Final draft ETSI EN 300 468 V1.11.1 (2009-12)

European Standard (Telecommunications series)

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

Reference
REN/JTC-DVB-266

Keywords
broadcasting, digital, DVB, MPEG,
service, TV, video

ETSI

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

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

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

Important notice

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

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

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

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

Copyright Notification

Contents

Intellectual Property Rights

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

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

Foreword

This European Standard (Telecommunications series) 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), and is now submitted for the ETSI standards One-step Approval Procedure.

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel: +41 22 717 21 11
Fax: +41 22 717 24 81

Founded in September 1993, the DVB Project is a market-led consortium of public and private sector organizations in the television industry. Its aim is to establish the framework for the introduction of MPEG-2 based digital television services. Now comprising over 200 organizations from more than 25 countries around the world, DVB fosters market-led systems, which meet the real needs, and economic circumstances, of the consumer electronics and the broadcast industry.

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

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 [18] 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 TR 101 211 [i.2].

2 References

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

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

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

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

2.1 Normative references

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

[1] 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: "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (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 102 114: "DTS Coherent Acoustics; Core and Extensions".

[13] ETSI TS 102 323: "Digital Video Broadcasting (DVB); Carriage and signalling of TV-Anytime information in DVB transport streams".

[14] ETSI TS 102 366: "Digital Audio Compression (AC-3, Enhanced AC-3) Standard".

[15] ETSI TS 102 812: "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1.1".

[16] ISO/IEC 10646 (2003): "Information technology - Universal Multiple-Octet Coded Character Set (UCS)".

[17] ISO/IEC 11172-3: "Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 3: Audio".

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

[19] ISO/IEC 13818-2: "Information technology - Generic coding of moving pictures and associated audio information: Video".

[20] ISO/IEC 13818-3: "Information technology - Generic coding of moving pictures and associated audio information - Part 3: Audio".

[21] ISO/IEC 14496-3 (2005): "Information technology - Coding of audio- visual objects - Part 3: Audio".

[22] ISO/IEC 6937: "Information technology - Coded graphic character set for text communication - Latin alphabet".

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

[24] ISO/IEC 8859-2: "Information technology - 8-bit single-byte coded graphic character sets - Part 2: Latin alphabet No. 2".

[25] ISO/IEC 8859-3: "Information technology - 8-bit single-byte coded graphic character sets - Part 3: Latin alphabet No. 3".

[26] ISO/IEC 8859-4: "Information technology - 8-bit single-byte coded graphic character sets - Part 4: Latin alphabet No. 4".

[27] ISO/IEC 8859-5: "Information technology - 8-bit single-byte coded graphic character sets - Part 5: Latin/Cyrillic alphabet".

[28] ISO/IEC 8859-6: "Information technology - 8-bit single-byte coded graphic character sets - Part 6: Latin/Arabic alphabet".

[29] ISO/IEC 8859-7: "Information technology - 8-bit single-byte coded graphic character sets - Part 7: Latin/Greek alphabet".

[30] ISO/IEC 8859-8: "Information technology - 8-bit single-byte coded graphic character sets - Part 8: Latin/Hebrew alphabet".

[31] ISO/IEC 8859-9: "Information technology - 8-bit single-byte coded graphic character sets - Part 9: Latin alphabet No. 5".

[32] ISO/IEC 8859-10: "Information technology - 8-bit single-byte coded graphic character sets - Part 10: Latin alphabet No. 6".

[33] ISO/IEC 8859-11: "Information technology - 8-bit single-byte coded graphic character sets - Part 11: Latin/Thai alphabet".

[34] ISO/IEC 8859-13: "Information technology - 8-bit single-byte coded graphic character sets - Part 13: Latin alphabet No. 7".

[35] ISO/IEC 8859-14: "Information technology - 8-bit single-byte coded graphic character sets - Part 14: Latin alphabet No. 8 (Celtic)".

[36] ISO/IEC 8859-15: "Information technology - 8-bit single-byte coded graphic character sets - Part 15: Latin alphabet No. 9".

[37] CENELEC EN 50221: "Common interface specification for conditional access and other digital video broadcasting decoder applications".

[38] IEC 61883 (parts 1 and 4): "Consumer audio/video equipment - Digital interface".

[39] IEEE 1394.1: "IEEE Standard for High Performance Serial Bus Bridges".

[40] ISO 8601: "Data elements and interchange formats - Information interchange - Representation of dates and times".

[41] ISO 3166 (all parts): "Codes for the representation of names of countries and their subdivisions".

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

[43] ITU-R Recommendation BS.1196-1 (annex 2): "Audio coding for digital terrestrial television broadcasting".

NOTE: Annex 2 contains additional information on the AC-3 audio encoding algorithm and decoding requirements, relevant to the present document. Appendix 1 to annex 2 of this Recommendation should be disregarded as it is not applicable to the present document.

[44] 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.

[45] ETSI ES 201 812: "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.0.3".

[46] ETSI TS 102 825 (parts 1 to 5, 7, 9 and 10): "Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM)".

[47] 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)".

[48] ETSI TS 102 770: "Digital Video Broadcasting (DVB); System Renewability Messages (SRM) in DVB Systems".

[49] ETSI EN 302 583: "Digital Video Broadcasting (DVB); Framing Structure, channel coding and modulation for Satellite Services to Handheld devices (SH) below 3 GHz".

[50] ETSI TS 102 772: "Digital Video Broadcasting (DVB); Specification of Multi-Protocol Encapsulation - inter-burst Forward Error Correction".

[51] IETF RFC 2045: "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies".

2.2 Informative references

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

[i.1] ETSI TS 101 162: "Digital Video Broadcasting (DVB); Allocation of Service Information (SI) and Data Broadcasting Codes for Digital Video Broadcasting (DVB) systems".

[i.2] ETSI TR 101 211: "Digital Video Broadcasting (DVB); Guidelines on implementation and usage of Service Information (SI)".

[i.3] ETSI TS 102 590: "Digital Video Broadcasting (DVB); Mulimedia Home Platform 1.2".

[i.4] ETSI TR 102 825 (parts 6, 8, 11 to 13): "Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM)".

[i.5] Void.

[i.6] ETSI TS 102 201: "Digital Video Broadcasting (DVB); Interfaces for DVB Integrated Receiver Decoder (DVB-IRD) ".

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 method

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 annex C of TS 101 154 [9].

bouquet: collection of services marketed as a single entity

broadcaster (SERVICE Provider): organization which assembles a sequence of events or programmes to be delivered to the viewer based upon a schedule

cell: geographical area that is covered with DVB-T signals delivering one or more particular transport 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, programmes and events

EXAMPLE: Videoguard, Eurocrypt.

delivery system: physical medium by which one or more multiplexes are transmitted

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

Entitlement Management Messages (EMM): private Conditional Access information which specify the authorization levels or the services of specific decoders

NOTE: They may be addressed to individual decoder or groups of decoders.

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

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

forbidden: when used in the clauses defining the coded bit stream, indicates that the value shall never be used

MPEG-2: See ISO/IEC 13818.

NOTE: Systems coding is defined in part 1 [18]. Video coding is defined in part 2 [19]. Audio coding is defined in part 3 [20].

multiplex: stream of all the digital data carrying one or more services within a single physical channel

network: collection of MPEG-2 Transport Stream (TS) multiplexes transmitted on a single delivery system

EXAMPLE: All digital channels on a specific cable system.

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

original_network_id: unique identifier of a network

programme: concatenation of one or more events under the control of a broadcaster e.g. news show, entertainment show

repeater: equipment which receives and re-transmits a DVB-T signal

NOTE: It can not change the TPS bits and thus 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 is 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 is set to "1".

section: syntactic structure used for mapping all service information defined in EN 300 468 into ISO/IEC 13818-1 [18] TS packets

service: sequence of programmes under the control of a broadcaster which can be broadcast as part of a schedule

service_id: unique identifier of a service within a TS

Service Information (SI): digital data describing the delivery system, content and scheduling/timing of broadcast data streams, etc.

NOTE: It includes MPEG-2 PSI together with independently defined extensions.

subcell: geographical area that is part of the cells coverage area and that is covered with DVB-T 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:

for a NIT: the same table_id_extension (network_id) and version_number;

for a BAT: the same table_id_extension (bouquet_id) and version_number;

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 allows to modulate a baseband transport stream and to broadcast it on one frequency

Transport Stream (TS): data structure defined in ISO/IEC 13818-1 [18]

NOTE: It is the basis of the DVB standards.

transport_stream_id: unique identifier of a TS within an original network

transposer: type of repeater which receives a DVB-T 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.

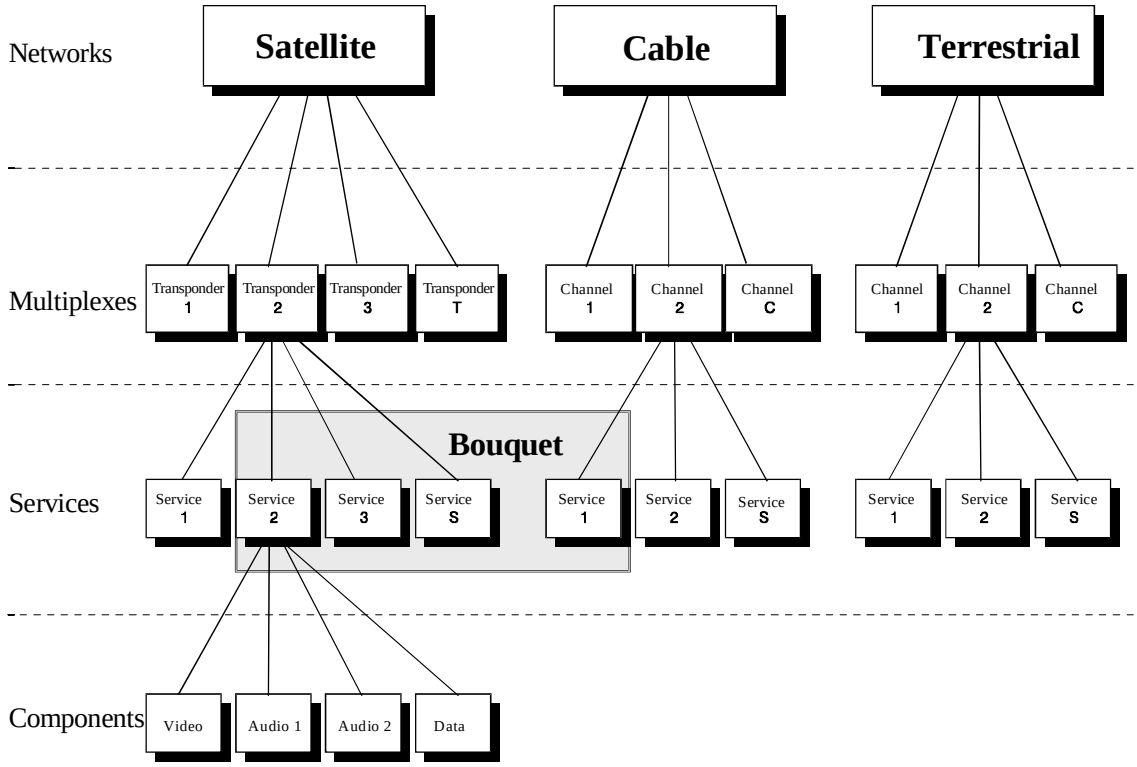


Figure 1: Digital broadcasting, service delivery model

3.2 Abbreviations

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

AAC Advanced Audio Coding

NOTE: See TS 101 154 [9].

AC-3 Dolby AC-3 audio coding

NOTE: See ITU-R Recommendation BS.1196-1 [43].

ASCII American Standard Code for Information Interchange
BAT Bouquet Association Table
BCD Binary Coded Decimal
bslbf bit string, left bit first
CA Conditional Access
CAT Conditional Access Table
CLUT Colour Look-Up Table
CPCM Content Protection Copy Management
CRC Cyclic Redundancy Check
CSA Common Scrambling Algorithm
DAB Digital Audio Broadcasting
DIT Discontinuity Information Table
DSNG Digital Satellite News Gathering
DTS Digital Theater Systems

NOTE: See TS 102 114 [12].

DVB Digital Video Broadcasting
DVD Digital Versatile Disc
EBU European Broadcasting Union
ECM Entitlement Control Message
EIT Event Information Table
EMM Entitlement Management Message
EPG Electronic Programme Guide
ES Elementary Stream
FEC Forward Error Correction
FTA Free-To-Air
HD High Definition (video)
HE-AAC High Efficiency AAC
HP High Priority
IEC International Electrotechnical Commission
INT IP Notification Table

NOTE: See EN 301 192 [4].

IP Internet Protocol
IRD Integrated Receiver Decoder
ISO International Organization for Standardization
LP Low Priority
LSB Least Significant Bit
MJD Modified Julian Date
MPE MultiProtocol Encapsulation
MPEG Moving Pictures Expert Group
MSB Most Significant Bit
NBC-BS Non Backwards Compatible Broadcast Services (DVB-S2)
NDA Non Disclosure Agreement
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
PID	Packet IDentifier
PIL	Programme Identification Label
PMT	Program Map Table
PSI	Program Specific Information
PSTN	Public Switched Telephone Network
QAM	Quadrature Amplitude Modulation
QPSK	Quaternary Phase Shift Keying
RAR	Resolution Authority Record
rpchof	remainder polynomial coefficients, highest order first
RDS	Radio Data System
RNT	RAR Notification Table
RS	Reed-Solomon
RST	Running Status Table
ScF	Scale Factor
SD	Standard Definition (video)
SDT	Service Description Table
SI	Service Information
SIT	Selection Information Table
SMATV	Satellite Master Antenna Television
SMI	Storage Media Interoperability
SRM	System Renewability Message
ST	Stuffing Table
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
TVA	TV-Anytime (http://www.tv-anytime.org)
UECP	Universal Encoder Communication Protocol (RDS)
uimsbf	unsigned integer most significant bit first
UTC	Universal Time, Co-ordinated
VBI	Vertical Blanking Interval
VPS	Video Programme System
WSS	Wide Screen Signalling

4 Service Information (SI) description

ISO/IEC 13818-1 [18] 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 [18] specification, but the data format is outside the scope of ISO/IEC 13818-1 [18]. 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.

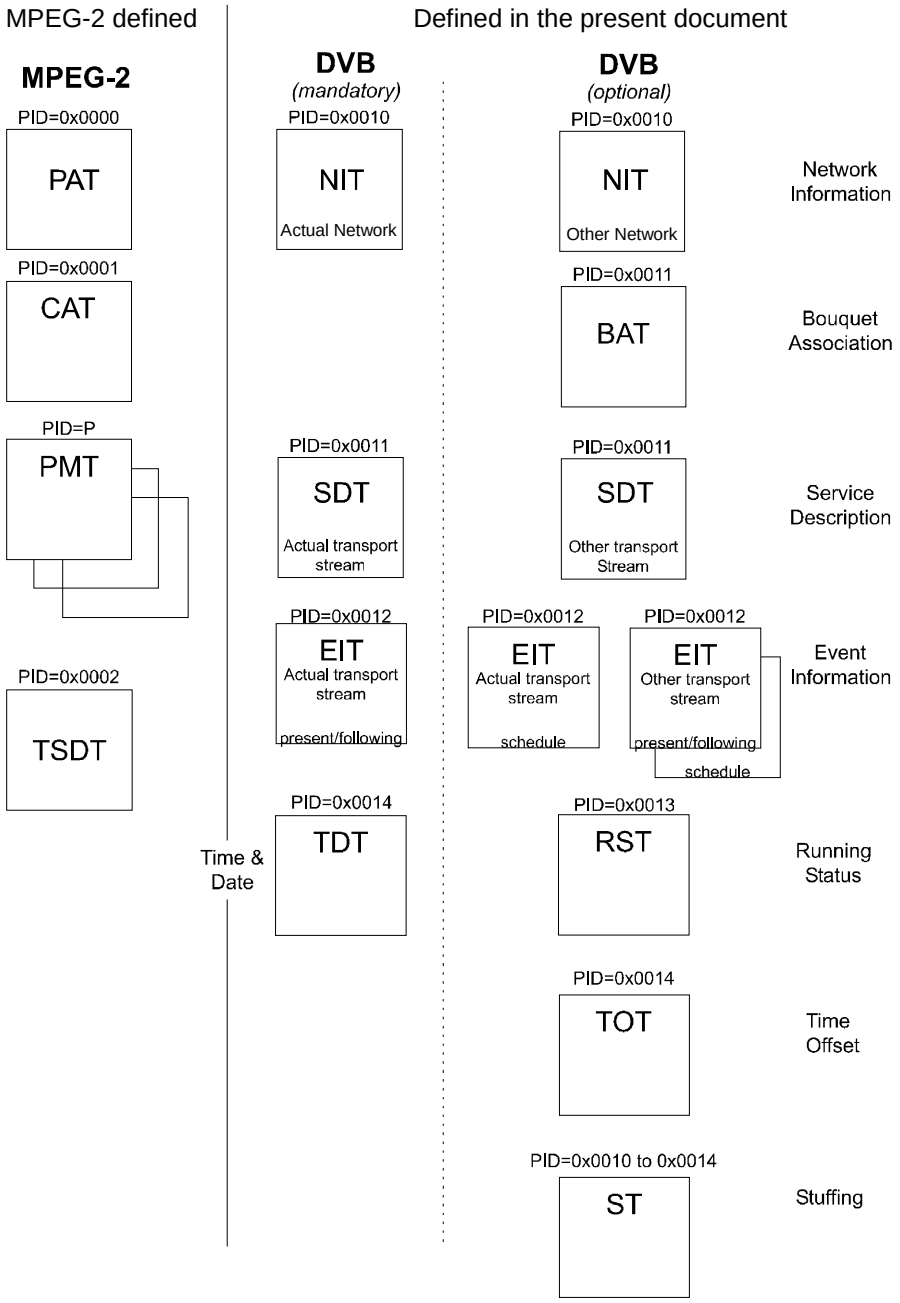


Figure 2: General organization of the Service Information (SI)

5 Service Information (SI) tables

5.1 SI table mechanism

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 [18].

5.1.1 Explanation

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 [18].

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 [13])	0x0012
RST, ST	0x0013
TDT, TOT, ST	0x0014
network synchronization	0x0015
RNT (TS 102 323 [13])	0x0016
reserved for future use	0x0017 to 0x001B
inband signalling	0x001C
measurement	0x001D
DIT	0x001E
SIT	0x001F

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 to 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 [15])
0x75	container section (TS 102 323 [13])
0x76	related content section (TS 102 323 [13])
0x77	content identifier section (TS 102 323 [13])
0x78	MPE-FEC section (EN 301 192 [4])
0x79	resolution notification section (TS 102 323 [13])
0x7A	MPE-IFEC section (TS 102 772 [50])

Value	Description
0x7B to 0x7D	reserved for future use
0x7E	discontinuity_information_section
0x7F	selection_information_section
0x80 to 0xFE	user defined
0xFF	reserved

5.1.4 Repetition rates and random access

In systems where random access is a consideration, it is recommended to re-transmit SI sections specified within the present document several times, even when changes do not occur in the configuration.

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 [18]) 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.

5.2 Table definitions

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 [18].

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 may be found in TS 101 162 [i.1]. 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 per TS 101 162 [i.1].

Guidelines for the processing of SI at transitions between delivery media boundaries, e.g. from satellite to cable or SMATV systems, can be found in TR 101 211 [i.2].

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 takes the value assigned to the actual network in TS 101 162 [i.1]. Any sections of an 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 allocated to the other network in TS 101 162 [i.1].

Table 3: Network information section

Syntax	Number of bits	Identifier
network_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
network_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
network_descriptors_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
reserved_future_use	4	bslbf
transport_stream_loop_length	12	uimsbf
for(i=0;i<N;i++){		
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
reserved_future_use	4	bslbf
transport_descriptors_length	12	uimsbf
for(j=0;j<N;j++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

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. Allocations of the value of this field are found in TS 101 162 [i.1].

version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table 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 B 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 described in TS 101 162 [i.1]. 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
for(i=0;i<N;i++){		
descriptor()		
}		
reserved_future_use	4	bslbf
transport_stream_loop_length	12	uimsbf
for(i=0;i<N;i++){		
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
reserved_future_use	4	bslbf
transport_descriptors_length	12	uimsbf
for(j=0;j<N;j++){		
descriptor()		
}		
}		
CRC_32	32	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. Allocations of the value of this field are found in TS 101 162 [i.1].

version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table 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 B 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	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
original_network_id	16	uimsbf
reserved_future_use	8	bslbf
for (i=0;i<N;i++){		
service_id	16	uimsbf
reserved_future_use	6	bslbf
EIT_schedule_flag	1	bslbf
EIT_present_following_flag	1	bslbf
running_status	3	uimsbf
free_CA_mode	1	bslbf
descriptors_loop_length	12	uimsbf
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 TR 101 211 [i.2] 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 TR 101 211 [i.2] 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 B 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 (NVOD) reference service where it may have more than two event descriptions. The event schedule tables for either the actual TS or other TSs, contain a list of events, in the form of a schedule, namely, including events taking place at some time beyond the next event. The EIT schedule tables are optional. 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 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).

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 1: 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 2: 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 B 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 B 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

This clause describes the different descriptors that can be used within the SI (for further information refer to the document, see TR 101 211 [i.2]).

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 (see note 1)
network_name_descriptor	0x40	*	-	-	-	-	-	-
service_list_descriptor	0x41	*	*	-	-	-	-	-
stuffing_descriptor	0x42	*	*	*	*	-	-	*
satellite_delivery_system_descriptor	0x43	*	-	-	-	-	-	-
cable_delivery_system_descriptor	0x44	*	-	-	-	-	-	-
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	*	-	-	-	-	-	-

Descriptor	Tag value	NIT	BAT	SDT	EIT	TOT	PMT	SIT (see note 1)
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	0x6F	-	-	-	-	-	*	-
adaptation_field_data_descriptor	0x70	-	-	-	-	-	*	-
service_identifier_descriptor (see [15])	0x71	-	-	*	-	-	-	-
service_availability_descriptor	0x72	-	-	*	-	-	-	-
default_authority_descriptor (TS 102 323 [13])	0x73	*	*	*	-	-	-	-
related_content_descriptor (TS 102 323 [13])	0x74	-	-	-	-	-	*	-
TVA_id_descriptor (TS 102 323 [13])	0x75	-	-	-	*	-	-	-
content_identifier_descriptor (TS 102 323 [13])	0x76	-	-	-	*	-	-	-
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.3])	0x7D	*	*	*	*	*	*	*
FTA_content_management_descriptor	0x7E	*	*	*	*	-	-	-
extension descriptor	0x7F	*	*	*	*	*	*	*
user defined	0x80 to 0xFE							
forbidden	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 [18]) and INT (TS 102 006 [11]).
NOTE 4: * Possible location.

6.2 Descriptor coding

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 [18]. 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.

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 coded according to ISO/IEC 13818-1 [18]. It 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	8	uimsbf
descriptor_length	8	uimsbf
adaptation_field_data_identifier	8	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 (LSB)	announcement switching data field (TS 101 154 [9])
b_1	AU_information data field (TS 101 154 [9])
b_2	PVR_assist_information_data_field (TS 101 154 [9])
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

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 ISO/IEC 13818-3 [20] and ISO/IEC 11172-3 [17]. 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	8	uimsbf
descriptor_length	8	uimsbf
ancillary_data_identifier	8	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 (LSB)	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 D.5)
b_6	RDS via UECP (TS 101 154 [9])
b_7	Reserved for future use

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 (LSB)	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

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. Allocations of the value of this field are found in TS 101 162 [i.1].

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 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 ($90^{o} / 2^{15}$). 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 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 ($180^{o} / 2^{15}$). 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 ($90^{o} / 2^{15}$).

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 ($180^{o} / 2^{15}$).

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 ($90^{o} / 2^{15}$). 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 ($180^{o} / 2^{15}$). 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 ($90^{o} / 2^{15}$).

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 ($180^{o} / 2^{15}$).

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).

Table 25: Component descriptor

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

Semantics for the component descriptor:

stream_content: This 4-bit field specifies the type (video, audio, or EBU-data) of stream. The coding of this field is specified in table 26.

component_type: This 8-bit field specifies the type of the video, audio or EBU-data 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.

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 [42]. 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 [23] 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 and component_type

Stream_content	Component_type	Description
0x00	0x00 to 0xFF	reserved for future use
0x01	0x00	reserved for future use
0x01	0x01	MPEG-2 video, 4:3 aspect ratio, 25 Hz (see note 2)
0x01	0x02	MPEG-2 video, 16:9 aspect ratio with pan vectors, 25 Hz (see note 2)
0x01	0x03	MPEG-2 video, 16:9 aspect ratio without pan vectors, 25 Hz (see note 2)
0x01	0x04	MPEG-2 video, > 16:9 aspect ratio, 25 Hz (see note 2)
0x01	0x05	MPEG-2 video, 4:3 aspect ratio, 30 Hz (see note 2)
0x01	0x06	MPEG-2 video, 16:9 aspect ratio with pan vectors, 30 Hz (see note 2)
0x01	0x07	MPEG-2 video, 16:9 aspect ratio without pan vectors, 30 Hz (see note 2)
0x01	0x08	MPEG-2 video, > 16:9 aspect ratio, 30 Hz (see note 2)
0x01	0x09	MPEG-2 high definition video, 4:3 aspect ratio, 25 Hz (see note 2)
0x01	0x0A	MPEG-2 high definition video, 16:9 aspect ratio with pan vectors, 25 Hz (see note 2)
0x01	0x0B	MPEG-2 high definition video, 16:9 aspect ratio without pan vectors, 25 Hz (see note 2)
0x01	0x0C	MPEG-2 high definition video, > 16:9 aspect ratio, 25 Hz (see note 2)
0x01	0x0D	MPEG-2 high definition video, 4:3 aspect ratio, 30 Hz (see note 2)
0x01	0x0E	MPEG-2 high definition video, 16:9 aspect ratio with pan vectors, 30 Hz (see note 2)
0x01	0x0F	MPEG-2 high definition video, 16:9 aspect ratio without pan vectors, 30 Hz (see note 2)
0x01	0x10	MPEG-2 high definition video, > 16:9 aspect ratio, 30 Hz (see note 2)
0x01	0x11 to 0xAF	reserved for future use

Stream_content	Component_type	Description
0x01	0xB0 to 0xFE	user defined
0x01	0xFF	reserved for future use
0x02	0x00	reserved for future use
0x02	0x01	MPEG-1 Layer 2 audio, single mono channel
0x02	0x02	MPEG-1 Layer 2 audio, dual mono channel
0x02	0x03	MPEG-1 Layer 2 audio, stereo (2 channel)
0x02	0x04	MPEG-1 Layer 2 audio, multi-lingual, multi-channel
0x02	0x05	MPEG-1 Layer 2 audio, surround sound
0x02	0x06 to 0x3F	reserved for future use
0x02	0x40	MPEG-1 Layer 2 audio description for the visually impaired
0x02	0x41	MPEG-1 Layer 2 audio for the hard of hearing
0x02	0x42	receiver-mixed supplementary audio as per annex E of TS 101 154 [9]
0x02	0x43 to 0x46	reserved for future use
0x02	0x47	MPEG-1 Layer 2 audio, receiver mix audio description as per annex E of TS 101 154 [9]
0x02	0x48	MPEG-1 Layer 2 audio, broadcaster mix audio description
0x02	0x49 to 0xAF	reserved for future use
0x02	0xB0 to 0xFE	user-defined
0x02	0xFF	reserved for future use
0x03	0x00	reserved for future use
0x03	0x01	EBU Teletext subtitles
0x03	0x02	associated EBU Teletext
0x03	0x03	VBI data
0x03	0x04 to 0x0F	reserved for future use
0x03	0x10	DVB subtitles (normal) with no monitor aspect ratio criticality
0x03	0x11	DVB subtitles (normal) for display on 4:3 aspect ratio monitor
0x03	0x12	DVB subtitles (normal) for display on 16:9 aspect ratio monitor
0x03	0x13	DVB subtitles (normal) for display on 2.21:1 aspect ratio monitor
0x03	0x14	DVB subtitles (normal) for display on a high definition monitor
0x03	0x15 to 0x1F	reserved for future use
0x03	0x20	DVB subtitles (for the hard of hearing) with no monitor aspect ratio criticality
0x03	0x21	DVB subtitles (for the hard of hearing) for display on 4:3 aspect ratio monitor
0x03	0x22	DVB subtitles (for the hard of hearing) for display on 16:9 aspect ratio monitor
0x03	0x23	DVB subtitles (for the hard of hearing) for display on 2.21:1 aspect ratio monitor
0x03	0x24	DVB subtitles (for the hard of hearing) for display on a high definition monitor
0x03	0x25 to 0x2F	reserved for future use
0x03	0x30	Open (in-vision) sign language interpretation for the deaf
0x03	0x31	Closed sign language interpretation for the deaf
0x03	0x32 to 0x3F	reserved for future use
0x03	0x40	video up-sampled from standard definition source material
0x03	0x41 to 0xAF	reserved for future use
0x03	0xB0 to 0xFE	user defined
0x03	0xFF	reserved for future use
0x04	0x00 to 0x7F	reserved for AC-3 audio modes (refer to table D.1)
0x04	0x80 to 0xFF	reserved for enhanced AC-3 audio modes (refer to table D.1)
0x05	0x00	reserved for future use
0x05	0x01	H.264/AVC standard definition video, 4:3 aspect ratio, 25 Hz (see note 2)
0x05	0x02	reserved for future use
0x05	0x03	H.264/AVC standard definition video, 16:9 aspect ratio, 25 Hz (see note 2)
0x05	0x04	H.264/AVC standard definition video, > 16:9 aspect ratio, 25 Hz (see note 2)
0x05	0x05	H.264/AVC standard definition video, 4:3 aspect ratio, 30 Hz (see note 2)
0x05	0x06	reserved for future use
0x05	0x07	H.264/AVC standard definition video, 16:9 aspect ratio, 30 Hz (see note 2)
0x05	0x08	H.264/AVC standard definition video, > 16:9 aspect ratio, 30 Hz (see note 2)
0x05	0x09 to 0x0A	reserved for future use
0x05	0x0B	H.264/AVC high definition video, 16:9 aspect ratio, 25 Hz (see note 2)
0x05	0x0C	H.264/AVC high definition video, > 16:9 aspect ratio, 25 Hz (see note 2)
0x05	0x0D to 0x0E	reserved for future use
0x05	0x0F	H.264/AVC high definition video, 16:9 aspect ratio, 30 Hz (see note 2)
0x05	0x10	H.264/AVC high definition video, > 16:9 aspect ratio, 30 Hz (see note 2)
0x05	0x11 to 0xAF	reserved for future use
0x05	0xB0 to 0xFE	user-defined
0x05	0xFF	reserved for future use
0x06	0x00	reserved for future use

Stream_content	Component_type	Description
0x06	0x01	*HE-AAC* audio, single mono channel
0x06	0x02	reserved for future use
0x06	0x03	*HE-AAC* audio, stereo
0x06	0x04	reserved for future use
0x06	0x05	*HE-AAC* audio, surround sound
0x06	0x06 to 0x3F	reserved for future use
0x06	0x40	*HE-AAC* audio description for the visually impaired
0x06	0x41	*HE-AAC* audio for the hard of hearing
0x06	0x42	*HE-AAC* receiver-mixed supplementary audio as per annex E of TS 101 154 [9]
0x06	0x43	*HE-AAC* v2 audio, stereo
0x06	0x44	*HE-AAC* v2 audio description for the visually impaired
0x06	0x45	*HE-AAC* v2 audio for the hard of hearing
0x06	0x46	*HE-AAC* v2 receiver-mixed supplementary audio as per annex E of TS 101 154 [9]
0x06	0x47	*HE-AAC* receiver mix audio description for the visually impaired
0x06	0x48	*HE-AAC* broadcaster mix audio description for the visually impaired
0x06	0x49	*HE-AAC* v2 receiver mix audio description for the visually impaired
0x06	0x4A	*HE-AAC* v2 broadcaster mix audio description for the visually impaired
0x06	0x4B to 0xAF	reserved for future use
0x06	0xB0 to 0xFE	user-defined
0x06	0xFF	reserved for future use
0x07	0x00 to 0x7F	reserved for DTS audio modes (refer to annex G)
0x07	0x80 to 0xFF	reserved for future use
0x08	0x00	reserved for future use
0x08	0x01	DVB SRM data [48]
0x08	0x02 to 0xFF	reserved for DVB CPCM modes [46] to [i.4]
0x09 to 0x0B	0x00 to 0xFF	reserved for future use
0x0C to 0x0F	0x00 to 0xFF	user defined
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", "25 Hz" and "30 Hz" are used as defined in TS 101 154 [9] clauses 5.1 to 5.4 for MPEG-2 and clauses 5.5 to 5.7 for H.264/AVC and clauses 5.8 to 5.11 for VC-1 respectively.		

6.2.9 Content descriptor

The intention of the content descriptor (see table 27) is to provide classification information for an event.

Table 27: 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 28.

content_nibble_level_2: This 4-bit field represents the second level of a content identifier. This field shall be coded according to table 28.

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

Table 28: 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)
0x1	0x1	detective/thriller
0x1	0x2	adventure/western/war
0x1	0x3	science fiction/fantasy/horror
0x1	0x4	comedy
0x1	0x5	soap/melodrama/folkloric
0x1	0x6	romance
0x1	0x7	serious/classical/religious/historical movie/drama
0x1	0x8	adult movie/drama
0x1	0x9 to 0xE	reserved for future use
0x1	0xF	user defined
		News/Current affairs:
0x2	0x0	news/current affairs (general)
0x2	0x1	news/weather report
0x2	0x2	news magazine
0x2	0x3	documentary
0x2	0x4	discussion/interview/debate
0x2	0x5 to 0xE	reserved for future use
0x2	0xF	user defined
		Show/Game show:
0x3	0x0	show/game show (general)
0x3	0x1	game show/quiz/contest
0x3	0x2	variety show
0x3	0x3	talk show
0x3	0x4 to 0xE	reserved for future use
0x3	0xF	user defined
		Sports:
0x4	0x0	sports (general)
0x4	0x1	special events (Olympic Games, World Cup, etc.)
0x4	0x2	sports magazines
0x4	0x3	football/soccer
0x4	0x4	tennis/squash
0x4	0x5	team sports (excluding football)
0x4	0x6	athletics
0x4	0x7	motor sport
0x4	0x8	water sport
0x4	0x9	winter sports
0x4	0xA	equestrian
0x4	0xB	martial sports
0x4	0xC to 0xE	reserved for future use
0x4	0xF	user defined
		Children's/Youth programmes:
0x5	0x0	children's/youth programmes (general)
0x5	0x1	pre-school children's programmes
0x5	0x2	entertainment programmes for 6 to14
0x5	0x3	entertainment programmes for 10 to 16
0x5	0x4	informational/educational/school programmes
0x5	0x5	cartoons/puppets
0x5	0x6 to 0xE	reserved for future use
0x5	0xF	user defined

Content_nibble_level_1	Content_nibble_level_2	Description
		Music/Ballet/Dance:
0x6	0x0	music/ballet/dance (general)
0x6	0x1	rock/pop
0x6	0x2	serious music/classical music
0x6	0x3	folk/traditional music
0x6	0x4	jazz
0x6	0x5	musical/opera
0x6	0x6	ballet
0x6	0x7 to 0xE	reserved for future use
0x6	0xF	user defined
		Arts/Culture (without music):
0x7	0x0	arts/culture (without music, general)
0x7	0x1	performing arts
0x7	0x2	fine arts
0x7	0x3	religion
0x7	0x4	popular culture/traditional arts
0x7	0x5	literature
0x7	0x6	film/cinema
0x7	0x7	experimental film/video
0x7	0x8	broadcasting/press
0x7	0x9	new media
0x7	0xA	arts/culture magazines
0x7	0xB	fashion
0x7	0xC to 0xE	reserved for future use
0x7	0xF	user defined
		Social/Political issues/Economics:
0x8	0x0	social/political issues/economics (general)
0x8	0x1	magazines/reports/documentary
0x8	0x2	economics/social advisory
0x8	0x3	remarkable people
0x8	0x4 to 0xE	reserved for future use
0x8	0xF	user defined
		Education/Science/Factual topics:
0x9	0x0	education/science/factual topics (general)
0x9	0x1	nature/animals/environment
0x9	0x2	technology/natural sciences
0x9	0x3	medicine/physiology/psychology
0x9	0x4	foreign countries/expeditions
0x9	0x5	social/spiritual sciences
0x9	0x6	further education
0x9	0x7	languages
0x9	0x8 to 0xE	reserved for future use
0x9	0xF	user defined
		Leisure hobbies:
0xA	0x0	leisure hobbies (general)
0xA	0x1	tourism/travel
0xA	0x2	handicraft
0xA	0x3	motoring
0xA	0x4	fitness and health
0xA	0x5	cooking
0xA	0x6	advertisement/shopping
0xA	0x7	gardening
0xA	0x8 to 0xE	reserved for future use
0xA	0xF	user defined

Content_nibble_level_1	Content_nibble_level_2	Description
		Special characteristics:
0xB	0x0	original language
0xB	0x1	black and white
0xB	0x2	unpublished
0xB	0x3	live broadcast
0xB	0x4 to 0xE	reserved for future use
0xB	0xF	user defined
0xC to 0xE	0x0 to 0xF	reserved for future use
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 29).

Table 29: Country availability descriptor

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

Semantics for the country availability descriptor:

c**ountry_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 using the 3-character code as specified in ISO 3166 [41]. Each character is coded into 8-bits according to ISO/IEC 8859-1 [23] and inserted in order into the 24-bit field. In the case that the 3 characters represent a number in the range 900 to 999, then country_code specifies an ETSI defined group of countries. These allocations are found in TS 101 162 [i.1].

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 30).

Table 30: Data broadcast descriptor

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

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. Allocations of the value of this field are found in TS 101 162 [i.1].

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 [42] 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 [23] 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 31). 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 31: Data broadcast id descriptor

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

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. Allocations of the value of this field are found in TS 101 162 [i.1].

id_selector_byte: For the purpose of application selection the id_selector_byte(s) might be used. The definition of the id_selector_byte(s) 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 32.

Table 32: Cable delivery system descriptor

Syntax	Number of bits	Identifier
cable_delivery_system_descriptor(){		
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 33.

Table 33: 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 34.

Table 34: 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 35.

Table 35: 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 36.

Table 36: 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 {		
"00"	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 37).

Table 37: 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.

Table 38: Roll-off factor

roll-off	Description
00	α = 0,35
01	α = 0,25
10	α = 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 39.

If DVB-S2 is used in non backwards compatible broadcast services mode (NBC-BS [8]), use of this descriptor is sufficient. If DVB-S2 is used in other modes, the S2_satellite_delivery_system_descriptor shall be used additionally (see clause 6.2.13.3).

Table 39: 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 40.

Table 40: 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 35.

6.2.13.3 S2 satellite delivery system descriptor

This descriptor is only required if DVB-S2 is not used in non backwards compatible broadcast services mode (NBC-BS [8]). In non backwards compatible broadcast services mode the satellite_delivery_system_descriptor is sufficient.

Table 41: 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
backwards_compatibility_indicator	1	bslbf
reserved_future_use	5	bslbf
if (scrambling_sequence_selector == 1){		
Reserved	6	bslbf
scrambling_sequence_index	18	uimsbf
}		
if (multiple_input_stream_flag == 1){		
input_stream_identifier	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.

backwards_compatibility_indicator: This 1-bit field shall be set according to EN 302 307 [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 [8].

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 [8].

6.2.13.4 Terrestrial delivery system descriptor

See table 42.

NOTE: The delivery system descriptor for DVB-T2 transmissions is specified in clause 6.4.4.1.

Table 42: 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.

Table 43: 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. In case the hierachy_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 44: 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 45.

Table 45: 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. 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 46: Signalling format for the α values and the used interleaver

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

code_rate: The code_rate is a 3-bit field specifying the inner FEC scheme used according to table 47. Non-hierarchical channel coding and modulation requires signalling of one code rate. In this case, 3 bits specifying code_rate according to table 47 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 47: 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 48.

Table 48: 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.

Table 49: 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 shall be present in the TSDT.

Table 50: 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 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 51: Extended event descriptor

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

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 [42]. 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 [23] 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 used to extend the 8-bit namespace of the descriptor_tag field.

Table 52: 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 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 53: 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 54.

Table 54: 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

The FTA content management descriptor 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 55: FTA content management descriptor

Syntax	Number of bits	Identifier
FTA_content_management_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved_future_use	4	
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.

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 56.

Table 56: 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 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 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 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 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

The linkage descriptor (see table 57) 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".

A service to which a mobile receiver might hand-over to can also be identified using 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.

Two events can be signalled as equivalent using the linkage_descriptor with linkage_type set to event linkage. 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.

Table 57: Linkage descriptor

Syntax	Number of bits	Identifier
linkage_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
service_id	16	uimsbf
linkage_type	8	uimsbf
if (linkage_type ==0x08){		
hand-over_type	4	bslbf
reserved_future_use	3	bslbf
origin_type	1	bslbf
if (hand-over_type ==0x01		
|| hand-over_type ==0x02		
|| hand-over_type ==0x03){		
network_id	16	uimsbf
}		
if (origin_type ==0x00){		
initial_service_id	16	uimsbf
}		
}		
if (linkage_type == 0x0D){		
target_event_id	16	uimsbf
target_listed	1	bslbf
event_simulcast	1	bslbf
reserved	6	bslbf
}		
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 58).

Table 58: 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 0x7F	reserved for future use
0x80 to 0xFE	user defined
0xFF	reserved for future use
NOTE: A linkage_type with value 0x0D is only valid when the descriptor is carried in the EIT.	

hand-over_type: This is a 4-bit field specifying the type of hand-over (see table 59).

Table 59: 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 60).

Table 60: 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.

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 (advanced codec HD digital television service) if the events are simulcast, otherwise the service_type shall be the same as for the service wherer 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.

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

6.2.20 Local time offset descriptor

The local time offset descriptor (see table 61) may be used in the TOT to describe country specific dynamic changes of the local time offset relative to UTC.

Table 61: Local time offset descriptor

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

Semantics for the local time offset descriptor:

country_code: This 24-bit field identifies a country using the 3-character code as specified in ISO 3166 [41]. Each character is coded into 8-bits according to ISO/IEC 8859-1 [23] and inserted in order into the 24-bit field. In the case of that the 3 characters represent a number in the range of 900 to 999, then country code specifies an ETSI defined group of countries. These allocations are in TS 101 162 [i.1]. 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 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 62: 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 in the range between 0 hours and 13 hours 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 in the range between 0 hours and 13 hours 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 63.

Table 63: 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 64 for coding.

Table 64: 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 65 for coding.

Table 65: 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 66.

Table 66: 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 67 for coding.

Table 67: 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 68) provides the bouquet name in text form in one or more languages.

Table 68: 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 [42] 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 [23] 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 69) provides a text description of a component in one or more languages. The component is identified by its component tag value.

Table 69: 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 [42]. 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 [23] 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 70) provides the network name in text form in one or more languages.

Table 70: 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
for (j=0;j<N;j++){		
char	8	uimsbf
}		
}		
}		

Semantics for the multilingual network name descriptor:

ISO_639_language_code: This 24-bit field contains the ISO 639-2 [42] 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 [23] 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 71) provides the names of the service provider and service in text form in one or more languages.

Table 71: Multilingual service name descriptor

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

Semantics for the multilingual service name descriptor:

ISO_639_language_code: This 24-bit field contains the ISO 639-2 [42] 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 [23] 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 72) 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 72: 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 73).

Table 73: 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 74) gives a rating based on age and allows for extensions based on other rating criteria.

Table 74: 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 using the 3-character code as specified in ISO 3166 [41]. Each character is coded into 8-bits according to ISO/IEC 8859-1 [23] and inserted in order into the 24-bit field. In the case that the 3 characters represent a number in the range 900 to 999, then country_code specifies an ETSI defined group of countries. These allocations are found in TS 101 162 [i.1].

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 75, giving the recommended minimum age in years of the end user.

Table 75: 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 7.2.1.

6.2.30 PDC descriptor

The PDC-descriptor 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 76: 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 is used to identify the specifier of any private descriptors or private fields within descriptors.

Table 77: Private data specifier descriptor

Syntax	Number of bits	Identifier
private_data_specifier_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
private_data_specifier	32	uimsbf
}		

Semantics for the private data specifier descriptor:

private_data_specifier: The assignment of values for this field is given in TS 101 162 [i.1].

6.2.32 Scrambling descriptor

The scrambling descriptor 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 78: Scrambling_descriptor

Syntax	Number of bits	Identifier
scrambling_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
scrambling_mode	8	uimsbf
}		

Semantics for the scrambling_descriptor:

scrambling_mode: This 8-bit field identifies the selected mode of the scrambling algorithm (see table 79). 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 79: 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 in standard mode.
0x04	This value indicates use of DVB-CSA3 in minimally enhanced mode.
0x05	This value indicates use of DVB-CSA3 in fully enhanced mode.
0x06 to 0x6F	Reserved for future use
0x70 to 0x7F	ATIS defined (ATIS-0800006, see annex J)
0x80 to 0xFE	User defined
0xFF	Reserved for future use

6.2.33 Service descriptor

The service descriptor (see table 80) provides the names of the service provider and the service in text form together with the service_type.

Table 80: 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 81.

Table 81: 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 [48]
0x09	reserved for future use
0x0A	advanced codec digital radio sound service
0x0B	advanced codec mosaic service
0x0C	data broadcast service
0x0D	reserved for Common Interface Usage (EN 50221 [37])
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	advanced codec SD digital television service
0x17	advanced codec SD NVOD time-shifted service
0x18	advanced codec SD NVOD reference service
0x19	advanced codec HD digital television service
0x1A	advanced codec HD NVOD time-shifted service
0x1B	advanced codec HD NVOD reference service
0x1C 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.	

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 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 82: Service availability descriptor

Syntax	Number of bits	Identifier
service_availbility_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
availability_flag	1	bslbf
reserved	7	bslbf
for (i=0;i<N;i++) {		
cell_id	16	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 cell(s) identified by the cell_id(s) in the following loop and not available on the other cell(s). When the bit is set to "0", it indicates that the service is unavailable on the cell(s) identified by the cell_id(s) in the following loop and available on the other cell(s).

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 83) provides a means of listing the services by service_id and service type.

Table 83: Service list descriptor

Syntax	Number of bits	Identifier
service_list_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for (i=0;i<N;I++){		
service_id	16	uimsbf
service_type	8	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 (NVOD 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 81.

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.

Table 84: Service move descriptor

Syntax	Number of bits	Identifier
service_move_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
new_original_network_id	16	uimsbf
new_transport_stream_id	16	uimsbf
new_service_id	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 85).

Table 85: Short event descriptor

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

Semantics for the short event descriptor:

ISO_639_language_code: This 24-bit field contains the ISO 639-2 [42] 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 [23] 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 [18] 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.

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 [18] (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 86: Short smoothing buffer descriptor

Syntax	Number of bits	Identifier
short_smoothing_buffer_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
sb_size	2	uimsbf
sb_leak_rate	6	uimsbf
for (i=0;i<N;i++){		
DVB_reserved	8	bslbf
}		
}		

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 87.

Table 87: 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 88.

Table 88: 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	
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	((value) - 16) × 0,5	(7,0 Mbit/s, 7,5 Mbit/s, 8,0 Mbit/s)
33 to 37	((value) - 24)	(9 Mbit/s, 10 Mbit/s, 11 Mbit/s, 12 Mbit/s, 13 Mbit/s)
38	13,5	
39 to 43	((value) - 25)	(14 Mbit/s, 15 Mbit/s, 16 Mbit/s, 17 Mbit/s, 18 Mbit/s)
44 to 47	((value) - 34) × 2	(20 Mbit/s, 22 Mbit/s, 24 Mbit/s, 26 Mbit/s)
48	27	
49 to 55	((value) - 35) × 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 89) 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 following the relevant ES_info_length field.

Table 89: Stream identifier descriptor

Syntax	Number of bits	Identifier
`stream_identifier_descriptor(){`		
`  descriptor_tag`	8	uimsbf
`  descriptor_length`	8	uimsbf
`  component_tag`	8	uimsbf
`}`		

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 90).

Table 90: 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 [18] Program Map Table (PMT) the value of stream_type for any PID carrying DVB subtitle shall be "0x06" (this indicates a PES carrying private data). See table 91.

Table 91: 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 [42] 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 [23] 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.6].

The telephone descriptor syntax is specified in table 92.

Table 92: Telephone descriptor

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

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 [23] 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 [23] 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 [23] 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 [23] 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 [23] gives one alphanumeric character of the core number.

6.2.43 Teletext descriptor

The Teletext descriptor (see table 93) shall be used in the PSI PMT to identify streams which carry EBU Teletext data. The descriptor is to be located in a program map section following the relevant ES_info_length field.

Table 93: 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 [42] 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 [23] 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 94.

Table 94: 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 95) is used in place of the short_event_descriptor to indicate an event which is a time shifted copy of another event.

Table 95: 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 96).

Table 96: 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, being transmitted in the TSDT (see ISO/IEC 13818-1 [18]) only, may be used to indicate the compliance of a transport stream with an MPEG based system, e.g. DVB.

Table 97: 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 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 98: 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 99. 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.

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.

Table 99: 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 0xFF	reserved for future use

6.2.48 VBI teletext descriptor

The VBI teletext descriptor shall be used in the PSI PMT to identify streams which carry VBI data as well as EBU Teletext data. The descriptor is to be located in a program map section following the relevant ES_info_length field.

Table 100: 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 101 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 101: Possible locations of extended descriptors

Descriptor	Tag extension value	NIT	BAT	SDT	EIT	TOT	PMT	SIT (see note)
image_icon_descriptor	0x00	*	*	*	*	-	-	*
cpcm_delivery_signalling_descriptor (TS/TR 102 825 [46],[i.4])	0x01	-	-	*	*	-	-	-
CP_descriptor (TS/TR 102 825 [46],[i.4])	0x02	-	-	-	-	-	*	-
CP_identifier_descriptor (TS/TR 102 825 [46],[i.4])	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	-	-	*	-	-	-	-
reserved for future use	0x0C to 0x7F							
user defined	0x80 to 0xFF							
NOTE: Only found in Partial Transport Streams.								

6.4 Extended descriptor coding

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 CP descriptor

The content protection descriptor 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 102: 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 indicating the type of CA system applicable. Allocations of the value of this field are found in TS 101 162 [i.1].

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

6.4.2 CP identifier descriptor

The CP identifier descriptor (see table 103) 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 system renewability messages). It identifies the CP system and the type of information by means of the CP_system_id.

Table 103: 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.). Allocations of the value of this field are found in TS 101 162 [i.1].

6.4.3 CPCM delivery signaling descriptor

The CPCM delivery signaling 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 [46].

6.4.4 Delivery system descriptors

6.4.4.1 SH delivery system descriptor

This descriptor is used to transmit the physical parameters for each DVB-SH [49] 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 104: SH delivery system descriptor

Syntax	Number of bits	Identifier
SH_delivery_system_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_tag_extension	8	uimsbf
descriptor_length	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 105.

Table 105: 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 106.

Table 106: modulation_type

modulation_type	Description
0	TDM
1	OFDM

interleaver_presence: This 1-bit flag indicates the use of an interleaver according to table 107.

Table 107: 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.

Table 108: 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.

Table 109: roll_off

roll_off	Description
00	α = 0,35
01	α = 0,25
10	α = 0,15
11	reserved for future use

modulation_mode: This 2-bit field indicates the chosen modulation.

Table 110: 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.

Table 111: 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	1/3 complementary
1010	2/3 standard
1011	2/3 complementary
1100 - 1111	reserved for future use

symbol_rate: This 5-bit field signals the TDM symbol rate expressed in Msymbols per second. 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 [50] where the OFDM parameters (bandwidth and guard interval) and the TDM roll-off determine a TDM symbol rate.

Table 112: 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 - 111111	reserved for future use				

bandwidth: This 3-bit fied gives the OFDM bandwidth.

Table 113: 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. Its semantics depend on the chosen constellation_and_hierarchy value.

Table 114: 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.

Table 115: 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 - 111	reserved for future use

code_rate: This 4-bit field indicates the current TS code rate according to table 111.

guard_interval: This 2-bits tag gives the chosen guard interval according to table 116.

Table 116: 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.

Table 117: 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.

6.4.4.2 T2 delivery system descriptor

The T2_delivery_system_descriptor (see table 104) shall be used in the TS loop of the Network Information Table to describe DVB-T2 transmissions according to EN 302 755 [47]. This descriptor maps Transport Streams to data Physical Layer Pipes (data PLPs) and T2 systems. The number of T2_delivery_system_descriptors per NIT equals the number of Transport Streams in the network carried over DVB-T2 The descriptor reflects a mapping of a TS - heading the NIT's TS loop - to a data PLP, whereby the T2_system_id 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 and 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 identifiy the Transport Stream that is carried by the PLP/T2 system pair identified with plp_id and T2_system_id in the descriptor.

NOTE: 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 118: 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
}		
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 uniquely identifies a data PLP within a T2 System, within a T2 Network. The term is defined in EN 302 755 [47].

T2_system_id: This 16-bit field uniquely identifies a T2 system within a T2 network. The term is defined in EN 302 755 [47].

The remaining part of this descriptor, immediately following the T2_system_id field, is only present 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 119.

Table 119: 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 120.

Table 120: 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 121.

Table 121: 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 122.

Table 122: 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 - 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.

Table 123: 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 [47].

NOTE: 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.

6.4.5 Image icon descriptor

The image icon descriptor 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 [13], 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.

The format for this descriptor is defined in table 124.

Table 124: 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 [13]).

icon_transport_mode: This field indicates the mode of delivery of the icon as defined in table 125.

Table 125: 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 126.

Table 126: Coordinate system

Value	Meaning
0x00	The coordinate system is 720 x 576
0x01	The coordinate system is 1 280 x 720
0x02	The coordinate system is 1 920 x 1 080
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 [51]. 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 [45], 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 [45], 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 [13]. The URL may be a DVB locator referencing a file in an object carousel, as specified in clause 6.4 of TS 102 323 [13] 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.

6.4.6 Message descriptor

This descriptor allows broadcasters to provide receivers with a textual message which the receiver may display to the user at appropriate times (not defined here).

Table: 127: 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 subtable 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 [42] 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 [23] 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.

6.4.7 Network change notify descriptor

This descriptor 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 128: 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++) {		uimsbf
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.

Table 129: 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 130. 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 "coverage change" category shall be used when power and/or modulation parameter changes may change the coverage of a transmitter. 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.

Table 130: 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.

6.4.8 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.

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 131: Service relocated descriptor

Syntax	Number of bits	Identifier
service_relocated_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
descriptor_tag_extension	8	uimbsf
old_original_network_id	16	uimsbf
old_transport_stream_id	16	uimsbf
old_service_id	16	uimsbf
}		

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.

6.4.9 Supplementary audio descriptor

The supplementary_audio descriptor provides additional information about the audio streams which 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.

Table 132: Supplementary audio descriptor

Syntax	Number of bits	Identifier
supplementary_audio_descriptor() {		
descriptor_tag	**8**	uimbsf
descriptor_length	**8**	uimbsf
descriptor_tag_extension	**8**	uimbsf
mix_type	**1**	uimbsf
editorial_classification	**5**	uimbsf
reserved	1	**uimbsf**
language_code_present	1	**uimbsf**
if (language_code_present == 1) {		
ISO_639_language_code	24	**bslbf**
}		
for (i=0;i<N;i++) {		
private_data_byte	8	**uimbsf**
}		
}		

Semantics for the supplemenrtary audio descriptor:

mix_type: This 1-bit field indicates whether the audio stream is a complete and independent stream or whether it is a supplementary stream intended to be mixed with another audio stream according to TS 101 154 [9], annex G.

Table 133: mix_type coding

mix_type	Description
0	Audio stream is a supplementary stream.
1	Audio stream is a complete and independent stream.

editorial_classification: This 5-bit field indicates the editorial intention of the audio stream.

Table 134: editorial_classification coding

editorial_classification	Description
00000	Main audio (contains all of the main audio components and can be presented on its own or mixed with a supplementary audio stream).
00001	Audio description for the visually impaired (contains a spoken description of the visual content of the service).
00010	Clean audio for the hearing impaired (the dialogue is concentrated on the centre channel).
00011	Spoken subtitles for the visually impaired (contains the main audio with a spoken rendition of the subtitles)
00100 to 10111	Reserved for future use.
11000 to 11111	User defined.

language_code_present: When this field is set to 1 (one), it indicates that the ISO_639_language_code field is present. When it is set to 0 (zero), the language of the audio stream shall be contained in the ISO_639_language descriptor in the same ES loop.

ISO_639_language_code: This 24 bit field identifies the language, as defined by ISO639, of the audio stream and overrides the value in the ISO_639_language descriptor in the same ES loop.

6.4.10 Target region descriptor

The target_region_descriptor identifies a set of target regions. The definition of a target region is given in clause 6.4.11. Each loop within a target_region_descriptor identifies a single region. The set of target regions identified by this descriptor is the union of the target regions identified in each loop.

The location of a particular instance of the target_region_descriptor defines the scope of the descriptor. The target_region_descriptor is a scoping descriptor (see clause 6.5), therefore the services covered by this descriptor are determined by the descriptor's location.

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:

a) in the NIT for all services within a network or all services within a transport stream of a network;

b) in the BAT for all services within a bouquet or all services within a transport stream of a bouquet;

c) 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. If the descriptor loop is empty then the target_region_descriptor identifies a single country.

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 135: 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	
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 the country using the 3-character code as specified in ISO 3166 [41]. Each character is coded into 8 bits according to ISO/IEC 8859-1 [23] and inserted in order into the 24-bit field. In the case that the 3 characters represent a number in the range of 900 to 999, then the country code specifies an ETSI defined group of countries. These allocations are defined in TS 101 162 [i.1].

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.

6.4.11 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 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.

Table 136: 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 using the 3-character code as specified in ISO 3166 [41]. Each character is coded into 8 bits according to ISO/IEC 8859-1 [23] and inserted in order into the 24-bit field. In the case that the 3 characters represent a number in the range of 900 to 999, then the country code specifies an ETSI defined group of countries. These allocations are defined in TS 101 162 [i.1].

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 [42]. 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 [23] 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.

6.5 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 property shall apply. These properties shall apply to all entities (eg. 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.

Only one instance of each type of scoping descriptor may be present in a descriptor loop.

In increasing precedence order, the descriptor loops where a Scoping Descriptor may appear, if allowed by the type of scoping descriptor, are:

a) NIT first loop (lowest precedence);

b) NIT TS loop;

c) BAT first loop;

d) BAT TS loop;

e) SDT;

f) EIT (highest precedence).

7 Storage Media Interoperability (SMI) measures

IEC 61883 [38] describes methods for delivering TS over the IEEE 1394.1 [39] 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

The SMI tables are encoded using the private section syntax defined in ISO/IEC 13818-1 [18]. 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 137) is to be inserted at transition points at which SI information may be discontinuous.

Table 137: Discontinuity information section

Syntax	Number of bits	Identifier
discontinuity_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transition_flag	1	uimsbf
reserved_future_use	7	bslbf
}		

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 service(s) and event(s) carried by the "partial" TS.

Table 138: 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 TR 101 211 [i.2].

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 B of ISO/IEC 13818-1 [18] after processing the entire section.

7.2 SMI descriptors

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 following descriptor is proposed to describe this information.

Table 139: 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 0x3FFFFF 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 0x3FFFFF is used to indicate that the maximum smoothing buffer size is undefined.

Annex A (normative): Coding of text characters

Text items can optionally include information to select a wide range of character tables as indicated below.

For the European languages a set of five character tables are available. 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 [16]

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 as follows:

Table A.3: Character coding tables

First byte value	Character code table	Table description	Reproduced in figure
0x01	ISO/IEC 8859-5 [27]	Latin/Cyrillic alphabet	A.2
0x02	ISO/IEC 8859-6 [28]	Latin/Arabic alphabet	A.3
0x03	ISO/IEC 8859-7 [29]	Latin/Greek alphabet	A.4
0x04	ISO/IEC 8859-8 [30]	Latin/Hebrew alphabet	A.5
0x05	ISO/IEC 8859-9 [31]	Latin alphabet No. 5	A.6
0x06	ISO/IEC 8859-10 [32]	Latin alphabet No. 6	A.7
0x07	ISO/IEC 8859-11 [33]	Latin/Thai (draft only)	A.8
0x08	reserved for future use (see note)		
0x09	ISO/IEC 8859-13 [34]	Latin alphabet No. 7	A.9
0x0A	ISO/IEC 8859-14 [35]	Latin alphabet No. 8 (Celtic)	A.10
0x0B	ISO/IEC 8859-15 [36]	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 [16]	Basic Multilingual Plane (BMP)	
0x12	KSX1001-2004 [44]	Korean Character Set	
0x13	GB-2312-1980	Simplified Chinese Character	
0x14	Big5 subset of ISO/IEC 10646 [16]	Traditional Chinese	
0x15	UTF-8 encoding of ISO/IEC 10646 [16]	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	

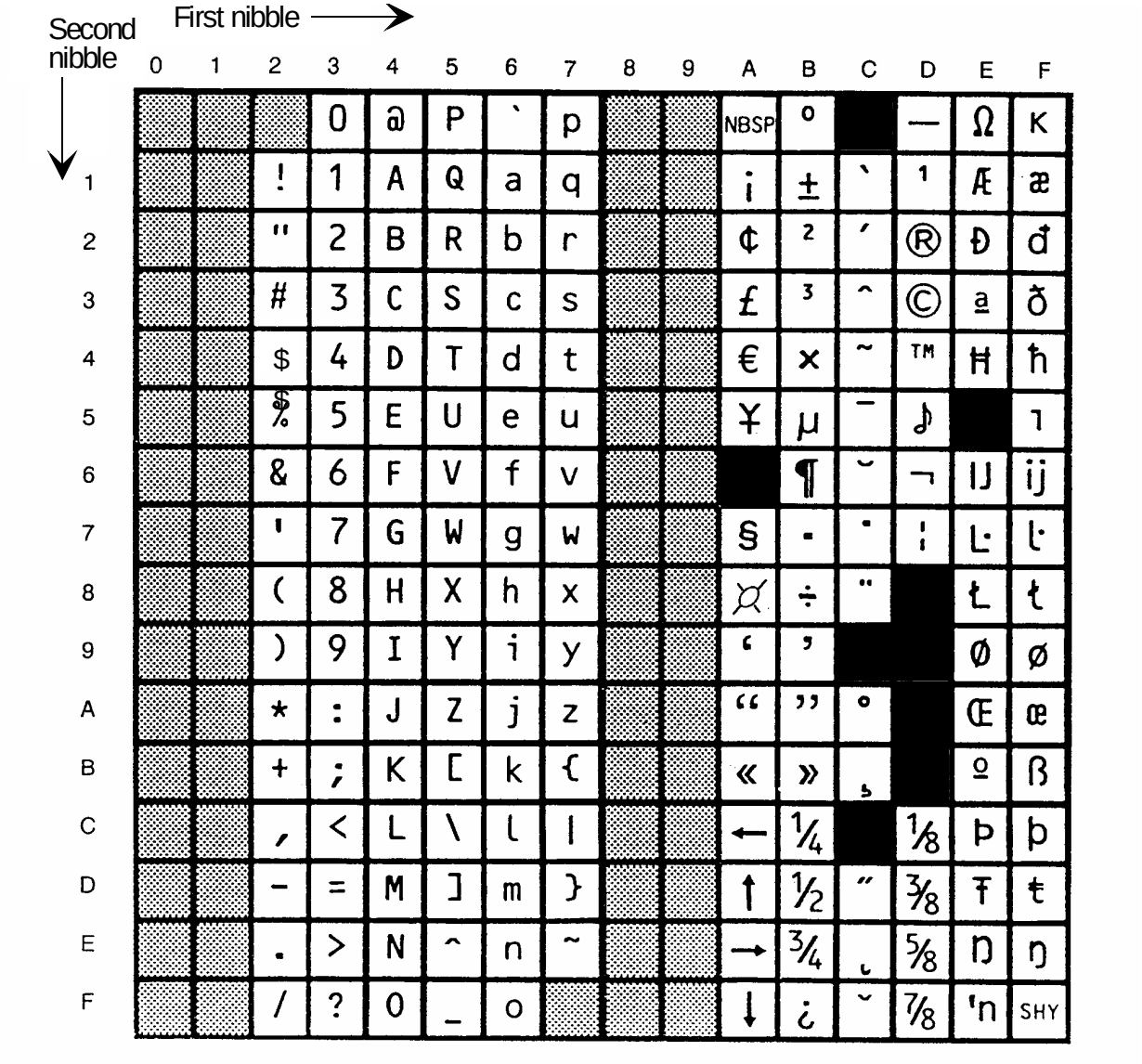
NOTE: The first byte value 0x08 was previously reserved for ISO/IEC 8859-12 which should have covered the Devanagri script. ISO/IEC however never published part 12 of the ISO/IEC 8859 series (see http://en.wikipedia.org/wiki/ISO/IEC_8859-12). Applications of the current document for the scripts used by South-Asian languages should use the BMP of ISO/IEC 10646 [16], 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 (uimsbf) containing the encoding_type_id. This value indicates the encoding scheme of the string. Allocations of the value of this field are found in TS 101 162 [i.1].

If the first byte of the text field has a value "0x10" then the following two bytes carry a 16-bit value (uimsbf) N to indicate that the remaining data of the text field is coded using the character code table specified in table A.4.

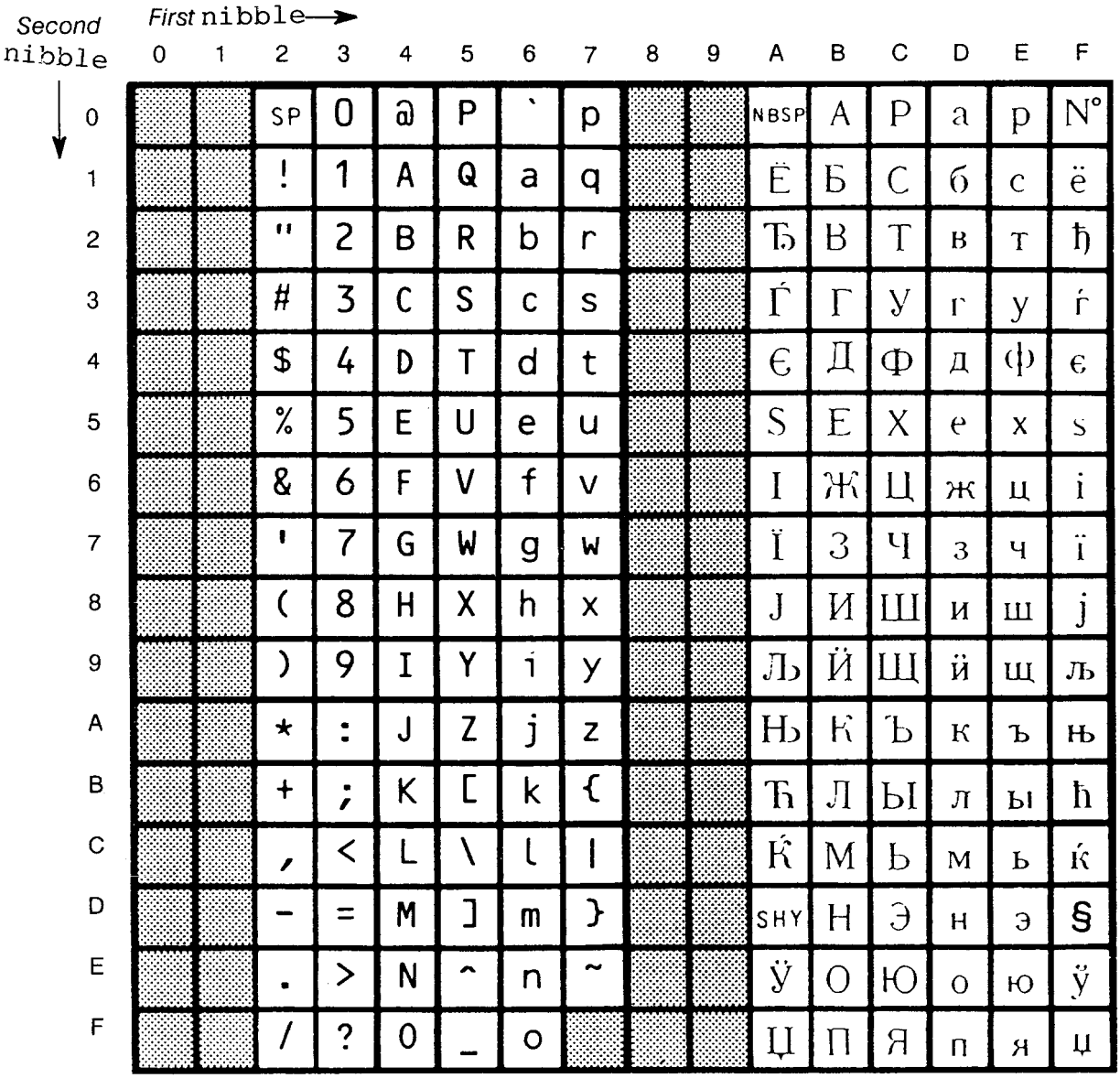
Table A.4: Character Coding Tables for first byte 0x10

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 [23]	West European	
0x10	0x00	0x02	ISO/IEC 8859-2 [24]	East European	
0x10	0x00	0x03	ISO/IEC 8859-3 [25]	South European	
0x10	0x00	0x04	ISO/IEC 8859-4 [26]	North and North-East European	
0x10	0x00	0x05	ISO/IEC 8859-5 [27]	Latin/Cyrillic	A.2
0x10	0x00	0x06	ISO/IEC 8859-6 [28]	Latin/Arabic	A.3
0x10	0x00	0x07	ISO/IEC 8859-7 [29]	Latin/Greek	A.4
0x10	0x00	0x08	ISO/IEC 8859-8 [30]	Latin/Hebrew	A.5
0x10	0x00	0x09	ISO/IEC 8859-9 [31]	West European & Turkish	A.6
0x10	0x00	0x0A	ISO/IEC 8859-10 [32]	North European	A.7
0x10	0x00	0x0B	ISO/IEC 8859-11 [33]	Thai	A.8
0x10	0x00	0x0C		Reserved for future use	
0x10	0x00	0x0D	ISO/IEC 8859-13 [34]	Baltic	A.9
0x10	0x00	0x0E	ISO/IEC 8859-14 [35]	Celtic	A.10
0x10	0x00	0x0F	ISO/IEC 8859-15 [36]	West European	A.11



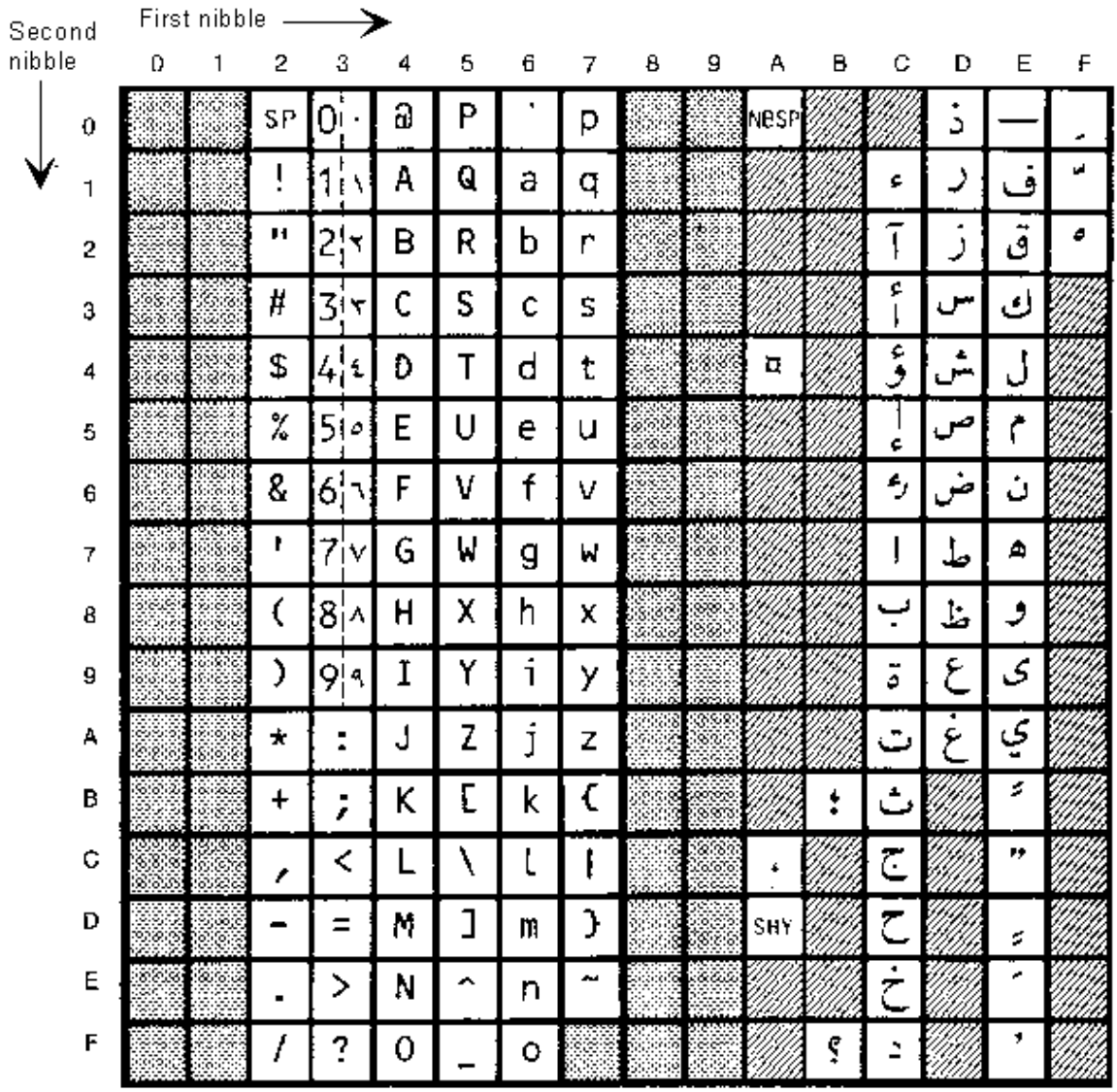
NOTE 1: The SPACE character is located in position 20h of the code table.
NOTE 2: NBSP = no-break space.
NOTE 3: SHY = soft hyphen.
NOTE 4: This table is a superset of ISO/IEC 6937 [22] with addition of the Euro symbol.
NOTE 5: All characters in column C are non-spacing characters (diacritical marks).

Figure A.1: Character code table 00 - Latin alphabet



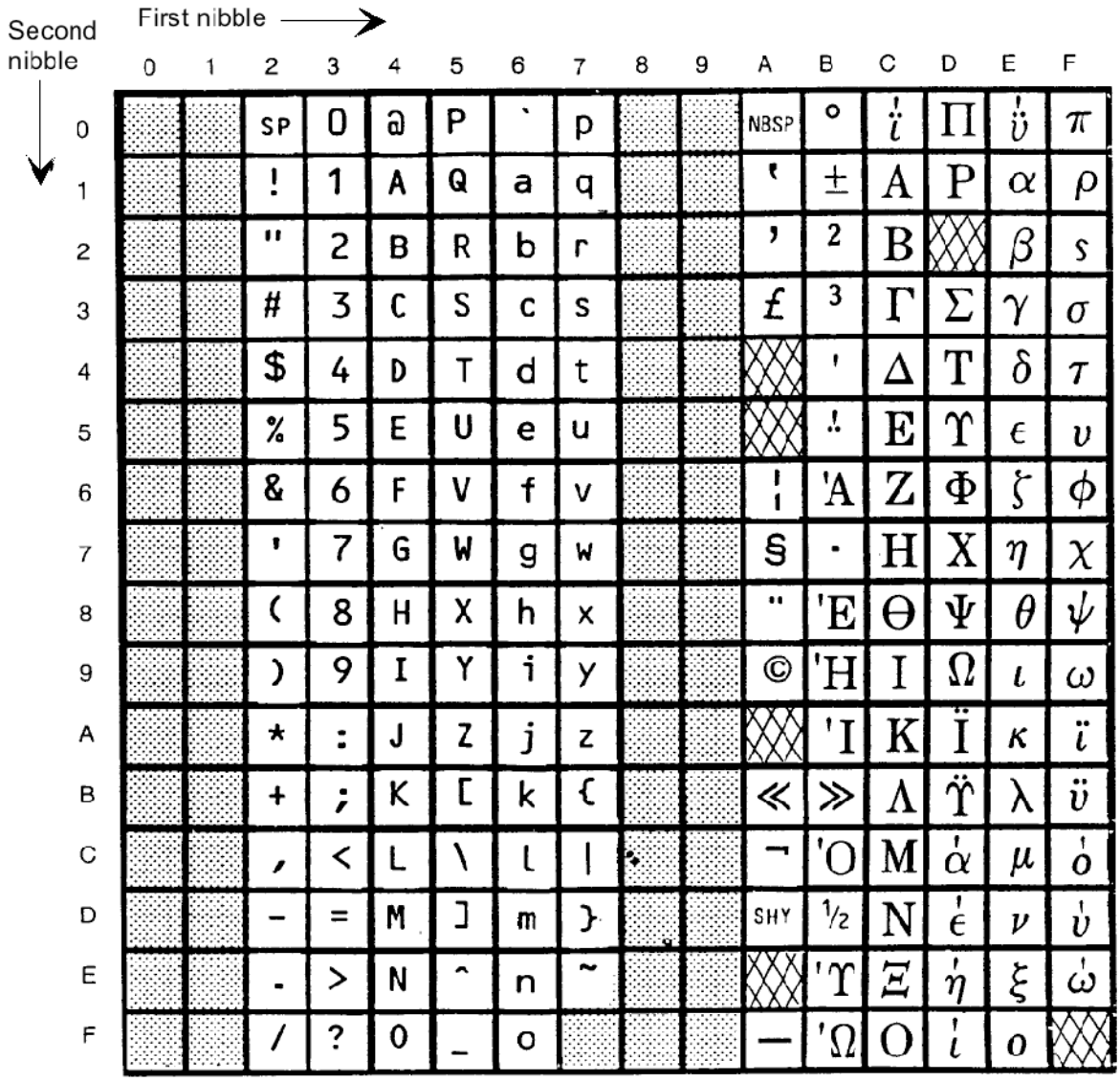
NOTE 1: For the Ruthenian language, the characters in code positions Ah/5h (S) and Fh/5h (s) are replaced by Г and Γ, respectively.
NOTE 2: Table reproduced from ISO/IEC 8859-5 [27].

Figure A.2: Character code table 01 - Latin/Cyrillic alphabet



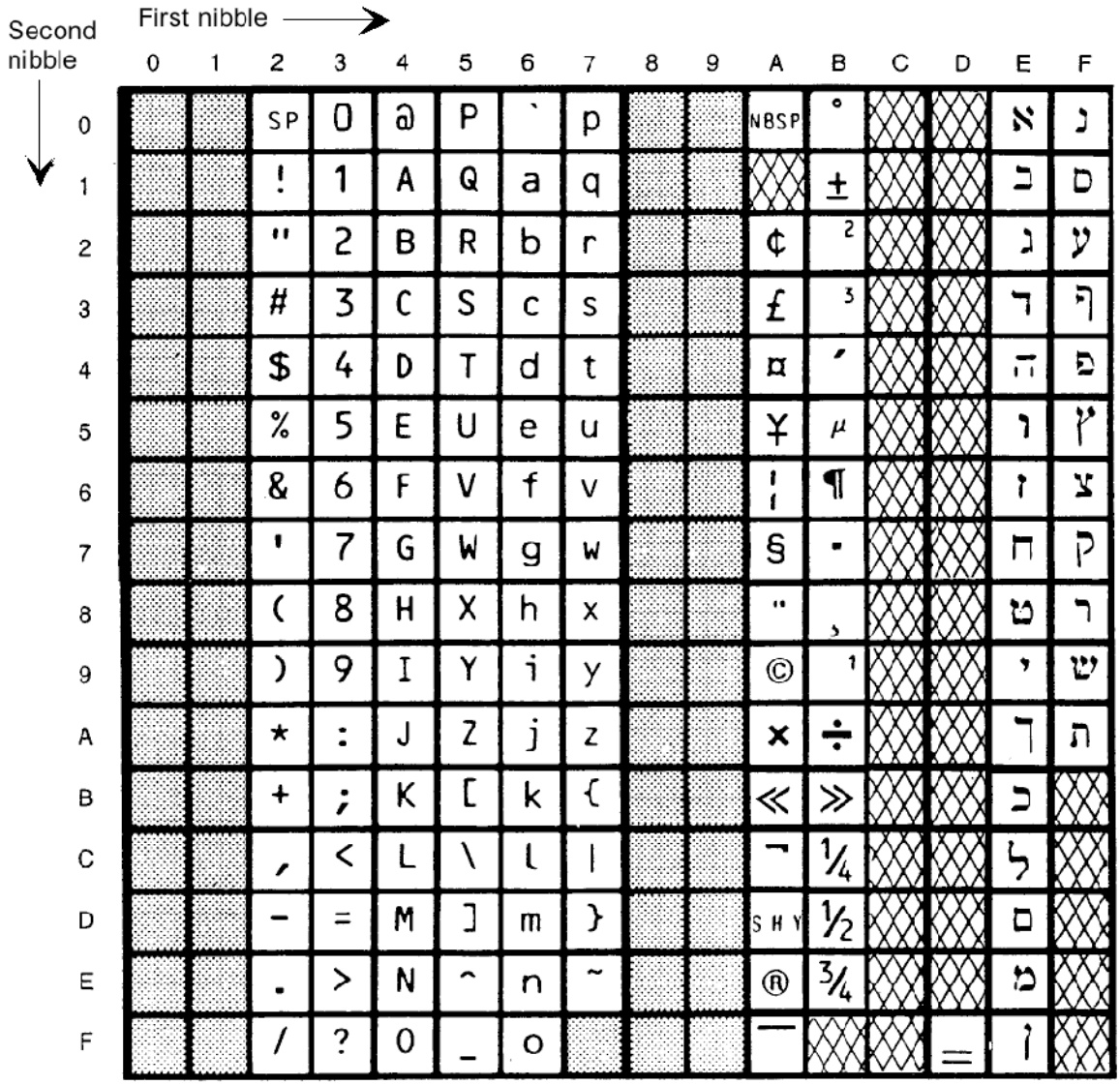
NOTE: Table reproduced from ISO/IEC 8859-6 [28].

Figure A.3: Character code table 02 - Latin/Arabic alphabet



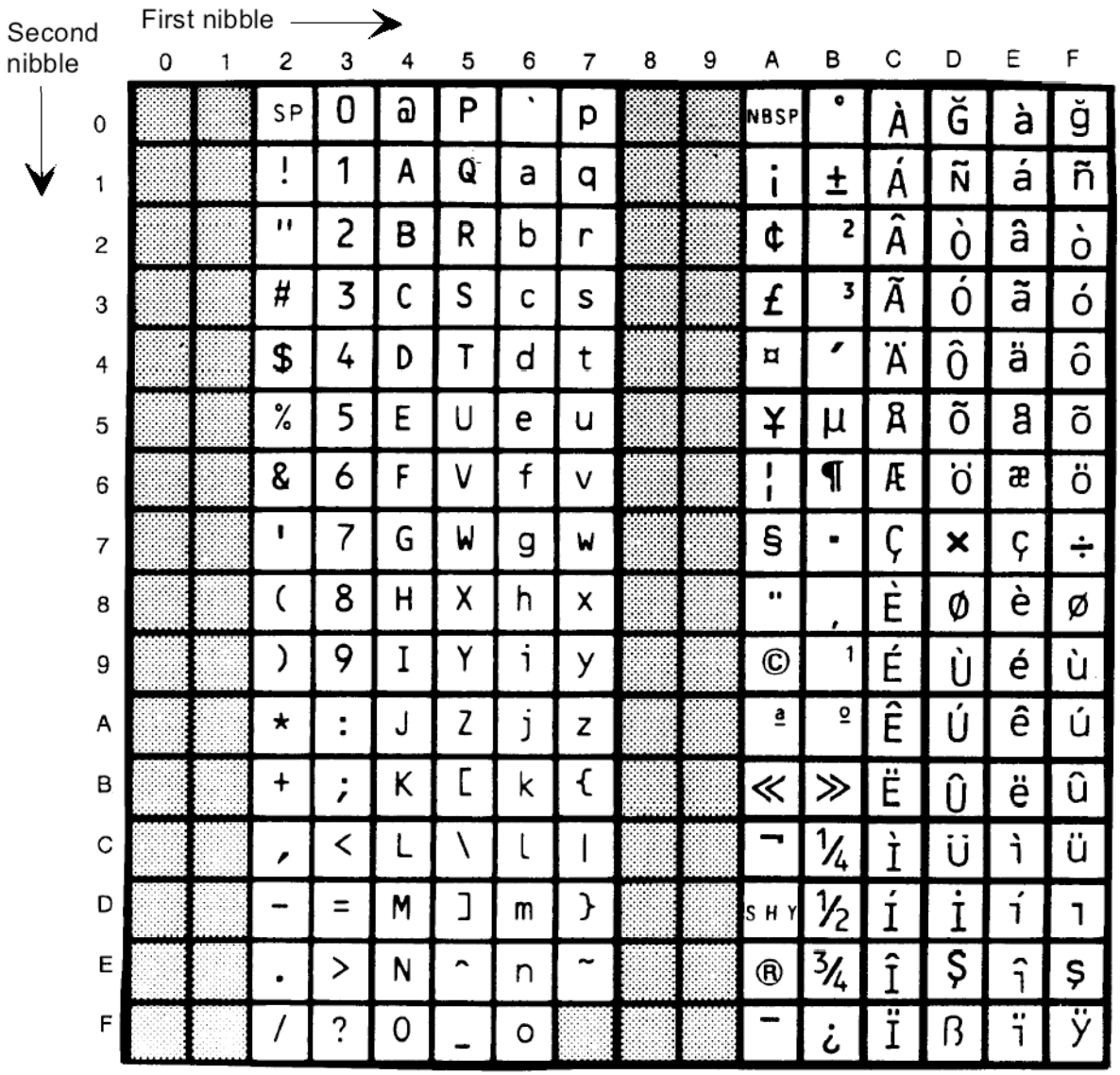
NOTE: Table reproduced from ISO/IEC 8859-7 [29].

Figure A.4: Character code table 03 - Latin/Greek alphabet



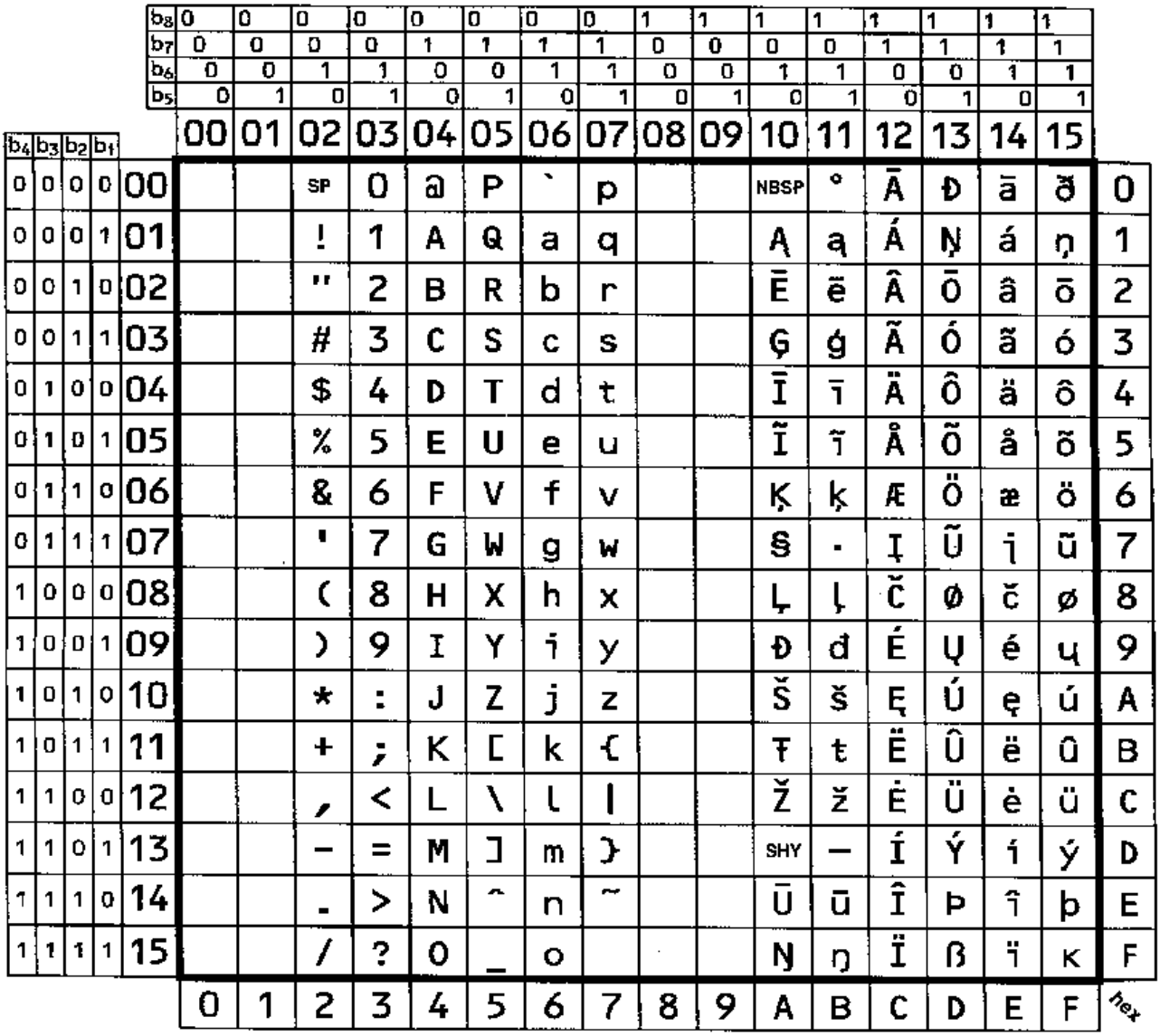
NOTE: Table reproduced from ISO/IEC 8859-8 [30].

Figure A.5: Character code table 04 - Latin/Hebrew alphabet



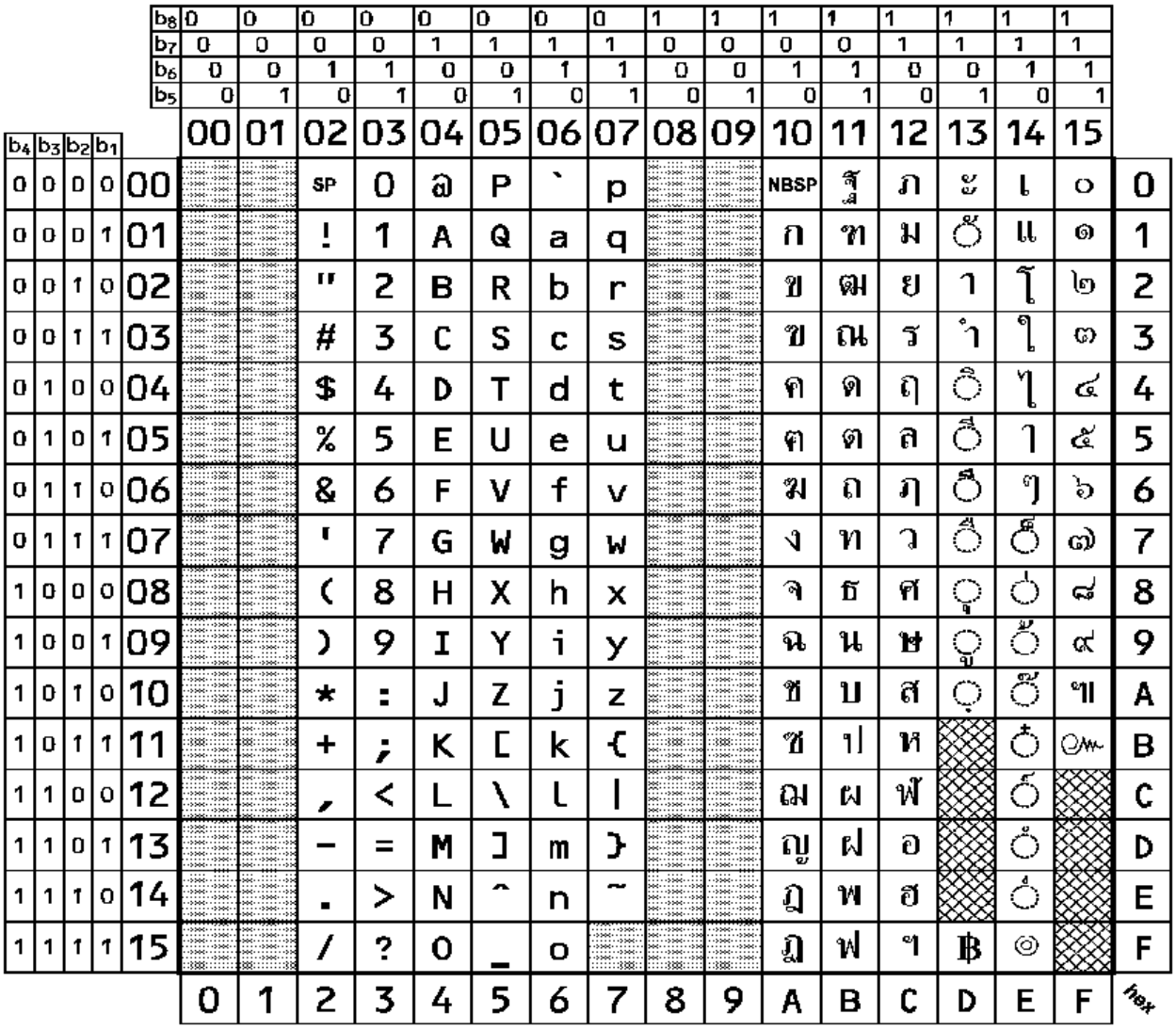
NOTE: Table reproduced from ISO/IEC 8859-9 [31].

Figure A.6: Character code table 05 - Latin alphabet number 5



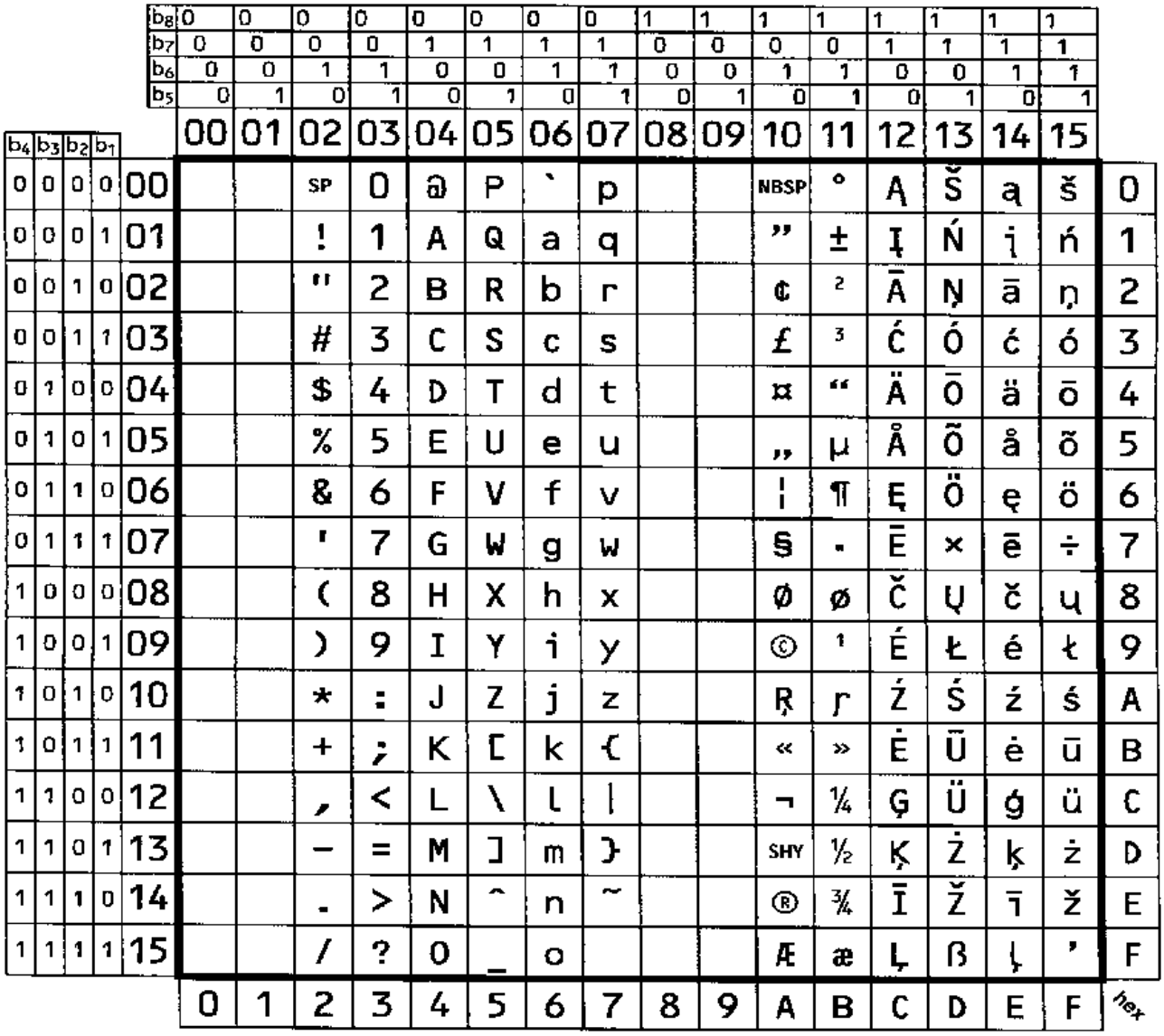
NOTE: Table reproduced from ISO/IEC 8859-10 [32].

Figure A.7: Character code table 06 - Latin alphabet number 6



NOTE: Table reproduced from ISO/IEC 8859-11 [33].

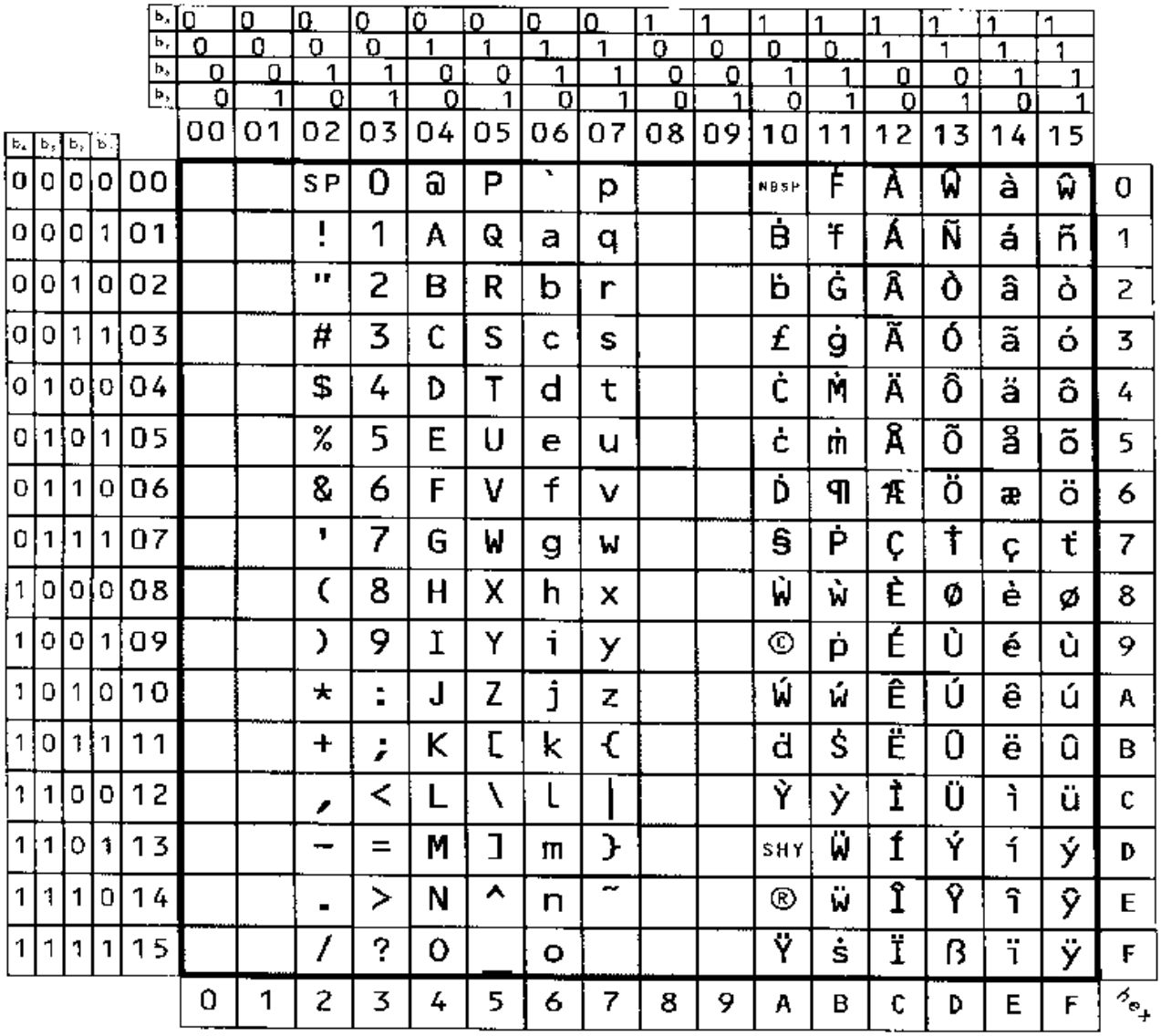
Figure A.8: Character code table 07 - Latin/Thai alphabet



b8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
b7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
b6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
b5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
b4 b3 b2 b1	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0 0 0 0 00			SP	0	@	P	`	p			NBSP	°	Ą	Š	ą	š	0
0 0 0 1 01			!	1	A	Q	a	q			”	±	Į	Ń	į	ń	1
0 0 1 0 02			"	2	B	R	b	r			¢	²	Ā	Ņ	ā	ņ	2
0 0 1 1 03			#	3	C	S	c	s			£	³	Ć	Ó	ć	ó	3
0 1 0 0 04			$	4	D	T	d	t			¤	“	Ä	Ō	ä	ō	4
0 1 0 1 05			%	5	E	U	e	u			„	µ	Å	Õ	å	õ	5
0 1 1 0 06			&	6	F	V	f	v			¦	¶	Ę	Ö	ę	ö	6
0 1 1 1 07			'	7	G	W	g	w			§	·	Ē	×	ē	÷	7
1 0 0 0 08			(	8	H	X	h	x			Ø	ø	Č	Ų	č	ų	8
1 0 0 1 09			)	9	I	Y	i	y			©	¹	É	Ł	é	ł	9
1 0 1 0 10			*	:	J	Z	j	z			Ŗ	ŗ	Ź	Ś	ź	ś	A
1 0 1 1 11			+	;	K	[	k	{			«	»	Ė	Ū	ė	ū	B
1 1 0 0 12			,	<	L	\	l	|			¬	¼	Ģ	Ü	ģ	ü	C
1 1 0 1 13			−	=	M	]	m	}			SHY	½	Ķ	Ż	ķ	ż	D
1 1 1 0 14			.	>	N	^	n	~			®	¾	Ī	Ž	ī	ž	E
1 1 1 1 15			/	?	O	_	o				Æ	æ	Ļ	ß	ļ	’	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	hex

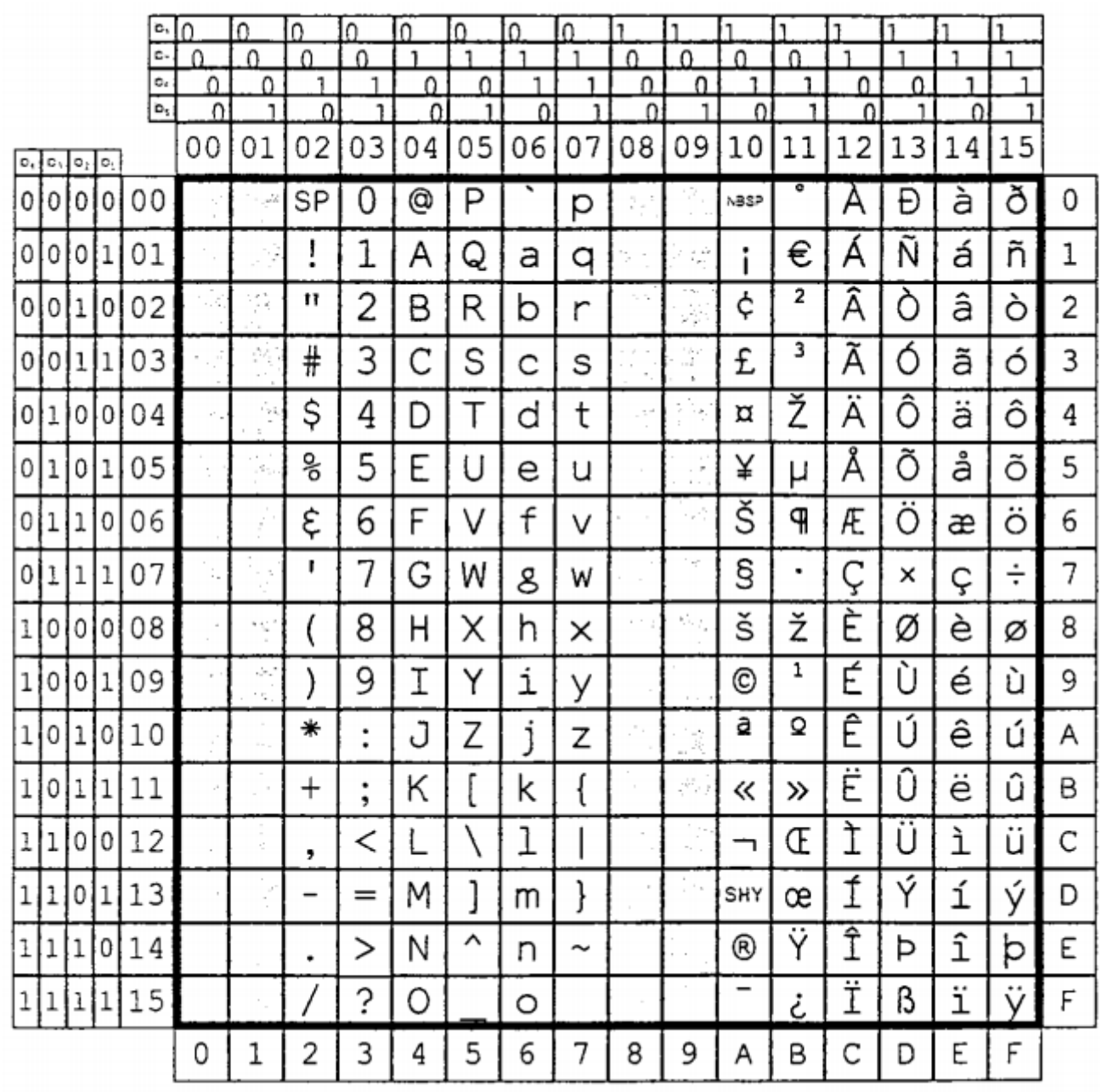
NOTE: Table reproduced from ISO/IEC 8859-13 [34].

Figure A.9: Character code table 09 - Latin alphabet number 7 (Baltic Rim)



NOTE: Table reproduced from ISO/IEC 8859-14 [35].

Figure A.10: Character code table 0A - Latin alphabet number 8 (Celtic)



b4 b3 b2 b1	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	
0 0 0 0			SP	0	@	P	`	p			NBSP	°	À	Ð	à	ð	0
0 0 0 1			!	1	A	Q	a	q			¡	€	Á	Ñ	á	ñ	1
0 0 1 0			"	2	B	R	b	r			¢	²	Â	Ò	â	ò	2
0 0 1 1			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó	3
0 1 0 0			$	4	D	T	d	t			¤	Ž	Ä	Ô	ä	ô	4
0 1 0 1			%	5	E	U	e	u			¥	µ	Å	Õ	å	õ	5
0 1 1 0			&	6	F	V	f	v			Š	¶	Æ	Ö	æ	ö	6
0 1 1 1			'	7	G	W	g	w			§	·	Ç	×	ç	÷	7
1 0 0 0			(	8	H	X	h	x			š	ž	È	Ø	è	ø	8
1 0 0 1			)	9	I	Y	i	y			©	¹	É	Ù	é	ù	9
1 0 1 0			*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú	A
1 0 1 1			+	;	K	[	k	{			«	»	Ë	Û	ë	û	B
1 1 0 0			,	<	L	\	l	|			¬	Œ	Ì	Ü	ì	ü	C
1 1 0 1			-	=	M	]	m	}			SHY	œ	Í	Ý	í	ý	D
1 1 1 0			.	>	N	^	n	~			®	Ÿ	Î	Þ	î	þ	E
1 1 1 1			/	?	O	_	o				¯	¿	Ï	ß	ï	ÿ	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

NOTE: Table reproduced from ISO/IEC 8859-15 [36].

Figure A.11: Character code table 0B - Latin alphabet number 9

Annex B (normative): CRC decoder model

The 32-bit CRC decoder is specified in figure B.1.

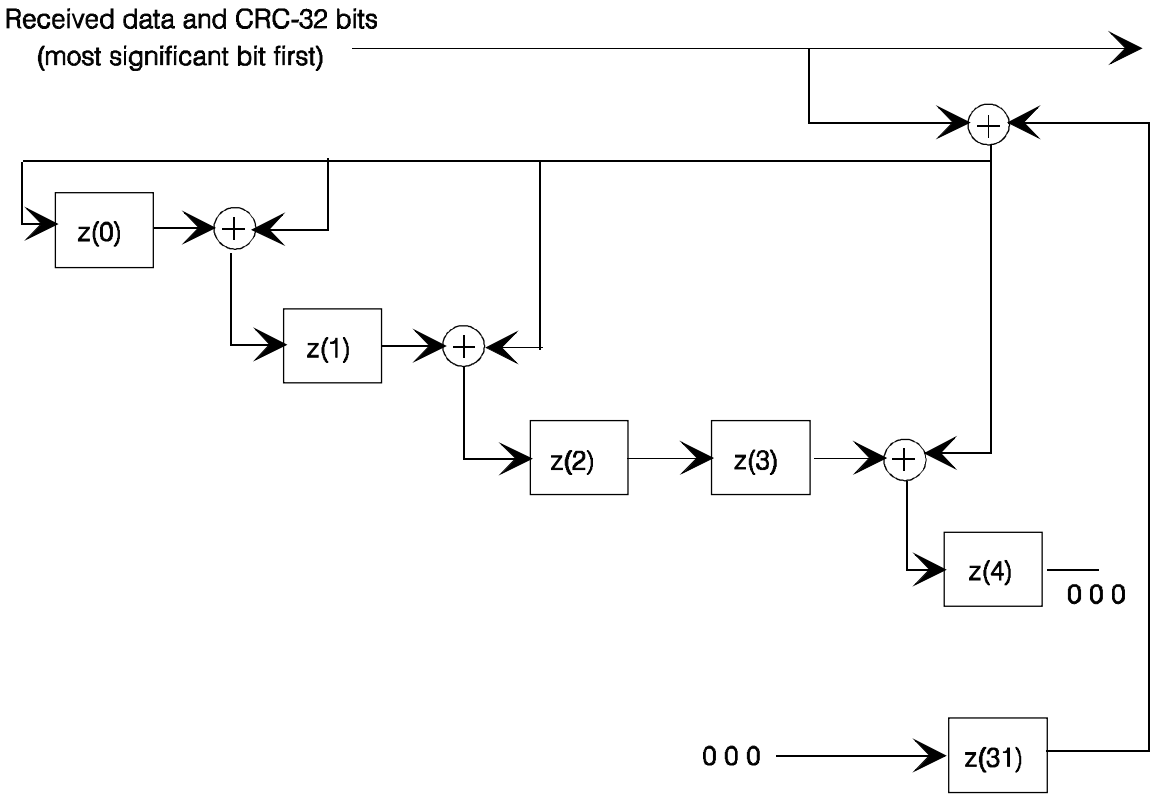


Figure B.1: 32-bit CRC decoder model

The 32-bit CRC decoder operates at bit level and consists of 14 adders + and 32 delay elements z(i). The input of the CRC decoder is added to the output of z(31), and the result is provided to the input z(0) and to one of the inputs of each remaining adder. The other input of each remaining adder is the output of z(i), while the output of each remaining adder is connected to the input of z(i+1), with i = 0, 1, 3, 4, 6, 7, 9, 10, 11, 15, 21, 22 and 25 (see figure B.1).

This is the CRC calculated with the polynomial:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

At the input of the CRC decoder bytes are received.

Each byte is shifted into the CRC decoder one bit at a time, with the Most Significant Bit (MSB) first, i.e. from byte 0x01 (the last byte of the startcode prefix), first the seven "0"s enter the CRC decoder, followed by the one "1".

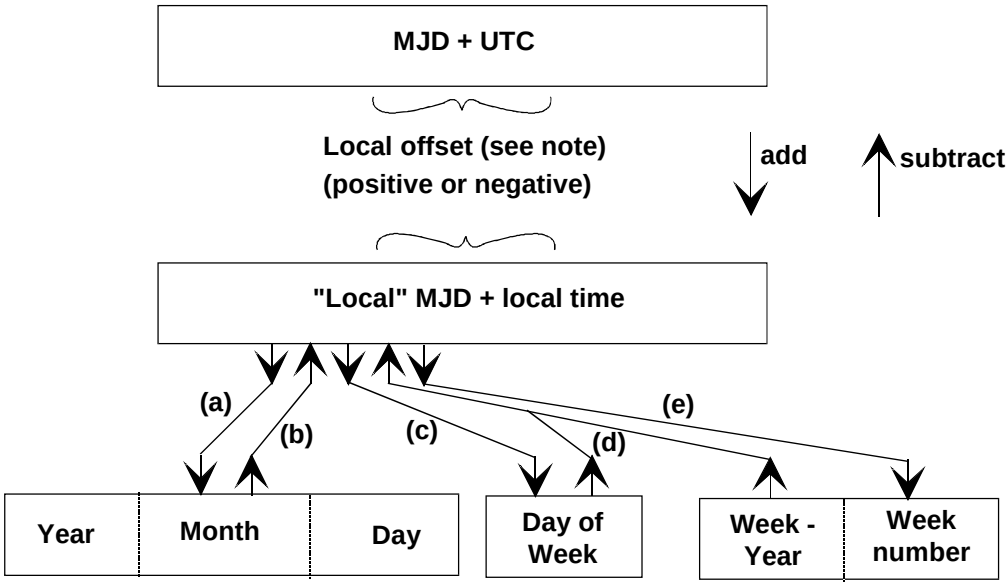
Before the CRC processing of the data of a section the output of each delay element z(i) is set to its initial value "1". After this initialization, each byte of the section is provided to the input of the CRC decoder, including the four CRC_32 bytes.

After shifting the last bit of the last CRC_32 byte into the decoder, i.e. into z(0) after the addition with the output of z(31), the output of all delay elements z(i) is read. In case of no errors, each of the outputs of z(i) has to be zero.

At the CRC encoder the CRC_32 field is encoded with such value that this is ensured.

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 [40]
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

$Y' = \text{int}\ [\ (MJD - 15\,078{,}2) / 365{,}25\ ]$

$M' = \text{int}\ \{\ [\ MJD - 14\,956{,}1 - \text{int}\ (Y' \times 365{,}25)\ ]\ /\ 30{,}6001\ \}$

$D = MJD - 14\,956 - \text{int}\ (Y' \times 365{,}25) - \text{int}\ (M' \times 30{,}6001)$

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 + int [(Y - L) × 365,25] + int [(M + 1 + L × 12) × 30,6001]

c) To find WD from MJD
WD = [(MJD + 2) mod 7] + 1

d) To find MJD from WY, WN, WD
MJD = 15 012 + WD + 7 × { WN + int [(WY × 1 461 / 28) + 0,41] }

e) To find WY, WN from MJD
W = int [(MJD / 7) - 2 144,64]
WY = int [(W × 28 / 1 461) - 0,0079]
WN = W - int [(WY × 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 (informative): Service information implementation of AC-3 and Enhanced AC-3 audio in DVB systems

This annex describes the implementation and implementation guidelines for DVB Service Information for conveying AC-3 and Enhanced AC-3 audio elementary streams within a DVB transport stream.

Since the transmission of AC-3 and Enhanced AC-3 audio streams is optional in DVB systems this annex has an informative status. However, if AC-3 or Enhanced AC-3 audio streams are transmitted in a DVB system the specifications in this annex are to be followed.

D.1 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.

Table D.1: AC-3 Component_type byte value assignments

Component_type byte values (permitted settings)								Description
AC-3 or E-AC-3 flag	full service flag	service type flags			number of channels flags			
b7	b6	b5	b4	b3	b2	b1	b0	
1	X	X	X	X	X	X	X	Stream is Enhanced AC-3 - interpret b0 to b6 as below
0	X	X	X	X	X	X	X	Stream is AC-3 - interpret b0 to b6 as below
X	1	X	X	X	X	X	X	Decoded audio stream is a full service. (suitable for decoding and presentation to the listener)
	0							Decoded audio stream is intended to be combined with another decoded audio stream before presentation to the listener
	X	X	X	X	0	0	0	Mono
					0	0	1	1+1 Mode
					0	1	0	2 channel (stereo)
					0	1	1	2 channel Dolby Surround encoded (stereo)
					1	0	0	Multichannel audio (> 2 channels)
1					1	0	1	Multichannel audio (> 5.1 channels)
1					1	1	0	Elementary stream contains multiple programmes carried in independent substreams
X					1	1	1	reserved
	1	0	0	0	X	X	X	Complete Main (CM)
	0	0	0	1				Music and Effects (ME)
	X	0	1	0				Visually Impaired (VI)
	X	0	1	1				Hearing Impaired (HI)
	0	1	0	0				Dialogue (D)
	X	1	0	1	0	0	0	Commentary (C)
	1	1	1	0				Emergency (E)
	0	1	1	1				Voiceover (VO)
	1	1	1	1	X	X	X	Karaoke (mono and "1+1" prohibited)

D.2 AC-3 Descriptor

The AC-3_descriptor identifies an AC-3 audio elementary stream that has been coded in accordance with TS 102 366 [14] (not including annex E thereof). The intended purpose is to provide configuration information for the IRD.

The descriptor is located in the PSI PMT, and used once in a program map section following the relevant ES_info_length field for any stream containing AC-3 audio coded in accordance with TS 102 366 [14] (not including annex E thereof).

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

The AC-3 descriptor (see table D.2) is used in the PSI PMT to identify streams which carry AC-3 audio. The descriptor is to be located once in a program map section following the relevant ES_info_length field.

D.3.1 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 service(s) 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.

Table D.2: 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){	8	uimsbf
component_type		
}		
if (bsid_flag == 1){	8	uimsbf
bsid		
}		
if (mainid_flag == 1){	8	uimsbf
mainid		
}		
if (asvc_flag == 1){	8	uimsbf
asvc		
}		
for(i=0;i<N;i++){	8	uimsbf
additional_info_byte		
}		
}		

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 TS 102 366 [14], annex E. The intended purpose is to provide configuration information for the IRD.

The descriptor is located in the PSI PMT, and used once in a program map section following the relevant ES_info_length field for any stream containing Enhanced AC-3 audio coded in accordance with TS 102 366 [14], annex E.

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 (main 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

The Enhanced AC-3 descriptor (see table D.3) is used in the PSI PMT to identify streams which carry Enhanced AC-3 audio. The descriptor is to be located once in a program map section following the relevant ES_info_length field.

Table D.3: 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		
}		
}		

D.5.1 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 service(s) 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.4.

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.4.

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.4.

additional_info_byte: These optional bytes are reserved for future use.

Table D.4: Substream1-3 field bit value assignments

Substream1 - 3 bit values								Description
mixing metadata flag	full service flag	Service type flags			number of channels flags			
b7	b6	b5	b4	B3	b2	B1	b0	
1	X	X	X	X	X	X	X	Mixing metadata present in substream
0								No mixing metadata present in substream
	1	X	X	X	X	X	X	Main Service
X	0							Associated Service
	X	X	X	X	0	0	0	Mono
					0	0	1	1+1 Mode
					0	1	0	2 channel (stereo)
					0	1	1	2 channel Dolby Surround encoded (stereo)
					1	0	0	Multichannel audio (> 2 channels)
					1	0	1	Multichannel audio (> 5.1 channels)
					1	1	0	Reserved
					1	1	1	Reserved
0	1	0	0	0	X	X	X	Complete Main (CM)
X	0	0	0	1				Music and Effects (ME)
	X	0	1	0				Visually Impaired (VI)
	X	0	1	1				Hearing Impaired (HI)
	0	1	0	0				Dialogue (D)
	X	1	0	1	0	0	0	Commentary (C)
0	1	1	1	0				Emergency (E)
X	0	1	1	1				Voiceover (VO)
X	1	1	1	1	X	X	X	Karaoke (mono and "1+1" prohibited)

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 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 [42].

Annex G (informative): Service information implementation of DTS coded audio in DVB systems

This annex describes the implementation and implementation guidelines for DVB Service Information for conveying DTS coded audio elementary streams within a DVB transport stream.

Since the transmission of DTS coded audio streams is optional in DVB systems this annex has an informative status. However, if DTS coded audio streams are transmitted in a DVB system the specifications in this annex is to be followed.

G.1 DTS Audio descriptor

The DTS_descriptor identifies a DTS coded audio elementary stream that has been coded in accordance with TS 102 114 [12]. The intended purpose is to provide configuration information for the IRD.

The descriptor is located in the PSI PMT, and used once in a program map section following the relevant ES_info_length field for any stream containing DTS coded audio coded in accordance with TS 102 114 [12].

The descriptor tag provides a unique identification of the presence of the DTS coded elementary stream. Other optional fields in the descriptor may be used to provide identification of the component type mode of the DTS audio coded in the stream (component_type field).

G.2 DTS Descriptor Syntax

The DTS descriptor (see table G.1) is used in the PSI PMT to identify streams which carry DTS audio. The descriptor is to be located once in a program map section following the relevant ES_info_length field.

G.2.1 Semantics for the DTS descriptor

The DTS audio descriptor is shown in table G.1. It is optional that the IRD decodes the DTS audio descriptor.

Table G.1: DTS Audio Descriptor

Syntax	Number of bits	Mnemonic
DTS_audio_stream_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 TS 102 114 [12].

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 TS 102 114 [12].

NOTE: It is recommended that DTS 5.1 compressed audio streams be transmitted at data rate of 384 kpbs 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 NBLKS in listed in TS 102 114 [12]. 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 × (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 FSIZE listed in TS 102 114 [12]. (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 AMODE in DTS Coherent Acoustics Specification. The values for surround_mode are given in table G.4. While broadcasters may use only a subset of these the complete table is given for consistency in TS 102 114 [12], table 5.4.

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 defined4
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 TS 102 114 [12]. 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. The syntax and semantics of the additional_info field are defined TS 102 114 [12].

Annex H (informative): Service information implementation of AAC coded audio in DVB systems

This annex describes the implementation and implementation guidelines for DVB Service Information for conveying AAC coded audio elementary streams within a DVB transport stream.

H.1 AAC Audio descriptor

The AAC_descriptor identifies a AAC coded audio elementary stream that has been coded in accordance with ISO/IEC 14496-3 [21]. The intended purpose is to provide configuration information for the IRD.

The descriptor is located in the PSI PMT, and used once in a program map section following the relevant ES_info_length field for any stream containing AAC coded audio coded in accordance with ISO/IEC 14496-3 [21].

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 (component_type field).

H.2 AAC_Descriptor Syntax

The AAC descriptor (see table H.1) is used in the PSI PMT to identify streams which carry AAC audio. The descriptor is to be located once in a program map section following the relevant ES_info_length field.

H.2.1 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 is located in the PSI PMT, and used once in a program map section following the relevant **ES_info_length** field for any 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
reserved	7	bslbf
if (AAC_type_flag == 1){		
AAC_type	8	uimsbf
}		
for(i=0;i<N;i++){		
additional_info_byte	8	uimsbf
}		
}		
}		

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 shall be set to the Profile and Level according to table 2-71 in ISO/IEC 13818-1:2007/AMD1 [18].

AAC_type_flag: This 1-bit field indicates the presence of the **AAC_type** field. This bit shall be set to "1" if the optional AAC_type field is included in the descriptor.

reserved: This 7-bit field is reserved for future use and shall all be set to "0".

AAC_type: This field shall be set according to table 26 to the value of the component_type field when stream_content is 0x06.

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 are defined TS 101 154 [9].

Annex I (normative): Assignment and interpretation of the service_type field

This annex describes the 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.

I.2 Assignment of service_type

For some services the assignment of a service_type from table 79 may be obvious, e.g. MPEG-2 HD digital television service. However, the decision is not always so straightforward.

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 79, 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 79, 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 is 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 value(s) for stream_type in PSI and/or use of the component_descriptor in SI.

I.2.2 service_type "advanced codec" (various)

The advanced codec service_types have been allocated so as to be able to indicate that a service has been encoded using something other than MPEG-2. More specifically, assignment of one of these service_types implies that the receiver must support a codec other than MPEG-2 to be able to decode and present the service. On this basis it is recommended that MPEG-2 SD-only receivers should not present any service assigned one of these service_types to the viewer for selection.

The assignment of one of these service_types provides a generic indication of the use of some advanced codec but not 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 one of these service_types.

NOTE: Of course, in the case of a specific platform a particular encoding could be implicitly linked to one of 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 one of the advanced codec service_types the component_descriptor should be used in SI 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.

Annex J (normative): Signalling of Receiver-Mixed and Broadcast-Mixed Supplementary Audio

J.1 Overview

Two mechanisms are defined for the provision of supplementary audio (for example audio description):

- Receiver-mixed; and
- Broadcast-mixed.

In both cases, the audio starts as two separate audio channels, one carrying the main audio and the other carrying the supplementary audio. These two audio channels may be mixed together either before broadcast or in the receiver.

This annex defines the signalling in each case to enable the receiver to present the correct audio to the user. Note that TS 101 154 [9] specifies that in the case of receiver mix, the main audio and the supplementary audio shall be encoded using the same codec family.

J.2 Receiver-mixed supplementary audio

J.2.1 Introduction

Where it is important to conserve bit rate, receiver-mixed 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 G defines additional signalling which controls how the receiver mixes the main audio stream and the supplementary audio stream before presentation to the user.

J.2.2 PSI PMT signalling

When a receiver-mixed supplementary stream is present, it shall be signalled in the PSI PMT with the ISO_639_language_code in the ISO_639_language descriptor set to reflect the content of the audio stream and a supplementary_audio descriptor in its ES loop. The other fields shall be set according to clause J.4.

A stream_identifier descriptor should be present.

J.2.3 EIT signalling

A component descriptor, with the same value of component_tag as that used in the PSI PMT descriptor of the supplementary audio stream, should be present in the event descriptors_loop in the EIT for each event for which supplementary audio is broadcast.

As defined in table 26, for visually impaired audio description the values of stream_content and component_type given in table J.1 shall be used.

Table J.1: Receiver-mixed component descriptor values

Audio coding	Stream_content	Component_type
MPEG-1 Layer 2	0x02	0x47
Enhanced AC-3	0x04	0x90 (see note)
HE-AAC	0x06	0x47
HE-AACv2	0x06	0x49
NOTE: As defined in annex D.		

J.3 Broadcast-mixed supplementary audio

J.3.1 Introduction

Broadcast-mixed 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 recognize that this audio stream includes a specific supplementary service and only present it when appropriate.

J.3.2 PSI PMT signalling

When a broadcast-mixed 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 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

A component descriptor, with the same value of component_tag as that used in the PSI PMT descriptor of the supplementary audio stream, should be present in the event descriptors_loop in the EIT for each event for which a supplementary audio stream is broadcast.

As defined in table 26, for visually impaired audio description the values of stream_content and component_type given in table J.2 shall be used.

Table J.2: Broadcast-mixed component descriptor values

Audio coding	Stream_content	Component_type
MPEG-1 Layer 2	0x02	0x48
AC-3	0x04	0b01010xxx (see note)
Enhanced AC-3	0x04	0b11010xxx (see note)
HE-AAC	0x06	0x48
HE-AACv2	0x06	0x4A
NOTE: "xxx" is dependent upon the number of channels, as defined in annex D.		

J.4 PSI signalling of audio purpose

Table J.3 lists the combination of valid parameters that shall be signalled in the PMT.

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	0
Audio description (broadcast mixed)	0x00, 0x01 or 0x03	1	1
Audio description (receiver mixed)	0x03	0	1
Clean audio	0x02	1	2
Spoken subtitles	0x00, 0x01 or 0x03	1	3
NOTE 1: audio_type is broadcast in the ISO_639_language descriptor. NOTE 2: mix_type and editorial_classification are broadcast in the supplementary_audio descriptor. NOTE 3: The supplementary_audio descriptor is optional in this case.			

All other combinations are invalid and shall not be used.

Annex K (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 102 594: " Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT): IPv6 Security; Conformance Abstract Test Suite (ATS) and partial Protocol Implementation eXtra Information for Testing (PIXIT) proforma".

History

Document history		
Edition 1	October 1995	Publication as ETS 300 468
Edition 2	January 1997	Publication as ETS 300 468
V1.3.1	February 1998	Publication
V1.4.1	November 2000	Publication
V1.5.1	May 2003	Publication
V1.6.1	November 2004	Publication
V1.7.1	May 2006	Publication
V1.8.1	July 2008	Publication
V1.9.1	March 2009	Publication
V1.10.1	November 2009	Publication
V1.11.1	December 2009	One-step Approval Procedure OAP 20100331: 2009-12-01 to 2010-03-31