154 [9]. The intended purpose is to provide configuration information for the IRD.

The descriptor shall be included in a program map section at most once in each relevant ES_info descriptor loop which describes an elementary stream carrying AC-4 audio, coded in accordance with clauses 6.6 or 6.7 of TS 101 154 [9].

The inclusion of the AC-4 TOC is optional.

D.7 AC-4 descriptor syntax and semantics

The AC-4 descriptor (see table D.11) uses the extension descriptor syntax according to clause 6.2.16. The combination of descriptor_tag and descriptor_tag_extension provides a unique identification of the presence of an AC-4 elementary stream.

The AC 4 descriptor is used in the PSI PMT to identify streams which carry AC-4 audio. The descriptor shall be included in a program map section at most once in each relevant ES_info descriptor loop.

The minimum length of the AC-4 descriptor, as given by the descriptor_length field, is 2 bytes.

However, the descriptor may be longer depending upon the inclusion of optional fields, such as channel mode, the dialog enhancement flag, the AC-4 TOC and the additional info loop.

Table D.11: AC-4 descriptor syntax

| Syntax | Number of bits | Identifier |
|---|----------------|------------|
| <code>ac-4_descriptor() {</code> | | |
| <code>descriptor_tag</code> | 8 | uimsbf |
| <code>descriptor_length</code> | 8 | uimsbf |
| <code>descriptor_tag_extension</code> | 8 | uimsbf |
| <code>ac4_config_flag</code> | 1 | uimsbf |
| <code>ac4_toc_flag</code> | 1 | uimsbf |
| <code>reserved_zero_future_use</code> | 6 | bslbf |
| <code>if (ac4_config_flag == 1) {</code> | | |
| <code>ac4_dialog_enhancement_enabled</code> | 1 | uimsbf |
| <code>ac4_channel_mode</code> | 2 | uimsbf |
| <code>reserved_zero_future_use</code> | 5 | bslbf |
| <code>}</code> | | |
| <code>if (ac4_toc_flag == 1) {</code> | | |
| <code>ac4_toc_len</code> | 8 | uimsbf |
| <code>for (i=0;i<N;i++) {</code> | | |
| <code>ac4_dsi_byte</code> | 8 | bslbf |
| <code>}</code> | | |
| <code>}</code> | | |
| <code>for (i=0;i<N;i++) {</code> | | |
| <code>additional_info_byte</code> | 8 | bslbf |
| <code>}</code> | | |
| <code>}</code> | | |

Semantics for the AC-4 descriptor:

ac4_config_flag: This 1-bit field indicates the presence of the `ac4_channel_mode` and the `ac4_dialog_enhancement_enabled` fields. This field shall be set to one if these optional fields are included in the descriptor.

ac4_toc_flag: This 1-bit field indicates the presence of the AC-4 TOC. This field shall be set to one if the optional AC-4 TOC is included in the descriptor.

ac4_dialog_enhancement_enabled: This 1-bit field shall be set to one if dialog enhancement information is carried in the bitstream, and to zero otherwise. The `ac4_dialog_enhancement_enabled` field shall only be present if the `ac4_config_flag` is set to one.

ac4_channel_mode: This field indicates the channel mode and shall be coded according to table D.12. The `ac4_channel_mode` field shall only be present if the `ac4_config_flag` is set to one.

Table D.12: ac4_channel_mode coding

| ac4_channel_mode | Description |
|------------------|-------------------------|
| 0 | Mono content |
| 1 | Stereo content |
| 2 | Multichannel content |
| 3 | Reserved for future use |

ac4_toc_len: This 8-bit field shall only be present if the `ac4_toc_flag` is set to 1. It shall indicate the length of the sequence of `ac4_dsi_byte` fields following it.

ac4_dsi_byte: This is an 8 bit field. The sequence of `ac4_dsi_byte` fields specifies the `ac4_dsi` field, which contains the AC-4 TOC (according to clause 6.6 of TS 101 154 [9]). The `ac4_dsi` field shall only be present if the `ac4_toc_len` is also present, and has a value greater than zero. The information contained in the `ac4_dsi` structure shall not be used to configure the AC-4 decoder. The AC-4 decoder shall obtain its configuration only from the AC-4 elementary stream.

additional_info_byte: These optional bytes are reserved for future use.

D.8 Use of the supplementary audio descriptor with AC-4

When AC-4 as specified in clause 6.6 of TS 101 154 [9] is used to carry a service containing supplementary audio, the `supplementary_audio_descriptor` identifies the purpose of the associated audio service (see annex J). In this case, the contents of the `supplementary_audio_descriptor` should match the AC-4 TOC carried in the AC-4 elementary stream of the associated audio service.

Annex E (normative): Usage of the Scrambling_descriptor

- Mixing of different scrambling modes within the same Transport Stream:

This situation may occur when a TS is made by multiplexing two or more independent TS streams.

- Mixing of different scrambling modes within the same service at the same time:

This is not allowed. The same mode shall be used by all scrambled components of a service at the same time.

- Change of scrambling mode over time for a given service (e.g. from event to event):

This situation may occur at any time, for instance when broadcasting events that were stored in scrambled mode or when inserting a local programme. Transitions should not be expected to be seamless.

Annex F (informative): ISO 639 Language Descriptor for "original audio" Soundtrack

Where audio in two languages is broadcast, TS 101 154 [9] specifies the order in which the ISO 639 Language Descriptors [15] will be used in order to associate a language with its respective audio channel. In some cases, broadcasters may wish to signal that one of the channels contains the audio in the original language of the source, without specifying the exact language being used. This "original audio" language may be signalled with an ISO 639 language code of "qaa", from the "local use" area of ISO 639 [36].

Annex G (normative):

Service information implementation of DTS® coded audio in DVB systems

G.0 Introduction

This annex describes the implementation and implementation guidelines for DVB Service Information for conveying DTS® and DTS-HD® coded audio elementary streams within a DVB transport stream.

The transmission of DTS® and DTS-HD® coded audio streams is optional in DVB systems, as specified in TS 101 154 [9]. However, if either DTS® or DTS-HD® coded audio streams are transmitted in a DVB system the specifications in this annex shall be followed.

G.1 DTS® and DTS-HD® descriptors

The DTS® descriptor identifies a DTS® coded audio elementary stream that has been coded in accordance with clause 6.3 of TS 101 154 [9] for a core substream. If an extension substream is present, as described in clause 6.3 of TS 101 154 [9], then the DTS-HD® descriptor is used to identify the coded audio stream. In systems supporting the DTS-HD® descriptor, use of the DTS® descriptor is deprecated and the DTS-HD® descriptor is used to identify DTS® and DTS-HD® audio streams. The intended purpose is to provide configuration information for the IRD.

Either a DTS descriptor or a DTS-HD® descriptor shall be included in a program map section in each relevant ES_info descriptor loop which describes an elementary stream containing DTS® or DTS-HD® coded audio coded in accordance with clause 6.3 of TS 101 154 [9].

The descriptor tag provides a unique identification of the presence of either DTS® or DTS-HD® coded elementary streams. Other optional fields in the descriptor may be used to provide identification of the component type mode of the DTS® or DTS-HD® audio coded in the stream (component_type field).

G.2 DTS® Descriptor

G.2.0 Use of the DTS® descriptor

The DTS® descriptor (see table G.1) is used in the PSI PMT to identify streams which carry DTS® audio. The descriptor shall be included in a program map section at most once in each relevant ES_info descriptor loop. If a DTS® descriptor is present, a DTS-HD® descriptor shall not be present in the same descriptor loop.

G.2.1 Syntax and semantics for the DTS® descriptor

The DTS® descriptor is shown in table G.1. It is optional that the IRD decodes the DTS® descriptor.

Table G.1: DTS® Audio Descriptor

| Syntax | Number of Bits | Mnemonic |
|------------------------|----------------|----------|
| DTS_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| sample_rate_code | 4 | bslbf |
| bit_rate_code | 6 | bslbf |
| nblks | 7 | bslbf |
| fsize | 14 | uimsbf |
| surround_mode | 6 | bslbf |
| lfe_flag | 1 | uimsbf |
| extended_surround_flag | 2 | uimsbf |
| for (i=0; i<N; i++) { | | |
| additional_info_byte | 8 | bslbf |
| } | | |

| | | |
|---|--|--|
| } | | |
|---|--|--|

sample_rate_code: This 4-bit field is equivalent to SFREQ in DTS® Coherent Acoustics. Specification and details are listed in table G.2. While broadcasters may use only a subset of these the complete table is given for consistency with the DTS® Coherent Acoustics specification as defined in clause 6.3 of TS 101 154 [9].

Table G.2: Sample Rate Code

| sample_rate_code | Sample Rate |
|------------------|-------------|
| 0000 | Invalid |
| 0001 | 8 kHz |
| 0010 | 16 kHz |
| 0011 | 32 kHz |
| 0100 | 64 kHz |
| 0101 | 128 kHz |
| 0110 | 11,025 kHz |
| 0111 | 22,05 kHz |
| 1000 | 44,1 kHz |
| 1001 | 88,02 kHz |
| 1010 | 176,4 kHz |
| 1011 | 12,kHz |
| 1100 | 24 kHz |
| 1101 | 48 kHz |
| 1110 | 96 kHz |
| 1111 | 192 kHz |

bit_rate_code: The specification and details of typical broadcast bit_rate_code are listed in table G.3. While broadcasters may use only a subset of these, the complete table of fixed transmission bit rate values is given for consistency with the DTS® Coherent Acoustics specification as defined in clause 6.3 of TS 101 154 [9].

NOTE: It is recommended that DTS® 5.1 compressed audio streams be transmitted at data rate of 384 kbps or above.

Table G.3: Bit Rate Table

| bit_rate_code | Transmission bit rate |
|---|-----------------------|
| x00101 | 128 kbps |
| x00110 | 192 kbps |
| x00111 | 224 kbps |
| x01000 | 256 kbps |
| x01001 | 320 kbps |
| x01010 | 384 kbps |
| x01011 | 448 kbps |
| x01100 | 512 kbps |
| x01101 | 576 kbps |
| x01110 | 640 kbps |
| x01111 | 768 kbps |
| x10000 | 960 kbps |
| x10001 | 1 024 kbps |
| x10010 | 1 152 kbps |
| x10011 | 1 280 kbps |
| x10100 | 1 344 kbps |
| x10101 | 1 408 kbps |
| x10110 | 1 411,2 kbps |
| x10111 | 1 472 kbps |
| x11000 | 1 536 kbps |
| x11001 | 1 920 kbps |
| x11010 | 2 048 kbps |
| x11011 | 3 072 kbps |
| x11100 | 3 840 kbps |
| x11101 | open |
| x11110 | variable |
| x11111 | lossless |
| NOTE: "x" indicates that the bit is reserved and should be ignored. | |

nblks: This 7-bit word is equivalent to the NBLKS parameter. This equals the number of PCM Sample Blocks. It indicates that there are (NBLKS+1) blocks (a block = 32 PCM samples per channel, corresponding to the number of PCM samples that are fed to the filterbank to generate one subband sample for each subband) in the current frame. The actual encoding window size is $32 \times (\text{NBLKS} + 1)$ PCM samples per channel. Valid range: 5 to 127. Invalid range: 0 to 4. For normal frames, this indicates a window size of either 2 048, 1 024, or 512 samples per channel. For termination frames, NBLKS can take any value in its valid range.

Fsize: This 14-bit word is equivalent to the FSIZE parameter. (FSIZE + 1) is the byte size of the current primary audio frame. The valid range for fsize is 95 to 8 192. The invalid range for fsize is 0 to 94, 8 193 to 16 384.

surround_mode: This 6-bit word is equivalent to the AMODE parameter. The values for surround_mode are given in table G.4.

Table G.4: Surround Mode

| surround_mode | Number of Channels/Channel Layout |
|--|------------------------------------|
| 000000 | 1 / mono |
| 000010 | 2 / L + R (stereo) |
| 000011 | 2 / (L+R) + (L-R) (sum-difference) |
| 000100 | 2 / LT +RT (left and right total) |
| 000101 | 3 / C + L + R |
| 000110 | 3 / L + R+ S |
| 000111 | 4 / C + L + R+ S |
| 001000 | 4 / L + R+ SL+SR |
| 001001 | 5 / C + L + R+ SL+SR |
| 001010 | User defined |
| 001011 | User defined |
| 001100 | User defined |
| 001101 | User defined |
| 001110 | User defined |
| 001111 | User defined |
| 010000 to 111111 | User defined |
| NOTE: L =left, R = right, C =centre, SL = surround left, SR = surround right, T = total. | |

lfe_flag: The lfe flag is set to 0 when the LFE (Low Frequency Effects) audio channel is OFF. The flag is set to 1 when the LFE audio channel is ON.

extended_surround_flag: The extended_surround_flag indicates the presence of DTS® ES rear centre audio as defined in clause 6.3 of TS 101 154 [9]. Its values are given in table G.5.

Table G.5: extended_surround_flag values

| extended_surround_flag | Description |
|------------------------|----------------------------|
| 00 | No Extended Surround |
| 01 | Matrixed Extended Surround |
| 10 | Discrete Extended Surround |
| 11 | Undefined |

additional_info_byte: This is an 8-bit field. The sequence of additional_info_byte fields specifies the additional_info field. It shall be coded according to clause 6.3 of TS 101 154 [9].

G.3 DTS-HD® descriptor

G.3.1 DTS-HD® descriptor syntax

Only one instance of the DTS-HD® descriptor shall be included in each relevant ES_info descriptor loop in the TS_program_map_section. If a DTS-HD® descriptor is present, a DTS® descriptor shall not be present in the same descriptor loop.

Note that when both core and extension substreams are present, the core substream descriptor values are valid for the core as an independent asset in the event that an IRD can only decode the core substream.

The DTS-HD® descriptor is shown in table G.6. It is optional that the IRD decodes the DTS-HD® descriptor.

Table G.6: DTS-HD® audio stream descriptor

| Syntax | Number of bits | Identifier |
|---------------------------------|----------------|------------|
| DTS-HD_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| substream_core_flag | 1 | bslbf |
| substream_0_flag | 1 | bslbf |
| substream_1_flag | 1 | bslbf |
| substream_2_flag | 1 | bslbf |
| substream_3_flag | 1 | bslbf |
| reserved_future_use | 3 | bslbf |
| if (substream_core_flag == 1) { | | |
| substream_info() | | |
| } | | |
| if (substream_0_flag == 1) { | | |
| substream_info() | | |
| } | | |
| if (substream_1_flag == 1) { | | |
| substream_info() | | |
| } | | |
| if (substream_2_flag == 1) { | | |
| substream_info() | | |
| } | | |
| if (substream_3_flag == 1) { | | |
| substream_info() | | |
| } | | |
| for (i=0; i<N; i++) { | | |
| additional_info_byte | 8 | bslbf |
| } | | |
| } | | |

Semantics for the DTS-HD® descriptor:

substream_core_flag: This 1-bit field is set to 1 if a core substream is present.

substream_0_flag: This 1 field is set to 1 if an extension substream with nuExtSSIndex = 0 exists in the audio stream.

substream_1_flag: This 1-bit field is set to 1 if an independent extension substream with nuExtSSIndex = 1 exists in the audio stream.

substream_2_flag: This 1-bit field is set to 1 if an independent extension substream with nuExtSSIndex = 2 exists in the audio stream.

substream_3_flag: This 1-bit field is set to 1 if an independent extension substream with nuExtSSIndex = 3 exists in the audio stream.

substream_info(): This field is coded according to clause G.3.2.

additional_info_byte: This 8-bit field is optional and reserves a place for future expansion of this descriptor.

G.3.2 Substream information

The substream_info field is described in table G.7.

Table G.7: Substream information

| Syntax | Number of bits | Identifier |
|-----------------------|----------------|------------|
| substream_info() { | | |
| substream_length | 8 | uimbsf |
| num_assets | 3 | uimbsf |
| channel_count | 5 | uimbsf |
| LFE_flag | 1 | bslbf |
| sampling_frequency | 4 | uimbsf |
| sample_resolution | 1 | bslbf |
| reserved_future_use | 2 | bslbf |
| for (i=0; i<N; i++) { | | |
| asset_info() | | |
| } | | |
| } | | |

Semantics for the substream_info:

substream_length: This 8-bit field specifies the total number of bytes following the substream_length field in the substream structure, including the embedded asset structures.

num_assets: This 3-bit field represents the number of audio assets stored in the substream. The number of audio assets stored in the substream is equal to num_assets+1. For a core substream, num_assets is always 0. For all independent extension substreams, this value is identical to nuNumAssets in the extension substream header.

channel_count: This 5-bit field represents the maximum number of output channels including LFE (if present). Note that channel_count represents the maximum number of channels after all relevant assets are mixed together in multi-asset presentations and may be less than or equal to the number of channels of all assets combined.

LFE_flag: This is a 1-bit field that is set to 1 if this substream contains an LFE channel.

sampling_frequency: This 4-bit field represents the audio sampling frequency. Use table G.8 to determine the maximum sampling frequency stored in the elementary stream. Note that not all values in the table are valid for the substream_core parameter.

Table G.8: sampling_frequency

| sampling_frequency | Sampling Frequency in kHz |
|--|---------------------------|
| 0 | 8 |
| 1 | 16 |
| 2 | 32 |
| 3 | 64 |
| 4 (see note) | 128 |
| 5 | 22,05 |
| 6 | 44,1 |
| 7 | 88,2 |
| 8 (see note) | 176,4 |
| 9 (see note) | 352,8 |
| 10 | 12 |
| 11 | 24 |
| 12 | 48 |
| 13 | 96 |
| 14 (see note) | 192 |
| 15 (see note) | 348 |
| NOTE: The indicated sampling frequencies are not to be used with a core substream. | |

sample_resolution: This 1-bit field is set to 1 if the decoded audio sample resolution is more than 16-bits. Otherwise, this field is set to 0.

asset_info(): This field is coded according to clause G.3.3. It appears as many times as indicated by the num_assets field.

G.3.3 Asset information

The `asset_info()` field is described in table G.9.

Table G.9: Asset information

| Syntax | Number of bits | Identifier |
|--|----------------|------------|
| <code>asset_info() {</code> | | |
| <code>asset_construction</code> | 5 | uimsbf |
| <code>vbr_flag</code> | 1 | bslbf |
| <code>post_encode_br_scaling_flag</code> | 1 | bslbf |
| <code>component_type_flag</code> | 1 | bslbf |
| <code>language_code_flag</code> | 1 | bslbf |
| <code>if (post_encode_br_scaling_flag == 1) {</code> | | |
| <code>bit_rate_scaled</code> | 13 | bslbf |
| <code>}</code> | | |
| <code>else {</code> | | |
| <code>bit_rate</code> | 13 | uimsbf |
| <code>}</code> | | |
| <code>reserved_future_use</code> | 2 | bslbf |
| <code>if (component_type_flag == 1) {</code> | | |
| <code>component_type</code> | 8 | bslbf |
| <code>}</code> | | |
| <code>if (language_code_flag == 1) {</code> | | |
| <code>ISO_639_language_code</code> | 24 | bslbf |
| <code>}</code> | | |
| <code>}</code> | | |

Semantics for the `asset_info()`:

asset_construction: This 5-bit field provides details about the internal construction of the audio stream and is interpreted according to table G.10.

Table G.10: Asset construction

| asset_construction | Core substream | | | | asset_construction | Extensions substream | | | | | |
|--------------------|----------------|-----|-----|------|--------------------|----------------------|------|-----|-----|-----|-----|
| | Core | XCH | X96 | XXCH | | Core | XXCH | X96 | XBR | XLL | LBR |
| 1 | X | | | | 1 | | | | | | |
| 2 | X | X | | | 2 | | | | | | |
| 3 | X | | | X | 3 | | | | | | |
| 4 | X | | X | | 4 | | | | | | |
| 5 | X | | | | 5 | | X | | | | |
| 6 | X | | | | 6 | | | | X | | |
| 7 | X | X | | | 7 | | | | X | | |
| 8 | X | | | X | 8 | | | | X | | |
| 9 | X | | | | 9 | | X | | X | | |
| 10 | X | | | | 10 | | | X | | | |
| 11 | X | X | | | 11 | | | X | | | |
| 12 | X | | | X | 12 | | | X | | | |
| 13 | X | | | | 13 | | X | X | | | |
| 14 | X | | | | 14 | | | | | X | |
| 15 | X | X | | | 15 | | | | | X | |
| 16 | X | | X | | 16 | | | | | X | |
| 17 | | | | | 17 | | | | | X | |
| 18 | | | | | 18 | | | | | | X |
| 19 | | | | | 19 | X | | | | | |
| 20 | | | | | 20 | X | X | | | | |
| 21 | | | | | 21 | X | | | | X | |

vbr_flag: This 1-bit field is set to 1 only when the audio asset has a variable bit rate.

post_encode_br_scaling_flag: This 1-bit field is set to 1 if scaling of the bit stream has occurred after it was encoded, otherwise this field is set to 0.

component_type_flag: This 1-bit field is set to 1 if the field component_type is present.

language_code_flag: This 1-bit field is set to 1 when ISO_639_language_code field is present.

bit_rate_scaled: This 13-bit field represents the scaled bit rate of the coded elementary stream as a 10,3 unsigned fractional fixed point value. This field exists in the descriptor when post_encode_br_scaling_flag = 1.

If the stream is variable bit rate, and dynamically updating the bit rate field is not possible or practical due to system limitations, or exceeds 8191 kbits/s, then bit_rate_scaled is set to 0.

bit_rate: This 13-bit field is an unsigned integer representing the bit rate of the coded elementary stream in kbits/s, ranging from 1 to 8191 kbits/s. This field exists in the descriptor when post_encode_br_scaling_flag = 0.

If the stream is variable bit rate, and dynamically updating the bit rate field is not possible or practical due to system limitations, then bit_rate is set to 0.

component_type: This field is coded according to clause G.3.4.

ISO_639_language_code: This 24-bit language code conforms to the ASCII language codes described in ISO/IEC 639 [36].

G.3.4 Component type

The component_type is an 8-bit field that specifies the type of the audio component as described in table G.11

Table G.11: Component type

| compopnent_type bits | Description |
|--|---|
| b ₇ (MSB) | reserved |
| b ₆ | full service flag (see table G.12) |
| b ₅ to b ₃ | service type flags (see table G.13) |
| b ₂ to b ₀ (see note) | number of channels flags (see table G.14) |
| NOTE: This bit is transmitted last (see clause 5.1.6). | |

Table G.12: Full service flag

| full service flag (b ₆) | Description |
|-------------------------------------|---|
| 0 | Decoded audio stream is intended to be combined with another decoded audio stream before presentation |
| 1 | Decoded audio stream is a full service (suitable for decoding and presentation to the listener) |

Table G.13: Service type flags

| service type flags | | | | Restrictions (see note 1) | |
|--|----------------|--------------------------------|------------------------|-------------------------------------|--|
| b ₂ | b ₁ | b ₀ (see note 2) | | full service flag (b ₆) | number of channels flags (b ₂ to b ₀ , see note 2) |
| 0 | 0 | 0 | Complete Main (CM) | always set to 1 | |
| 0 | 0 | 1 | Music and Effects (ME) | always set to 0 | |
| 0 | 1 | 0 | Visually Impaired (VI) | | |
| 0 | 1 | 1 | Hearing Impaired (HI) | | |
| 1 | 0 | 0 | Dialogue (D) | always set to 0 | |
| 1 | 0 | 1 | Commentary (C) | | always set to 000 |
| 1 | 1 | 0 | Emergency (E) | always set to 1 | always set to 000 |
| 1 | 1 | 1 | Voiceover (VO) | always set to 0 | always set to 000 |
| 1 | 1 | 1 | Reserved | always set to 1 | |
| NOTE 1: The values of the service type flags are only valid if the conditions identified in the restrictions column are satisfied. | | | | | |
| NOTE 2: This bit is transmitted last (see clause 5.1.6). | | | | | |

Table G.14: Number of channels flags

| number of channels flags | | | Description |
|--|----------------|------------------------------|---|
| b ₂ | b ₁ | b ₀ (see note) | |
| 0 | 0 | 0 | Mono |
| 0 | 0 | 1 | Reserved |
| 0 | 1 | 0 | 2 channel (stereo, LoRo) |
| 0 | 1 | 1 | 2 channel matrix encoded (stereo, LtRt) |
| 1 | 0 | 0 | Multichannel audio (>2 channels) |
| 1 | 0 | 1 | Reserved |
| 1 | 1 | 0 | Reserved |
| 1 | 1 | 1 | Reserved |
| NOTE: This bit is transmitted last (see clause 5.1.6). | | | |

G.4 Use of DTS-HD® in Receiver Mixed Applications for Single PID and Multiple PID Implementations

For applications where audio is being mixed in the receiver, DTS-HD® can accommodate two methods of asset delivery:

- Multiple PID delivery, where each PID is delivering a single asset.

For multiple PID delivery, the DTS-HD® stream contains a single asset and a component descriptor in the PMT loop. In this case, `component_type_flag` and `language_code_flag` may be set to 0, thus disabling the respective fields. If the respective flags are enabled, the corresponding values need to agree with the component descriptor in the PMT loop.

- Single PID delivery, where all assets are combined into a single multi-asset DTS-HD® audio stream.

For single PID delivery where multiple assets are delivered in a single DTS-HD® bitstream, multiple `asset_info()` components exist in the DTS-HD® descriptor. In this case, the `component_type` flag and `language_code_flag` are normally set, thus enabling the respective fields for each asset. The receiver can select which program combination is to be mixed and presented. The component descriptor in the PMT loop describes the default complete main presentation as configured by the broadcaster.

Annex H (normative): Service information implementation of AAC coded audio in DVB systems

H.0 Introduction

This annex describes the implementation and implementation guidelines for DVB Service Information for conveying AAC coded audio elementary streams within a DVB transport stream.

The transmission of AAC coded audio streams is optional in DVB systems, as specified in TS 101 154 [9]. However, if AAC coded audio streams are transmitted in a DVB system the specifications in this annex shall be followed.

H.1 AAC Audio descriptor

The AAC_descriptor identifies an AAC coded audio elementary stream that has been coded in accordance with clause 6.4 of TS 101 154 [9]. The intended purpose is to provide configuration information for the IRD.

The descriptor shall be included in a program map section at most once in each relevant ES_info descriptor loop which describes an elementary stream containing AAC coded audio coded in accordance with clause 6.4 of TS 101 154 [9].

The descriptor tag provides a unique identification of the presence of the AAC coded elementary stream. Other optional fields in the descriptor may be used to provide identification of the component type mode of the AAC audio coded in the stream (AAC_type field).

H.2 AAC descriptor

H.2.0 Use of the AAC descriptor

The AAC descriptor (see table H.1) is used in the PSI PMT to identify streams which carry AAC audio. The descriptor shall be included in a program map section at most once in each relevant ES_info descriptor loop.

H.2.1 Syntax and semantics for the AAC descriptor

The AAC_descriptor provides information about individual MPEG-4 AAC, MPEG-4 HE AAC and MPEG-4 HE AAC v2 elementary streams to be identified in the PSI PMT sections. The descriptor shall be included in a program map section at most once in each relevant ES_info descriptor loop which describes an elementary stream containing MPEG-4 AAC, MPEG-4 HE AAC or MPEG-4 HE AAC v2 audio.

Table H.1: AAC descriptor syntax

| Syntax | Number of Bits | Identifier |
|------------------------------|----------------|------------|
| AAC_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| profile_and_level | 8 | uimsbf |
| if (descriptor_length > 1) { | | |
| AAC_type_flag | 1 | bslbf |
| SAOC_DE_flag | 1 | bslbf |
| reserved_zero_future_use | 6 | bslbf |
| if (AAC_type_flag == 1) { | | |
| AAC_type | 8 | uimsbf |
| } | | |
| for (i=0; i<N; i++) { | | |
| additional_info_byte | 8 | uimsbf |
| } | | |
| } | | |
| } | | |

Semantics for the AAC descriptor:

profile_and_level: This 8-bit field specifies the Profile and Level used in MPEG-4 AAC, MPEG-4 HE AAC or MPEG-4 HE AAC v2. This field is set to the Profile and Level according to table 2-71 in ISO/IEC 13818-1 [15].

AAC_type_flag: This 1-bit field indicates the presence of the **AAC_type** field. This bit is set to "1" if the optional AAC_type field is included in the descriptor.

SAOC_DE_flag: This 1-bit flag indicates the presence of embedded SAOC-DE parametric data in AAC audio ancillary data and PES_private_data according to table H.2.

Table H.2: SAOC_DE_flag semantics

| SAOC_DE_flag | parametric data in AAC audio ancillary data | parametric data in PES_private_data (see note) |
|---|---|--|
| 0 | shall not be present | shall not be present |
| 1 | SAOC-DE parametric data shall be present | DE_control_data may be present |
| NOTE: PES_private_data within the PES packet header of the audio component as defined in TS 101 154 [9] clause E.7.2. | | |

reserved_future_use: This 6-bit field is reserved for future use and are all set to "0".

AAC_type: This field is set according to table 26 to the value of the component_type field when stream_content is 0x06. This field shall not be set to 0xA0.

additional_info_byte: This is an 8-bit field. The sequence of additional_info_byte fields specifies the additional_info field. The syntax and semantics of the additional_info field may be defined in future versions of the present document.

Annex I (normative): Assignment and interpretation of the service_type field

I.1 Background

The service_type field is present within both the service_descriptor and service_list_descriptor and is used to specify the type of a service. The intention of this field is to allow the service provider to describe the nature of the service, e.g. broadcast television, on-demand television, broadcast radio, data broadcast.

This information is deliberately provided at a very high-level within DVB Service Information (SI) to allow the receiver to make some decisions as soon as possible after the discovery of a service (through re-scan or some other mechanism) about how, and indeed whether, to present the service to the viewer for selection.

For example, information provided by the service_type field about the nature of a service can be used to group services into dedicated service lists for presentation to the viewer, e.g. separate television and radio lists.

The service_type field is not meant to override information provided at lower levels within SI or within PSI, such as the assignment of stream_type for a component within the PMT or the actual coding within the component itself, particularly with respect to the decoding and presentation of components of a service. It is provided because such low-level information may be difficult to interpret, slow to acquire, or (in the case of time-exclusive services) not even present at the time of service discovery.

NOTE: The receiver may also consider other factors, such as conditional access, when making such decisions. However, this is beyond the scope of the present document.

Broadcasters are recommended to use newly introduced service types when services are not intended to be listed on older receivers, and to use existing service types when services are intended to be listed on older receivers.

I.2 Assignment of service_type

I.2.0 General principles

For some services the assignment of a service_type from table 89 may be obvious, e.g. MPEG-2 HD digital television service. However, the decision is not always so straightforward.

Since the SDT is intended to be used by receivers to build service lists, the service_type shall be changed only when required due to operational changes (for example a reconfiguration of the corresponding multiplex or delivery system), to avoid unintended service list updates.

I.2.1 service_type "digital television service" (0x01)

In the generic case this service_type provides no explicit indication to the receiver about the way in which the components of a service have been encoded.

NOTE 1: Of course, in the case of a specific platform a particular encoding could be implicitly linked to this service_type and so inferred by the receiver. However, any such arrangement is beyond the scope of the present document.

As indicated by note 1 for table 89, this service_type should be used for MPEG-2 SD digital television service. However, it may also be used for services using other encodings, including encodings that have a specific entry elsewhere in table 89, e.g. MPEG-2 HD digital television service.

NOTE 2: DVB has deliberately not refined the definition of this service_type from "digital television service" to "MPEG-2 SD digital television service" due pre-existing use in the context of other (non-MPEG-2 SD) encodings.

On the assumption that all receivers will be able to decode and present MPEG-2 SD encoded material, all receivers will present any service assigned this service_type to the viewer for selection on the basis that it may be MPEG-2 SD coded material. However, as described above, this may not be the case and the receiver may not support the actual encoding used. This inability for the receiver to determine whether or not it will actually be able to decode and present a service assigned this service_type means that the service provider needs to allocate it with care depending on the viewer experience it wishes to achieve.

EXAMPLE: Consider a platform where some services are based on MPEG-2 SD encoding and others are based on MPEG-2 HD encoding, both of which are delivered to a mixed population of MPEG-2 SD-only and MPEG-2 SD/HD receivers.

For a service based on MPEG-2 SD encoding the assignment of service_type is obvious: 0x01 ("digital television service").

However, for a service based on MPEG-2 HD encoding the assignment of service_type depends on whether the service provider wants the service to be included in any service list presented to viewers of MPEG-2 SD-only receivers, even though they will not actually be able to view the service if selected. If this is the desired viewer experience then the service should be allocated service_type 0x01 ("digital television service"). If, however, the desired viewer experience is only to list services that the viewer of an MPEG-2 SD-only receiver will actually be able to view then the service should be allocated service_type 0x11 ("MPEG-2 HD digital television service").

This service_type may also be allocated to a service that contains both an MPEG-2 SD encoding and an alternative encoding (e.g. an MPEG-4 HD) of the same material. This is reasonable on the assumption that all receivers will be able to decode and present MPEG-2 SD encoded material, hence the viewer will at least be presented with the MPEG-2 SD coded form. However, depending on the capabilities of the receiver in use the viewer may be presented with the alternative, typically superior, coded form.

NOTE 3: The components used for the different encodings can be discriminated between at the point of decode by the assigned values for stream_type in PSI and/or use of the component_descriptor in SI.

1.2.2 service_type "H.264/AVC" (various)

In previous versions of this document the H.264/AVC service_types were called "advanced codec" and were allocated so as to be able to indicate that the video component of a service had been encoded using something other than MPEG-2. However, many IRD implementations had assumed such services would always contain an H.264/AVC video component. To avoid confusion the service types have been renamed to indicate H.264/AVC only.

The H.264/AVC service types shall hence only be used for services that provide an H.264/AVC video component. They may also be used with newer codecs, provided the service includes a video component which is backwards compatible with an H.264/AVC IRD.

1.2.3 service_type "H.264/AVC frame compatible stereoscopic HD" (various)

The H.264/AVC frame compatible stereoscopic HD values allow a broadcaster to signal that a service operates (primarily) as a frame compatible stereoscopic service. The use of these values requires careful consideration of the consequences for legacy receiver populations, which as a result may ignore these services. Therefore, a broadcaster may instead choose to signal a frame compatible stereoscopic service as a normal HD service, and use alternative signalling to indicate that the service (or event) is in a frame compatible stereoscopic format.

Further discussion of the signalling of frame compatible stereoscopic services is provided in TS 101 547-2 [46].

1.2.4 service_type "advanced codec digital radio sound service" (0x0A)

The advanced codec digital radio sound service_type has been allocated to indicate that an audio service has been encoded using something other than MPEG-1 Layer 2. The assignment of this service_type provides a generic indication of the use of some advanced audio codec without stating specifically which one. As such, on its own, it does not fully allow a receiver to determine that it is able to decode and present a service assigned this service_type.

NOTE: In the case of a specific platform a particular encoding may be implicitly linked to this service_type and so inferred by the receiver. However, any such arrangement is beyond the scope of the present document.

Where a service is assigned this service_type, the component_descriptor should be used to indicate the particular advanced codec used. This allows a receiver to unambiguously determine whether or not it will be able to decode and present the service and handle as appropriate.

1.2.5 service_type "HEVC digital television service" (0x1F)

1.2.5.0 General principles

The HEVC digital television service type has been allocated, and shall only be used for all the possible types of an HEVC video services that are decodable by HEVC HDTV IRD and HEVC UHDTV IRD as defined in TS 101 154 [9]. The service_type alone does not fully allow a receiver to determine that it is able to decode a service assigned the 0x1F service_type. Where a service is assigned this service_type, the component_descriptor shall hence be used to indicate the specific type of the HEVC service. This allows a receiver to unambiguously determine whether or not it will be able to decode and present the service and handle as appropriate.

One or more component descriptors shall be present in the SDT entry of each HEVC digital television service. There shall be one and only one component descriptor using values listed in table I.1, which is derived from table 26. Other component descriptors may also be present in the SDT, but no component descriptor using values from table I.9 shall be present.

Table I.1: HEVC component types from table 26 for service_type 0x1F

| stream_content | stream_content_ext | component_type | Description |
|---|--------------------|----------------|--|
| 0x9 | 0x0 | 0x00 | HEVC Main Profile high definition video, 50 Hz |
| | | 0x01 | HEVC Main 10 Profile high definition video, 50 Hz |
| | | 0x02 | HEVC Main Profile high definition video, 60 Hz |
| | | 0x03 | HEVC Main 10 Profile high definition video, 60 Hz |
| | | 0x04 | HEVC ultra high definition video (see note) conformant to one of the following UHDTV bit stream conformance points defined in table 18b in TS 101 154 [9]: <ul style="list-style-type: none">• SDR frame rate up to 60 Hz• SDR HFR dual PID with temporal scalability• HDR with HLG10 frame rate up to 60 Hz• HDR with HLG10 HFR dual PID and temporal scalability |
| NOTE: This value should be used for HLG10 HDR services, and/or HFR services with dual PID and temporal scalability as defined in TS 101 154 [9]. See also clause I.2.5.2. | | | |

Since component descriptors when used within the SDT can be used by receivers to build service lists, it is recommended that they only be modified occasionally. If broadcasters wish to indicate a different HEVC bit stream conformance point in use for a particular event, but which is also decodable by the receiver, it is recommended they signal this via informative component descriptors in the EIT. These SDT component descriptors should indicate the highest HEVC receiver profile that will be used on the service. Such a service may however also contain programmes encoded at a lower profile or resolution. Broadcasters should note that by signalling a particular profile, they may prevent receivers implementing a lower profile from acquiring that service. However, a broadcaster signalling a lower profile than the maximum profile actually used by the service, will result in programmes encoded with such a higher profile not being viewable on lower profile receivers.

The service_type value shall not be changed during a running service, i.e. a service with service_type 0x1F shall not be changed to service_type 0x20, to signal the carriage of content that contains any component with HDR or a frame rate above 60Hz.

I.2.5.1 Signalling for service frame compatible plano-stereoscopic 3DTV for HEVC coded services

Table I.4 shows the usage of the attributes specified in previous clauses for the SFC-3DTV service classes listed in clause 4.1 of TS 101 547-4 [53]. The table is not intended to represent an exhaustive list of possibilities nor specify the only signalling option for each form of 3DTV service. Rather, it is meant as guidance and for raising awareness of possible issues that need to be considered with the deployment of SFC-3DTV services.

The present document describes a single service_type for HEVC, and uses component descriptors to provide detailed differentiation between the service content. Thus, to signal SFC-3DTV services that are intended to be listed by legacy 2D receivers (i.e. backwards compatible signalling), a combination of a service_type indicating HEVC, and an SDT component descriptor indicating one of the four high definition HEVC component_type values as listed in table I.2 is used. Other component descriptors may also be present in the SDT, but no component descriptor using values from table I.5 or table I.9 shall be present.

Table I.2: HEVC component types from table 26 for SFC-3DTV

| stream_content | stream_content_ext | component_type | Description |
|----------------|--------------------|----------------|--|
| 0x9 | 0x0 | 0x00 | HEVC Main Profile high definition video 50 Hz |
| | | 0x01 | HEVC Main 10 Profile high definition video 50 Hz |
| | | 0x02 | HEVC Main Profile high definition video 60 Hz |
| | | 0x03 | HEVC Main 10 Profile high definition video 60 Hz |

To signal SFC-3DTV events that are intended to be listed by legacy 2D receivers (i.e. backwards compatible signalling), an additional component descriptor in the EIT is required to indicate the SFC-3DTV is plano-stereoscopic to SFC-3DTV cognisant receivers. This additional component descriptor shall be placed in the EIT. Only one additional component descriptor is currently defined for HEVC SFC-3DTV services and show in table I.3. Other values maybe defined later.

Table I.3: Currently defined HEVC plano-stereoscopic frame-packing component types from table 26

| stream_content | stream_content_ext | component_type | Description |
|----------------|--------------------|----------------|---|
| 0xB | 0xF | 0x03 | plano-stereoscopic top and bottom (TaB) frame-packing |

NOTE: Information about the current coding parameters of SFC-3DTV video is provided at the elementary stream level as specified in TS 101 547-4 [53] and TS 101 154 [9].

IRDs already existing in the field might not recognize the newly defined SFC-3DTV signalling, and this legacy may be supported through the video level signalling as described in clause 5.14.1.5.6 of TS 101 154 [9], in which case a 2D picture may be extracted from the 3D content by receivers supporting the HEVC default display window.

Table I.4: Summary of service compatible frame compatible plano-stereoscopic 3DTV service signalling scenarios

| Service Category | SDT | | EIT Component descriptors | Comment |
|---|--------------|-------------------------------|---|---|
| | service_type | Component descriptors | | |
| Occasional 3DTV service | 0x1F (HEVC) | One from table I.2 (see note) | For 3DTV events one SFC-3DTV related from table I.3 (see note); other component descriptors may be present, including multiple for the same elementary stream | This is the recommended signalling; 3DTV events are signalled via EIT |
| NOTE: SFC-3DTV signalling requires two component descriptors, one in the SDT, and one in the EIT that might be modified on an event basis. The component descriptor in the SDT indicates the type of HEVC video using stream_content set to 0x9, stream_content_ext set to 0x0, and the component_type set to a value in the range 0x00 to 0x03. The plano-stereoscopic top and bottom (TaB) frame-packing format is indicated by an additional component descriptor in the EIT with stream_content 0xB, stream_content_ext 0xF, and component_type 0x03. | | | | |

I.2.5.2 Signalling for HDR and/or frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz, but with a HEVC half frame rate temporal video sub-bitstream frame rate lower than or equal to 60 Hz

An Ultra high definition video service with a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz containing a half frame rate HEVC temporal video sub-bitstream and/or HLG10 HDR is decodable by HEVC_UHDTV_IRD as defined in TS 101 154 [9]. A component descriptor shall be present in the SDT entry of such a service using the values defined in table I.5. Other component descriptors may also be present in the SDT, but no component descriptor using values from table I.2 or table I.9 shall be present.

Table I.5: HEVC component types from table 26 for ultra high definition video using service_type 0x1F

| stream_content | stream_content_ext | component_type | Description |
|----------------|--------------------|----------------|--|
| 0x9 | 0x0 | 0x04 | HEVC ultra high definition video (notes 2, 9, and 10) conformant to one of the following UHDTV bit stream conformance points defined in table 18b in TS 101 154 [9]: <ul style="list-style-type: none"> • SDR frame rate up to 60 Hz • SDR HFR dual PID with temporal scalability • HDR with HLG10 frame rate up to 60 Hz • HDR with HLG10 HFR dual PID and temporal scalability |

Additional component descriptors may be present in the SDT and/or in the EIT to allow receiver to distinguish between services or events containing a HEVC UHDTV Bitstream (as defined in TS 101 154 [9]) and those which exceed those requirements but are still decodable by HEVC_UHDTV_IRD. These component descriptors are shown in table I.6. HEVC ultra high definition IRDs that are neither HDR nor capable of decoding a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz can ignore the component descriptor from table I.6 and decode the service as an SDR with a frame rate of up-to 60 Hz.

Table I.6: HEVC component types from table 26 for a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz and HDR using service_type 0x1F

| stream_content | stream_content_ext | component_type | Description |
|----------------|--------------------|----------------|--|
| 0xB | 0xF | 0x04 | HLG10 HDR |
| | | 0x05 | HEVC temporal video subset for a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz |

I.2.5.3 Spatial, temporal, and dynamic range characteristics

Additional component descriptors may also be present in the SDT and EIT to inform on the source spatial, temporal and dynamic range characteristics. These component descriptors (as shown in table I.7) are intended to be used in the EIT when the broadcaster wants to indicate the video characteristics for a particular event are less than the capabilities of the service indicated by the service_type and component descriptors in the SDT.

Table I.7: Component types from table 26 to inform on the source's spatial, temporal, and dynamic range characteristics

| stream_content | stream_content_ext | component_type | Description |
|----------------|--------------------|----------------|--|
| 0x3 | n/a | 0x40 | video spatial resolution has been upscaled from lower resolution source material |
| | | 0x41 | video is standard dynamic range (SDR) |
| | | 0x42 | video is high dynamic range (HDR) remapped from standard dynamic range (SDR) source material |
| | | 0x43 | video is high dynamic range (HDR) up-converted from standard dynamic range (SDR) source material |
| | | 0x44 | video is standard frame rate, less than or equal to 60 Hz |
| | | 0x45 | high frame rate video generated from lower frame rate source material |

I.2.5.4 Summary of signalling for different bitstream profiles using service_type 0x1F

Table I.8: Signalling for HEVC bitstream profiles

| HEVC profile | Mandatory SDT component descriptor | | | Optional SDT component descriptors (note 2) | | |
|--|------------------------------------|--------------------|----------------|---|--------------------|----------------|
| | stream_content | stream_content_ext | component_type | stream_content | stream_content_ext | component_type |
| HEVC Main Profile high definition video, 50 Hz | 0x9 | 0x0 | 0x00 | | | |
| HEVC Main 10 Profile high definition video, 50 Hz | 0x9 | 0x0 | 0x01 | | | |
| HEVC Main Profile high definition video, 60 Hz | 0x9 | 0x0 | 0x02 | | | |
| HEVC Main 10 Profile high definition video, 60 H | 0x9 | 0x0 | 0x03 | | | |
| HEVC ultra high definition video conformant to the conformance point "SDR frame rate up to 60 Hz" as is defined in table 18b in TS 101 154 [9] | 0x9 | 0x0 | 0x04 | | | |
| HEVC ultra high definition video conformant to the conformance point "SDR HFR dual PID and temporal scalability" as is defined in table 18b in TS 101 154 [9] | 0x9 | 0x0 | 0x04 | 0xB | 0xF | 0x05 |
| HEVC ultra high definition video conformant to the conformance point "HDR with HLG10 frame rate up to 60 Hz" as is defined in table 18b in TS 101 154 [9] | 0x9 | 0x0 | 0x04 | 0xB | 0xF | 0x04 |
| HEVC ultra high definition video conformant to the conformance point "HDR with HLG10 HFR dual PID and temporal scalability" as is defined in table 18b in TS 101 154 [9] (note 1) | 0x9 | 0x0 | 0x04 | 0xB | 0xF | 0x04 |
| | | | | 0xB | 0xF | 0x05 |
| NOTE 1: Both optional component descriptors are intended to be present. | | | | | | |
| NOTE 2: These component descriptors are optional to broadcast. A broadcaster can signal them if they wish for receivers to be able to group services by their HEVC bit stream profile. There are no requirements for receivers to use these. | | | | | | |

I.2.6 service_type "HEVC digital television service with HDR and/or a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz" (0x20)

I.2.6.1 General principles

In order to avoid compatibility issues with already deployed HEVC_HDTV_IRDs and HEVC_UHDTV_IRDs, a new service_type is defined for HEVC digital television service with HDR and/or a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz which cannot be decoded by HEVC_UHDTV_IRD as defined in TS 101 154 [9].

The "HEVC digital television with HDR and/or a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz" service type shall only be used for all the possible types of an HEVC video services with HDR and/or a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz that are not decodable by a HEVC_UHDTV_IRD (as defined in TS 101 154 [9]). The service_type on its own does not fully allow a receiver to determine that it is able to decode a service assigned this 0x20 service_type. Where a service is assigned this service_type, the component_descriptor shall hence be used to indicate the specific type of the HEVC digital television service with HDR and/or a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz. This allows a receiver to unambiguously determine whether or not it will be able to decode and present the service.

One or more component descriptors shall be present in the SDT entry of each HEVC digital television service with HDR and/or a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz. There shall be one and only one component descriptor using values from table I.9, a subset derived from table 26. Other component descriptors may also be present in the SDT (such as those defined in table I.7), but no component descriptor using values from table I.1 shall be present.

Table I.9: HEVC component types from table 26 for service_type 0x20

| stream_content | stream_content_ext | component_type | Description |
|-----------------------|---------------------------|-----------------------|--|
| 0x9 | 0x0 | 0x05 | <p>HEVC ultra high definition video with PQ10 HDR with a frame rate lower than or equal to 60 Hz conformant to one of the following UHDTV bit stream conformance points defined in table 18b in TS 101 154 [9]:</p> <ul style="list-style-type: none"> • HDR with PQ10 frame rate up to 60 Hz <p>or</p> <p>HEVC ultra high definition video with PQ10 HDR with a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz with a half frame rate HEVC temporal video sub-bit-stream conformant to one of the following UHDTV bit stream conformance points defined in table 18b in TS 101 154 [9]:</p> <ul style="list-style-type: none"> • HDR with PQ10 HFR dual PID and temporal scalability |
| | | 0x06 | <p>HEVC ultra high definition video, frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz without a half frame rate HEVC temporal video sub-bit-stream conformant to one of the following UHDTV bit stream conformance points defined in table 18b in TS 101 154 [9]:</p> <ul style="list-style-type: none"> • SDR HFR single PID • HDR with HLG10 HFR single PID |
| | | 0x07 | <p>HEVC ultra high definition video with PQ10 HDR, frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz without a half frame rate HEVC temporal video sub-bit-stream conformant to one of the following UHDTV bit stream conformance points defined in table 18b in TS 101 154 [9]:</p> <ul style="list-style-type: none"> • HDR with PQ10 HFR single PID |

Since component descriptors when used within the SDT can be used by receivers to build service lists, it is recommended that they only be modified occasionally. If broadcasters wish to indicate a different HEVC bit stream conformance point in use for a particular event, but which is also decodable by the receiver, it is recommended they signal this via informative component descriptors in the EIT.

Additional component descriptors may be present in the SDT and/or in the EIT to allow receiver to distinguish between services or events containing a half frame rate HEVC temporal video sub-bitstream or HLG10 HDR. These component descriptors are listed in table I.10.

Additional component descriptors may also be present in the SDT and EIT to inform on the source spatial, temporal and dynamic range characteristics, as described in I.2.5.3.

The service_type value shall not be changed during a running service, even in the case when a service with service_type 0x20 intermittently carries content that does not contain any component with HDR, nor a frame rate above 60Hz, whereas such content would be compatible with a service with service_type 0x1F.

Table I.10: HEVC component types from table 26 indicating the presence of a half frame rate HEVC temporal video sub-bitstream or HLG10 HDR using service_type 0x20

| stream_content | stream_content_ext | component_type | Description |
|----------------|--------------------|----------------|--|
| 0xB | 0xF | 0x04 | HLG10 HDR |
| | | 0x05 | HEVC temporal video subset for a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz |

I.2.6.1 Summary of signalling for different bitstream profiles using service_type 0x20

Table I.11: Signalling for HEVC bitstream profiles

| HEVC profile | service_type | Mandatory SDT component descriptor | | | Optional SDT component descriptors (note) | | |
|---|--------------|------------------------------------|--------------------|----------------|---|--------------------|----------------|
| | | stream_content | stream_content_ext | component_type | stream_content | stream_content_ext | component_type |
| HEVC ultra high definition video, frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz without a half frame rate HEVC temporal video sub-bit-stream conformant to the conformance point "SDR HFR single PID" as is defined in in table 18b in TS 101 154 [9] | 0x20 | 0x9 | 0x0 | 0x06 | | | |
| HEVC ultra high definition video, frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz without a half frame rate HEVC temporal video sub-bit-stream conformant to the conformance point "HDR with HLG10 HFR single PID" as is defined in in table 18b in TS 101 154 [9] | 0x20 | 0x9 | 0x0 | 0x06 | 0xB | 0xF | 0x04 |
| HEVC ultra high definition video with PQ10 HDR with a frame rate lower than or equal to 60 Hz conformant to the conformance point "HDR with PQ10 frame rate up to 60 Hz" as is defined in in table 18b in TS 101 154 [9] | 0x20 | 0x9 | 0x0 | 0x05 | | | |
| HEVC ultra high definition video with PQ10 HDR with a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz containing a half frame rate HEVC temporal video sub-bit-stream | 0x20 | 0x9 | 0x0 | 0x05 | 0xB | 0xF | 0x05 |
| HEVC ultra high definition video with PQ10 HDR, frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz without a half frame rate HEVC temporal video sub-bit-stream conformant to the conformance point "HDR with PQ10 HFR single PID" as is defined in in table 18b in TS 101 154 [9] | 0x20 | 0x9 | 0x0 | 0x07 | | | |
| NOTE: These component descriptors are optional to broadcast. A broadcaster can signal them if they wish for receivers to be able to group services by their HEVC bit stream profile. There are no requirements for receivers to use these. | | | | | | | |

I.2.7 Summary of signalling for HEVC bitstream profiles using service_type 0x1F or 0x20 (informative)

Table I.12 summarizes the different ultra high definition bitstream signalling for convenience.

Table I.12: Signalling for HEVC bitstream profiles

| HEVC profile | service_type | Mandatory SDT component descriptor | | | Optional SDT component descriptors (note 2) | | |
|---|--------------|------------------------------------|--------------------|----------------|---|--------------------|----------------|
| | | stream_content | stream_content_ext | component_type | stream_content | stream_content_ext | component_type |
| HEVC Main Profile high definition video, 50 Hz | 0x1F | 0x9 | 0x0 | 0x0 | | | |
| HEVC Main 10 Profile high definition video, 50 Hz | 0x1F | 0x9 | 0x0 | 0x1 | | | |
| HEVC Main Profile high definition video, 60 Hz | 0x1F | 0x9 | 0x0 | 0x2 | | | |
| HEVC Main 10 Profile high definition video, 60 Hz | 0x1F | 0x9 | 0x0 | 0x3 | | | |
| HEVC ultra high definition video conformant to the conformance point "SDR frame rate up to 60 Hz" as is defined in table 18b in TS 101 154 [9] | 0x1F | 0x9 | 0x0 | 0x4 | | | |
| HEVC ultra high definition video conformant to the conformance point "SDR HFR dual PID and temporal scalability" as is defined in table 18b in TS 101 154 [9] | 0x1F | 0x9 | 0x0 | 0x4 | 0xB | 0xF | 0x05 |
| HEVC ultra high definition video conformant to the conformance point "HDR with HLG10 frame rate up to 60 Hz" as is defined in table 18b in TS 101 154 [9] | 0x1F | 0x9 | 0x0 | 0x4 | 0xB | 0xF | 0x04 |
| HEVC ultra high definition video conformant to the conformance point "HDR with HLG10 HFR dual PID and temporal scalability" as is defined in table 18b in TS 101 154 [9] (note 1) | 0x1F | 0x9 | 0x0 | 0x4 | 0xB | 0xF | 0x04 |
| | | | | | 0xB | 0xF | 0x05 |
| HEVC ultra high definition video, frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz without a half frame rate HEVC temporal video sub-bit-stream conformant to the conformance point "SDR HFR single PID" as is defined in in table 18b in TS 101 154 [9] | 0x20 | 0x9 | 0x0 | 0x6 | | | |
| HEVC ultra high definition video, frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz without a half frame rate HEVC temporal video sub-bit-stream conformant to the conformance point "HDR with HLG10 HFR single PID" as is defined in in table 18b in TS 101 154 [9] | 0x20 | 0x9 | 0x0 | 0x6 | 0xB | 0xF | 0x04 |
| HEVC ultra high definition video with PQ10 HDR with a frame rate lower than or equal to 60 Hz conformant to the conformance point "HDR with PQ10 frame rate up to 60 Hz" as is defined in in table 18b in TS 101 154 [9] | 0x20 | 0x9 | 0x0 | 0x5 | | | |
| HEVC ultra high definition video with PQ10 HDR with a frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz containing a half frame rate HEVC temporal video sub-bit-stream | 0x20 | 0x9 | 0x0 | 0x5 | 0xB | 0xF | 0x05 |
| HEVC ultra high definition video with PQ10 HDR, frame rate of 100 Hz, 120 000/1 001 Hz, or 120 Hz without a half frame rate HEVC temporal video sub-bit-stream conformant to the conformance point "HDR with PQ10 HFR single PID" as is defined in in table 18b in TS 101 154 [9] | 0x20 | 0x9 | 0x0 | 0x7 | | | |
| NOTE 1: Both optional component descriptors are intended to be present. | | | | | | | |
| NOTE 2: These component descriptors are optional to broadcast. A broadcaster can signal them if they wish for receivers to be able to group services by their HEVC bit stream profile. There are no requirements for receivers to use these. | | | | | | | |

Annex J (normative): Signalling of Supplementary Audio

J.1 Overview

The supplementary audio descriptor provides additional information about audio streams to allow a receiver to present the audio to the user in the appropriate way. The supplementary audio descriptor can also be used to describe non-supplementary audio components.

The supplementary audio descriptor can override the language code in the ISO_639_language_descriptor [15] or other descriptors in the same ES loop of a PMT. This is commonly used where the language code in the ISO_639_language_descriptor is set to aid legacy receivers. In such cases the more accurate language code is conveyed in the supplementary audio descriptor.

For legacy receivers, the ISO_639_language_descriptor may also continue to be carried.

Three mechanisms are defined for the provision of supplementary audio (for example audio description):

- receiver-mix,
- broadcast-mix, and
- parametric data stream.

In the first two cases, the audio exists as two separate audio recordings, one recording is a complete and independent audio, the other is an additional audio recording, that will only be listened to by the viewer when mixed with the complete and independent audio. A broadcaster then broadcasts a complete and independent audio, and in addition one or both of the following two options.

- 1) the broadcaster transmits an extra audio component which is a mix of the complete and independent audio, and the additional audio recording. This is called broadcast-mix audio.
- 2) the broadcaster transmits the additional audio recording as an extra audio component (known as dependent audio) because it is intended to be mixed with a complete and independent audio stream by the receiver. This is called receiver-mix audio.

SAOC-DE parametric data streams are described further in clause J.5.

This annex defines the signalling in each case to enable the receiver to present the correct audio to the user.

J.2 Receiver-mix supplementary audio

J.2.1 Introduction

Receiver-mix audio allows receivers to give the listener control over the relative levels of the mix, to improve intelligibility. Where it is important to conserve bit rate, receiver-mix supplementary audio offers the broadcaster a way to provide a useful feature for users without having to broadcast another complete audio channel. However, it requires the receiver to implement two audio decoders and a mixer.

TS 101 154 [9], Annex E defines additional signalling which controls how the receiver mixes the complete and independent audio stream with the dependent audio stream before presentation to the user.

Receivers shall use attributes such as language_code and editorial_classification to determine the complete and independent audio stream to mix with a dependent audio stream.

Receiver mixing of audio streams shall apply the following restrictions, which depend on values signalled in the supplementary audio descriptor:

- A dependent stream (mix_type 0) shall only be mixed with a complete and independent stream (mix_type 1)

- A dependent stream with audio description for the visually impaired (editorial_classification 0x01) shall only be mixed with a complete and independent stream associated with the same language code. For other editorial classifications, the dependent and complete and independent streams may use different language codes.
- An audio stream with an editorial_classification in the range 0x00 to 0x16 (inclusive) shall not be mixed with another stream which has the same editorial classification as itself, regardless of whether it is either a dependent, or a complete and independent stream.

NOTE: It is, for example, hence not allowed to mix a broadcast-mix spoken subtitles stream with a dependent spoken subtitles stream.

Note that TS 101 154 [9] specifies that in the case of receiver-mix, the main audio and the dependent audio shall be encoded using the same codec family.

J.2.2 PSI PMT signalling

When a receiver-mix supplementary dependent stream is present, it shall be signalled in the ES descriptor loop of the PSI PMT with both an ISO_639_language_code in the ISO_639_language_descriptor [15] set to reflect the content of the audio stream and a supplementary_audio descriptor. The supplementary_audio descriptor shall set the mix_type to zero and the editorial_classification to any appropriate value other than 0. The other fields shall be set according to clause J.4.

A stream_identifier descriptor should be present.

J.2.3 EIT signalling

J.2.3.0 General principles

A component descriptor, with the same value of component_tag as that used in the PSI PMT stream identifier descriptor of the dependent audio stream, should be present in the descriptor loop in the EIT for each event for which supplementary audio is broadcast.

J.2.3.1 Visually impaired audio description

As defined in table 26, for visually impaired audio description the values of stream_content, stream_content_ext and component_type given in table J.1 shall be used.

Table J.1: Receiver-mix component descriptor values for visually impaired audio description

| Audio coding | stream_content | stream_content_ext | component_type |
|---|----------------|--------------------|---------------------------|
| MPEG-1 Layer 2 | 0x2 | 0xF | 0x47 |
| Enhanced AC-3 | 0x4 | 0xF | 0x90 (see note 1) |
| AC-4 | 0x9 | 0x1 | 0x0C to 0x0D (see note 4) |
| HE AAC, AAC (see note 3) | 0x6 | 0xF | 0x47 |
| HE AAC v2 | 0x6 | 0xF | 0x49 |
| DTS-HD® | 0x7 | 0xF | 0bx0010xxx (see note 2) |
| NOTE 1: As defined in clause D. | | | |
| NOTE 2: As defined in clause G.3.4. | | | |
| NOTE 3: The AAC profile (which includes low-complexity AAC) also uses this type (see clause 6.2.8) | | | |
| NOTE 4: The component_type value (defined in clause 6.2.8) is dependent upon the AC-4 coding parameters | | | |

J.3 Broadcast-mix supplementary audio

J.3.1 Introduction

Broadcast-mix supplementary audio may be used in broadcast systems where bit rate is available for a separate audio stream or where compatibility with legacy receivers is important. It requires no special handling in the audio decoder of

the receiver, however receivers should recognise that this audio stream is different from the main programme audio stream using the information in the ISO_639_language_descriptor and supplementary audio descriptors.

J.3.2 PSI PMT signalling

When a broadcast-mix supplementary audio stream is present, it shall be signalled in the PSI PMT with a supplementary_audio_descriptor. The ISO_639_language_code field in the ISO_639_language_descriptor [15] should be set to reflect the content of the audio stream, however this may be an issue with legacy receivers which do not interpret the supplementary_audio_descriptor. For this reason, some broadcasters may decide to use a different language code in the ISO_639_language_code field (e.g. "NAR" or "qad") in which case the correct language code shall be signalled in the supplementary_audio_descriptor. The other fields shall be set according to clause J.4.

A stream_identifier descriptor should be present.

J.3.3 EIT signalling

J.3.3.0 General principles

A component descriptor, with the same value of component_tag as that used in the PSI PMT stream identifier descriptor of the supplementary audio stream, should be present in the descriptor loop in the EIT for each event for which a supplementary audio stream is broadcast.

J.3.3.1 Visually impaired audio description

As defined in table 26, for visually impaired audio description the values of stream_content, stream_content_ext and component_type given in table J.2 shall be used.

Table J.2: Broadcast-mix component descriptor values for visually impaired audio description

| Audio coding | stream_content | stream_content_ext | component_type |
|---|----------------|--------------------|---------------------------|
| MPEG-1 Layer 2 | 0x2 | 0xF | 0x48 |
| AC-3 | 0x4 | 0xF | 0b01010xxx (see note 1) |
| Enhanced AC-3 | 0x4 | 0xF | 0b11010xxx (see note 1) |
| AC-4 | 0x9 | 0x1 | 0x06 to 0x0B (see note 4) |
| HE AAC, AAC (see note 3) | 0x6 | 0xF | 0x48 |
| HE AAC v2 | 0x6 | 0xF | 0x4A |
| DTS® | 0x7 | 0xF | 0bx1010xxx (see note 2) |
| DTS-HD® | 0x7 | 0xF | 0bx1010xxx (see note 2) |
| NOTE 1: 'xxx' is dependent upon the number of channels, as defined in clause D. | | | |
| NOTE 2: 'xxx' is dependent upon the number of channels, as defined in clause G.3.4. | | | |
| NOTE 3: The AAC profile (which includes low-complexity AAC) also uses this type (see clause 6.2.8) | | | |
| NOTE 4: The component_type value (defined in clause 6.2.8) is dependent upon the AC-4 coding parameters | | | |

J.4 PSI signalling of audio purpose

Table J.3 lists the combination of valid parameters that shall be signalled in the PMT. All other combinations are invalid and shall not be used.

Table J.3: PMT field values for different audio purposes

| Audio purpose | audio_type (see note 1) | mix_type (see note 2) | editorial_classification (see note 2) |
|--|----------------------------|--------------------------|--|
| Main audio (see note 3) | 0x00 or 0x01 | 1 | 0x00 |
| Audio description (broadcast-mix) | 0x00, 0x01 or 0x03 | 1 | 0x01 |
| Audio description (receiver-mix) | 0x03 | 0 | 0x01 |
| Clean audio (broadcast-mix) | 0x02 | 1 | 0x02 |
| Parametric data dependent stream (see note 4) | 0x02 | 0 | 0x04 |
| Spoken subtitles (broadcast-mix) | 0x00, 0x01 or 0x03 | 1 | 0x03 |
| Spoken subtitles (receiver-mix) | 0x03 | 0 | 0x03 |
| Unspecific audio for the general audience | any | 0 or 1 | 0x17 |
| user defined | any | 0 or 1 | 0x18 to 0x1F |
| NOTE 1: audio_type is broadcast in the ISO_639_language_descriptor [15]. | | | |
| NOTE 2: mix_type and editorial_classification are broadcast in the supplementary_audio descriptor (see clause 6.4.11). | | | |
| NOTE 3: The supplementary_audio descriptor is optional in this case. | | | |
| NOTE 4: Embedded parametric data shall not be signalled with editorial_classification 0x04. | | | |

J.5 SAOC-DE parametric data streams

J.5.1 Introduction

The SAOC-DE parametric data is usually embedded as ancillary data in the audio stream. Alternatively, if the audio codec does not provide such capabilities, the SAOC-DE parametric data may also be conveyed as a separate dependent data stream on a separate component.

When the SAOC-DE parametric data is conveyed in a separate elementary stream, receivers shall use attributes such as language_code and editorial_classification to determine which complete and independent audio stream to combine with the dependent SAOC-DE parametric data stream.

Receivers shall apply the following restrictions when combining an audio and a dependent SAOC-DE parametric data stream. The restrictions depend on values signalled in the supplementary audio descriptor:

- A dependent SAOC-DE parametric data stream shall be signalled as mix_type 0 and shall only be combined with a complete and independent audio stream (mix_type 1).
- A dependent SAOC-DE parametric data stream shall only be combined with a complete and independent stream associated with the same language code.

J.5.2 PSI PMT signalling

When a dependent SAOC-DE parametric data stream is present, it shall be signalled in the ES descriptor loop of the PSI PMT with both an ISO_639_language_code in the ISO_639_language_descriptor [36] set to reflect the content of the audio stream, and a supplementary_audio descriptor. The audio_type in the ISO_639_language_descriptor shall be set to 0x2. The supplementary_audio descriptor shall set the mix_type to zero, and the editorial_classification to 0x4.

When the SAOC-DE parametric data is embedded as ancillary data in an AAC audio stream, the presence of embedded SAOC-DE shall be signalled using the AAC descriptor defined in clause H.2.1. The ISO_639_language_descriptor [36] and an optional supplementary_audio descriptor shall be used to reflect the content of the audio stream, but set as if the SAOC-DE parametric data was not present. A supplementary audio descriptor with editorial_classification 0x4 shall not be used for audio components with embedded SAOC-DE parametric data.

A stream_identifier descriptor should be present for both dependent, and embedded SAOC-DE parametric data elementary streams.

J.5.3 EIT signalling

When a dependent SAOC-DE parametric data stream is present, a component descriptor (stream_content 0x3, stream_content_ext 0xF, component_type 0x80) with the same value of component_tag as that used in the PSI PMT

stream identifier descriptor of the dependent SAOC-DE parametric data stream, should be present in the descriptor loop in the EIT for each event for which the dependent SAOC-DE data stream is broadcast.

When the SAOC-DE parametric data is embedded as ancillary data in an AAC audio stream, a component descriptor (stream_content 0x6, stream_content_ext 0xF, component_type 0xA0) with the same value of component_tag as that used in the PSI PMT stream identifier descriptor for the audio stream, should be present in the descriptor loop in the EIT for each event for which the SAOC-DE data stream is broadcast. Additional component descriptors with the same value of component_tag should also be present to indicate the type of the AAC audio stream itself.

Table J.4: Use of the component descriptor for SAOC-DE parametric data streams

| Audio coding | stream_content_ext | stream_content | component_type |
|--|---------------------------|-----------------------|-----------------------|
| HE AAC, HE AAC v2, or AAC (see note) with SAOC-DE ancillary data | 0xF | 0x6 | 0xA0 |
| Dependent SAOC-DE data stream | 0xF | 0x3 | 0x80 |
| NOTE: AAC also uses this type (see clause 6.2.8) | | | |

Annex K (normative)

Extended event linkage descriptor usage

This annex is to explain the valid signalling combinations for the extended event linkage descriptor which is defined in clause 6.2.19.3. Other signalling combinations not specified in this Annex are reserved for future use. It is expected this Annex will expand as use cases for new signalling combinations become evident.

The extended event linkage descriptor allows a broadcaster to signal linkages between EPG events in the same or different services. The descriptor is added to the EIT of the “source” event and signals a linkage to one or more “destination” events.

As explained in annex I, the exact type of a service cannot be determined using the service_type alone. So in the following valid use-cases, the description of the type of service (during the event) is described generically rather than by service_type.

Linkages from an event to itself shall not be used, since receivers can determine the same information from the PSI and SI without the need for an extended event linkage descriptor.

For certain use cases it may be applicable to link from a "source" event to a single "destination" event multiple times, each with a different combination of linkage_type and link_type.

The valid signalling combinations of extended event linkage descriptor are shown in table K.1.

Table K.1: Valid use-cases of extended event linkage descriptor

| Type of source event | Type of destination event | linkage_type (see note 1) | link_type (see note 2) | Use-case |
|----------------------|---------------------------|---------------------------|------------------------|---|
| SD | SD | 0x0E | 0 | Link to alternate event instances also in SD. |
| SD | HD | 0x0E | 1 | Link to event in HD. |
| SD | FC-3DTV | 0x0E | 2 | Link to event in frame compatible plano-stereoscopic. |
| SD | SC-3DTV MVC | 0x0E | 1 | Link to event in service compatible plano-stereoscopic MVC which provides a HD alternative. (see note 3) |
| SD | SC-3DTV MVC | 0x0E | 3 | Link to event in service compatible plano-stereoscopic MVC. |
| SD | UHD | 0x0F | 0 | Link to event in UHD. |
| SD | SFC-3DTV HEVC | 0x0F | 1 | Link to event in service frame compatible plano-stereoscopic. |
| Type of source event | Type of destination event | linkage_type (see note 1) | link_type (see note 2) | Use-case |
| HD | SD | 0x0E | 0 | Link to event in SD. |
| HD | HD | 0x0E | 1 | Link to alternate event instances also in HD. |
| HD | FC-3DTV | 0x0E | 2 | Link to event in frame compatible plano-stereoscopic. |
| HD | SC-3DTV MVC | 0x0E | 3 | Link to event in service compatible plano-stereoscopic MVC. |
| HD | UHD | 0x0F | 0 | Link to event in UHD. |
| HD | SFC-3DTV HEVC | 0x0F | 1 | Link to event in service frame compatible plano-stereoscopic. |
| Type of source event | Type of destination event | linkage_type (see note 1) | link_type (see note 2) | Use-case |
| FC-3DTV | SD | 0x0E | 0 | Link to event in SD. |
| FC-3DTV | HD | 0x0E | 1 | Link to event in HD. |
| FC-3DTV | FC-3DTV | 0x0E | 2 | Link to alternate event instances also in frame compatible plano-stereoscopic. |
| FC-3DTV | SC-3DTV MVC | 0x0E | 1 | Link to event in service compatible plano-stereoscopic MVC which provides a HD alternative. (see note 3) |
| FC-3DTV | SC-3DTV MVC | 0x0E | 3 | Link to event in service compatible plano-stereoscopic MVC. |
| FC-3DTV | UHD | 0x0F | 0 | Link to event in UHD. |
| FC-3DTV | SFC-3DTV HEVC | 0x0F | 1 | Link to event in service frame compatible plano-stereoscopic. |
| Type of source event | Type of destination event | linkage_type (see note 1) | link_type (see note 2) | Use-case |
| SC-3DTV MVC | SD | 0x0E | 0 | Link to event in SD. |
| SC-3DTV MVC | HD | 0x0E | 1 | Link to event in HD. |
| SC-3DTV MVC | FC-3DTV | 0x0E | 2 | Link to event in frame compatible plano-stereoscopic. |
| SC-3DTV MVC | SC-3DTV MVC | 0x0E | 1 | Link to event in service compatible plano-stereoscopic MVC which provides a better HD alternative. (see note 3) |
| SC-3DTV MVC | SC-3DTV MVC | 0x0E | 3 | Link to alternate event instances also in service compatible plano-stereoscopic MVC. |
| SC-3DTV MVC | UHD | 0x0F | 0 | Link to event in UHD. |
| SC-3DTV MVC | SFC-3DTV HEVC | 0x0F | 1 | Link to event in service frame compatible plano-stereoscopic. |
| Type of source event | Type of destination event | linkage_type (see note 1) | link_type (see note 2) | Use-case |
| SFC-3DTV HEVC | SD | 0x0E | 0 | Link to event in SD. |
| SFC-3DTV HEVC | HD | 0x0E | 1 | Link to event in HD. |
| SFC-3DTV HEVC | FC-3DTV | 0x0E | 2 | Link to event in frame compatible plano-stereoscopic. |

| | | | | |
|---|----------------------------------|----------------------------------|-------------------------------|---|
| SFC-3DTV HEVC | SC-3DTV MVC | 0x0E | 1 | Link to event in service compatible plano-stereoscopic MVC which provides a better HD alternative. (see note 3) |
| SFC-3DTV HEVC | SC-3DTV MVC | 0x0E | 3 | Link to alternate event instances also in service compatible plano-stereoscopic MVC. |
| SFC-3DTV HEVC | UHD | 0x0F | 0 | Link to event in UHD. |
| SFC-3DTV HEVC | SFC-3DTV HEVC | 0x0F | 1 | Link to alternate event instances also in service frame compatible plano-stereoscopic. |
| Type of source event | Type of destination event | linkage_type (see note 1) | link_type (see note 2) | Use-case |
| UHD | SD | 0x0E | 0 | Link to event in SD. |
| UHD | HD | 0x0E | 1 | Link to event in HD. |
| UHD | FC-3DTV | 0x0E | 2 | Link to event in frame compatible plano-stereoscopic. |
| UHD | SC-3DTV MVC | 0x0E | 3 | Link to event in service compatible plano-stereoscopic MVC. |
| UHD | UHD | 0x0F | 0 | Link to alternate event instances also in UHD. |
| UHD | SFC-3DTV HEVC | 0x0F | 1 | Link to event in service frame compatible plano-stereoscopic. |
| NOTE 1: See table 60 | | | | |
| NOTE 2: See table 66 | | | | |
| NOTE 3: Only use if destination event is the best HD alternative. | | | | |

Annex L (normative)

Service information implementation of DTS Neural Surround™ coded audio in DVB systems

L.0 Introduction

DTS Neural Surround™ constitutes a method of encoding and decoding additional audio channels into a host audio stream. A common application example would be to encode 5.1 channel surround sound into a stereo audio program.

DTS Neural Surround™ coding is an active intensity phase encoding methodology, and does not carry embedded metadata signaling. A system unaware of this processing will operate without impediment. The nature of this encoding permits the carriage independent of audio encoding algorithm, provided sufficient quality is maintained.

A common system application example would be to use DTS Neural Surround™ to encode 5.1 channel surround sound for a stereo MPEG 1 layer 2 host audio, enabling the delivery of surround sound audio in early generations of digital media distribution systems which were designed for stereo audio distribution.

The transmission of DTS Neural Surround™ is optional in DVB systems, as specified in TS 101 154 [9]. However, if DTS Neural Surround™ is transmitted in a DVB system the specifications in this annex shall be followed.

L.1 DTS® Neural Descriptor

The DTS® Neural descriptor is used in the PMT to identify which audio streams have been processed with DTS Neural Surround™ encoding. In such cases, a DTS® Neural descriptor follows the associated audio descriptor in the TS_program_map_section ES_info descriptor loop. The DTS® Neural descriptor is described in table L.1.

Table L.1: DTS® Neural descriptor

| Syntax | Number of bits | Identifier |
|---------------------------|----------------|------------|
| DTS_Neural_descriptor() { | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| descriptor_tag_extension | 8 | uimsbf |
| config_id | 8 | uimsbf |
| for (i=0; i<N; i++) { | | |
| additional_info_byte | 8 | bslbf |
| } | | |
| } | | |

Semantics for the DTS® Neural descriptor:

config_id: This 8-bit field indicates the audio channel configuration of the host audio stream. For a stereo host audio stream it is coded according to table L.2, and for a 5.1 host audio stream it is coded according to table L.3.

Table L.2: Config ID for Neural stereo host audio stream

| config_id | Original audio configuration | Original channel count (see note) |
|--|------------------------------|--------------------------------------|
| 0 | Unknown or undefined | |
| 1 | L, R | 2 |
| 2 | L, R, C | 3 |
| 3 | L, R, Ls, Rs | 4 |
| 4 | L, R, C, Ls, Rs | 5 |
| 5 | L, R, C, Ls, Rs, Cs | 6 |
| 6 | L, R, C, Ls, Rs, Lb, Rb | 7 |
| 7 | L, R, Ls, Rs, Cs | 5 |
| 8 | L, R, Ls, Rs, Lb, Rb | 6 |
| 9 to 255 | reserved for future use | |
| NOTE: The LFE channel is omitted from the original audio configuration for stereo host audio streams. If an LFE channel existed in the original audio configuration, that channel was mixed into the full band channels during DTS Neural Surround™ encoding. An LFE channel can be generated by a bass management system. | | |

Table L.3: Config ID for Neural 5.1 host audio stream

| config_id | Original audio configuration | Original channel count |
|-----------|------------------------------|------------------------|
| 0 | Unknown or undefined | |
| 1 | L, R, C, LFE, Ls, Rs | 5.1 |
| 2 | L, R, C, LFE, Ls, Rs, Cs | 6.1 |
| 3 | L, R, C, LFE, Ls, Rs, Lb, Rb | 7.1 |
| 4 to 255 | reserved for future use | |

additional_info_byte: This 8-bit field is optional and reserves a place for future expansion of this descriptor.

Annex M (normative): Signalling of next-generation audio

M.1 Overview

Next Generation Audio systems provide broadcasters, operators and content providers with more flexibility to create and deliver their content. These NGA systems introduce a number of new concepts and techniques, including:

- the capability to provide Immersive Audio with the addition of height elements;
- the capability to provide personalization options to broadcasters and consumers;
- the introduction of Audio Objects to facilitate immersive and personalized audio.

The usage of multiple audio preselections allows broadcasters, operators and content providers to offer different variations of the content from a single NGA audio programme. A more detailed explanation of these concepts and techniques is given in annex K of TS 101 154 [9].

The audio preselection descriptor is used for signalling the NGA features that the receiver may use for selection of the appropriate audio preselection to present the audio programme to the user.

For each signalled audio preselection, the descriptor provides information about the available accessibility features and languages, indication about the preferred reproduction layout and the interactivity options.

M.2 PSI PMT Signalling

If the audio preselection descriptor is used for one audio programme contained in one or more NGA elementary streams, neither the `ISO_639_language_descriptor` nor the `supplementary_audio_descriptor` should be present in the `ES_info` descriptor loop of any of the NGA streams (main or auxiliary) of an audio programme.

When an `audio_preselection_descriptor` is present in the `ES info` loop for an NGA elementary stream in the PMT, receivers shall ignore any `supplementary_audio_descriptor` or `ISO_639_language_descriptor` which are present in the same `ES info` loop.

M.3 Mapping of codec specific values to the audio preselection descriptor (informative)

Table M.1 lists for each NGA codec the mapping between the codec specific values and the fields present in the audio preselection descriptor.

Table M.1: Mapping of codec specific values to the audio preselection descriptor

| NGA codec | Field in the audio preselection descriptor | Mapping |
|------------------|---|---|
| AC-4 Part-2 | num_preselection | num_presentation field (within the AC-4 TOC according to clause 6.7 of TS 101 154 [9]). |
| AC-4 Part-2 | preselection_id | presentation_group_index field of the Preselection (within the AC-4 TOC according to clause 6.7 of TS 101 154 [9]). |
| AC-4 Part-2 | ISO_639_language_code | The first language indicated by this field should correspond to the information conveyed in the language_tag_bytes field (within the AC-4 TOC according to clause 6.7 of TS 101 154 [9]) for the presentation, that is tagged as dialog or complete main. |
| AC-4 Part-2 | audio_description | This field should correspond to the information conveyed in the content_classifier field of the ac4_presentation_info structure (within the AC-4 TOC according to clause 6.7 of TS 101 154 [9]) for the presentation where at least one substream is tagged as visually impaired (VI). |
| AC-4 Part-2 | spoken_subtitles | This field should correspond to the information conveyed in the content_classifier field of the ac4_presentation_info structure (within the AC-4 TOC according to clause 6.7 of TS 101 154 [9]) for the presentation where at least one substream is tagged as voice over (VO). |
| AC-4 Part-2 | dialogue_enhancement | This field should correspond to the information conveyed in the content_classifier field of the ac4_presentation_info structure (within the AC-4 TOC according to clause 6.7 of TS 101 154 [9]) for the presentation where at least one substream is tagged as hearing impaired (HI) or dialog (D). |
| MPEG-H Audio | num_preselection | mae_numGroupPresets field as specified in clause 6.8 of TS 101 154 [9]. |
| MPEG-H Audio | preselection_id | mae_GroupPresetID field as specified in clause 6.8 of TS 101 154 [9]. |
| MPEG-H Audio | ISO_639_language_code | The first language indicated by this field should correspond to the information conveyed in mae_contentLanguage of the default dialog element: the maeGroup which is marked as default in mae_switchGroupDefaultGroupID and is tagged in mae_contentKind as dialog. This information is carried in the AudioSceneInformation() of the MPEG-H Audio stream according to clause 6.8 of TS 101 154 [9]. |
| MPEG-H Audio | audio_description | This field should correspond to the mae_groupPresetKind value in the mae_GroupPresetDefinition() structure and the mae_contentKind values in the mae_ContentData() structures in the AudioSceneInformation() of the MPEG-H Audio stream according to clause 6.8 of TS 101 154 [9]. A value of 1 in audio_description field corresponds to value 7 in mae_groupPresetKind and value 9 in mae_contentKind. |
| MPEG-H Audio | spoken_subtitles | This field should correspond to the mae_groupPresetKind value in the mae_GroupPresetDefinition() structure and the mae_contentKind values in the mae_ContentData() structures in the AudioSceneInformation() of the MPEG-H Audio stream according to clause 6.8 of TS 101 154 [9]. A value of 1 in spoken_subtitles field corresponds to value 8 in mae_groupPresetKind and value 8 in mae_contentKind. |
| MPEG-H Audio | dialogue_enhancement | This field should correspond to the mae_groupPresetKind value in the mae_GroupPresetDefinition() structure and the mae_contentKind values in the mae_ContentData() structures in the AudioSceneInformation() of the MPEG-H Audio stream according to clause 6.8 of TS 101 154 [9]. A value of 1 in dialogue_enhancement field corresponds to value 2 in mae_contentKind. |

Annex N (informative): Bibliography

- IEC 62106: "Specification of the radio data system (RDS) for VHF/FM sound broadcasting in the frequency range from 87,5 to 108,0 MHz".
- RDS-Forum SPB 490: "RDS Universal Encoder Communication Protocol", Final Version 6.01, June 2003.
- Alliance for Telecommunications Industry Solutions (ATIS), ATIS-0800006: "IIF Default Scrambling Algorithm (IDSA) IPTV Interoperability Specification".
- ETSI TS 102 584: "Digital Video Broadcasting (DVB); DVB-SH implementation guidelines".

Annex O (informative): Change History

| Date | Version | Information about changes |
|-----------|---------|--|
| June 2015 | 1.15.1 | <p>Implemented Change Requests:</p> <ul style="list-style-type: none"> • 847: 3DTV phase 3 signalling • 953: new URI linkage type for MRS identifier • 986: Expanding the capability of the AAC Descriptor to support clean_audio • 1008: Allocate bit in adaptation field_data_identifier for TSAP timeline • 1031: MVC dependent view component descriptor (Note about 24 Hz is confusing) • 1035: annex A code point differences since revision 1.12.1 • 1041: a reserved field should be reserved for future use in supp. audio desc. • 1093: service type 0x1F for HEVC digital television service • 1096: update Annex I for HEVC • 1098: stream content and component type for HEVC • 1142: signal presence of multi-region disparity SEI data • 1156: naming of HEVC UHD component type? • 1181: promote various audio annexes from informative to normative • 1203: various typos in r99 of draft of En300 468 • 1204: explain use of component descriptor for service list filtering • 1205: add signalling for Dolby AC-4 • 1298: Note 7 in component descriptor table 26 in 6.2.8 • 1310: Note 4 in Annex J tables J.1 and J.2 confusing. • 1311: Annex I.2.5 add additional text to warn explicit about future backwards compatible services • 1313: table K.1 in Annex K needs updating to include new SFC-3DTV event linkage value. • 1317: update C2 delivery system descriptor for PLP bundling • 1320: add a revision history table at the end • 1326: descriptor_tag_extension 0x16 for C2 bundle delivery system descriptor |

| Date | Version | Information about changes |
|-----------|---------|---|
| June 2017 | 1.16.1 | <p>Implemented Change Requests:</p> <ul style="list-style-type: none"> • 671: confusion wrt. bit order in bitmapped fields • 944: ETSI reference for TM-CSS spec • 1398: new classifications in content descriptor requested by UK DTG • 1399: typo in 6.4.5.4 • 1420: table K.1 typo • 1450: UHD-1 Phase 2 signalling • 1458: unify wording to sub_table • 1463: add user defined range for data_service_id • 1710: next-generation audio (NGA) signalling • 1713: clarify MSB/LSB for uimsbf fields in definitions • 1714: signalling support for TM-MIS authentication solution in TS 102 809 • 1736: replace BlueBook reference in table 14 • 1737: add signalling for TTML subtitles • 1744: update codec references to point to TS101154 instead of/in addition to the codec specs • 1951: reference MPEG for CRC32 (void annex B) • 1953: stale ISO/IEC 13818-1:2007/AMD1 reference • 2065: enhanced modes of CSA3 were never specified • 2066: ambiguous reference to ATIS-0800006 • 2096: harmonise descriptor naming for DTS descriptors • 2099: URI linkage type registration solution • 2176: next-generation audio overview • 2206: promote TS 101 162 to a normative reference • 1008: adaptation_field_data_identifier for TSAP timeline (not) • 1417: S2 satellite delivery descriptor - wording and referencing issues. • 1418: is the S2 delivery system descriptor stand-alone or used together with an S delivery system descriptor? • 1419: remove references to S2 NBC-BS mode • 1479: Incomplete title for Table 26: stream_content and component_type • 1570: new section to clarify of bit order and transmission order • 1711: Inconsistent table lines in Table 26: stream_content and component_type • 1718: Annex N changelog corrections for v1.15.1 • 2053: Ambiguous wording implying a descriptor ordering in 6.2.39 Stream Identifier Descriptor • 2168: typo in heading of 6.4.16.0 • 2170: 6.2.1 Adaptation field data descriptor: add reference to annex D of Ts101154 • 2183: Use of the supplementary audio descriptor with AC-4 |

History

| Document history | | |
|------------------|--|--|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |