ETSI EN 300 468 V1.5.1 (2003-05)

European Standard (Telecommunications series)

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

European Broadcasting Union   Union Européenne de Radio-Télévision

EBU·UER

D/B
Digital Video
Broadcasting

ETSI

Reference

REN/JTC-DVB-128

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, send your comment to:
editor@etsi.org

Copyright Notification

ETSI

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).

All published ETSI deliverables shall include information which directs the reader to the above source of information.

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).

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.

National transposition dates	
Date of adoption of this EN:	23 May 2003
Date of latest announcement of this EN (doa):	31 August 2003
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	29 February 2004
Date of withdrawal of any conflicting National Standard (dow):	29 February 2004

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 [1] 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 (i.e. EN 300 468) are specified in TR 101 211 [7].

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- 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.

- For a non-specific reference, the latest version applies.

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

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

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

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

[4]   ETSI EN 300 706: "Enhanced Teletext specification".

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

[6]   ETSI ETR 162: "Digital Video Broadcasting (DVB); Allocation of Service Information (SI) codes for DVB systems".

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

[8]   ISO/IEC 10646-1: "Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane".

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

[10]   Void.

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

[12]   ETSI TR 101 154: "Digital Video Broadcasting (DVB); Implementation guidelines for the use of MPEG-2 Systems, Video and Audio in satellite, cable and terrestrial broadcasting applications".

[13]   IEEE 1394: "High Performance Serial Bus".

[14]   ETSI EN 300 231: "Television systems; Specification of the domestic video Programme Delivery Control system (PDC)".

[15]   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".

[16]   ETSI EN 301 775: "Digital Video Broadcasting (DVB); Specification for the carriage of Vertical Blanking Information (VBI) data in DVB bitstreams".

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

[18]   Korea Industrial Standards Association: "Code for Information Interchange (Hangul and Hanja)" Korean Industrial Standard, 1987, Ref. No. KS C 5601-1987.

[19]   ITU-R Recommendation BS.1196-E, annex 2: "Digital Audio Compression (AC-3) Standard (ATSC Standard)".

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.

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

[21]   ETSI TS 102 006: "Digital Video Broadcasting (DVB); Specification for System Software Update in DVB Systems".

[22]   ISO 2015: "Numbering of weeks".

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

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

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

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

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

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

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

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

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

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

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

[34]   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".

[35]   ETSI EN 301 790: "Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems".

[36]   Void.

[37]   Void.

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

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

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

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

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 is described in annex D. The carriage of AC-3 elementary streams as private data within MPEG systems is described in annex C of TR 101 154 [12].

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 shall be 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, e.g. video, audio, teletext

Conditional Access (CA) system: system to control subscriber access to services, programmes and events e.g. Videoguard, Eurocrypt

delivery system: physical medium by which one or more multiplexes are transmitted e.g. 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, e.g. 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: Refers to the standard ISO/IEC 13818.

> NOTE: Systems coding is defined in part 1 [1]. Video coding is defined in part 2 [41]. Audio coding is defined in part 3 [33].

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, e.g. 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. 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 shall be 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 shall be set to "1".

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

   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.
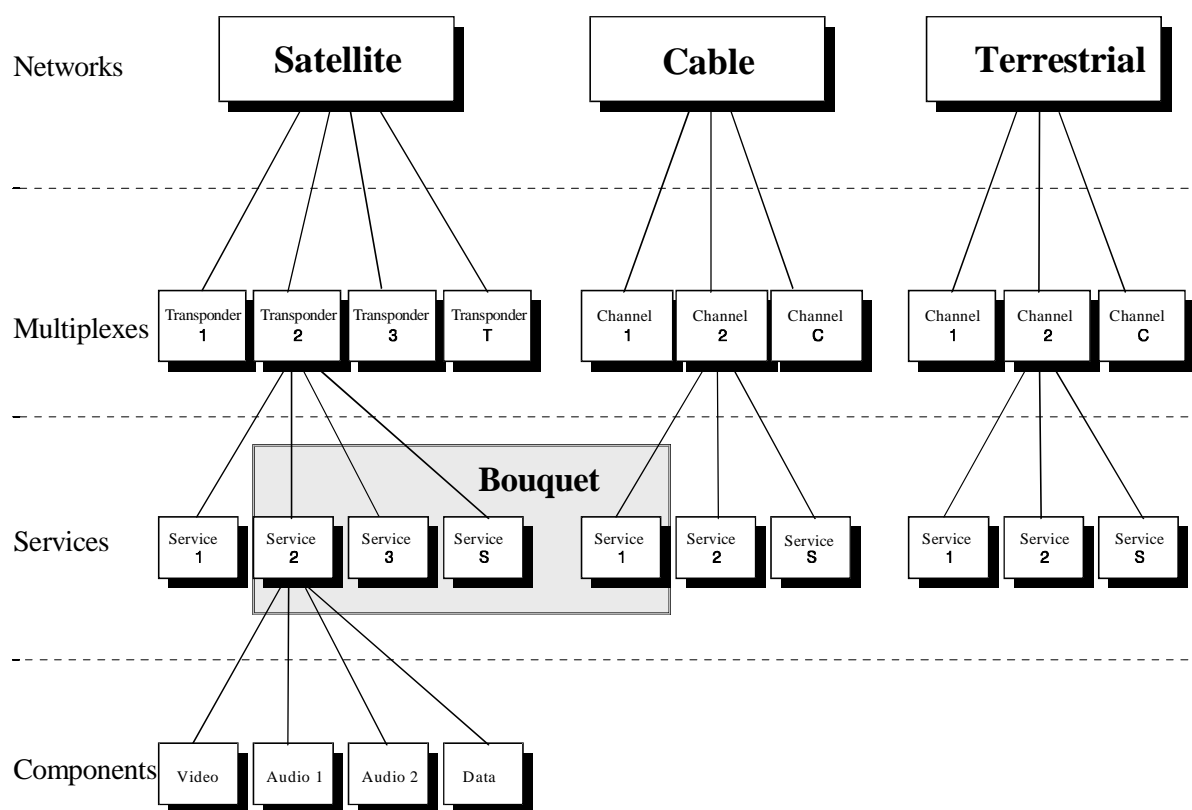
Figure 1: Digital broadcasting, service delivery model

3.2   Abbreviations

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

AC-3	Dolby AC-3 audio coding [19]
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
CRC	Cyclic Redundancy Check
DAB	Digital Audio Broadcasting
DIT	Discontinuity Information Table
DVB	Digital Video Broadcasting
DVD	Digital Versatile Disc
EBU	European Broadcasting Union

EIT	Event Information Table
EMM	Entitlement Management Message
EPG	Electronic Programme Guide
FEC	Forward Error Correction
IEC	International Electrotechnical Commission
IRD	Integrated Receiver Decoder
ISO	International Organization for Standardization
LSB	Least Significant Bit
MJD	Modified Julian Date
MPEG	Moving Pictures Expert Group
NIT	Network Information Table
NVOD	Near Video On Demand
PAT	Program Association Table
PDC	Programme Delivery Control
PID	Packet IDentifier
PMT	Program Map Table
PSI	Program Specific Information
PSTN	Public Switched Telephone Network
QAM	Quadrature Amplitude Modulation
QPSK	Quaternary Phase Shift Keying
rpchof	remainder polynomial coefficients, highest order first
RS	Reed - Solomon
RST	Running Status Table
ScF	Scale Factor
SDT	Service Description Table
SI	Service Information
SIT	Selection Information Table
SMI	Storage Media Interoperability
ST	Stuffing Table
TDT	Time and Date Table
TOT	Time Offset Table
TPS	Transmission Parameter Signalling
TS	Transport Stream
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 [1] 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 [1] specification, but the data format is outside the scope of ISO/IEC 13818-1 [1]. 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.
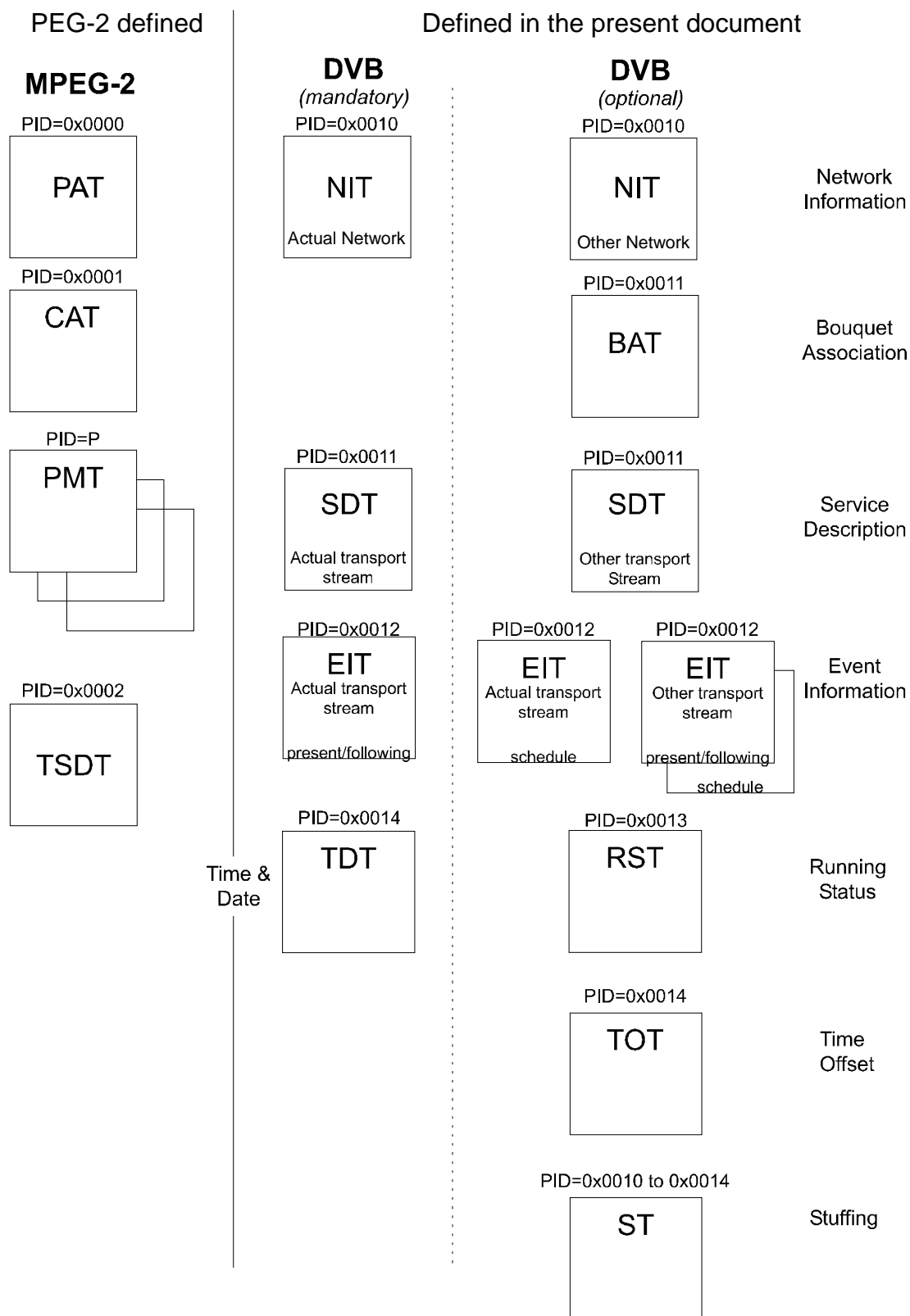
Figure 2: General organization of the Service Information (SI)

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 [1].

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 [1].

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	0x0012
RST, ST	0x0013
TDT, TOT, ST	0x0014
network synchronization	0x0015
reserved for future use	0x0016 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 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, bibliography. 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 [1]) 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 [1].

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 ETS application area. Networks are assigned individual network_id values, which serve as unique identification codes for networks. The allocation of these codes may be found in ETR 162 [6]. In the case that the NIT is transmitted on the network on which the TS was originated, the network_id and the original_network_id shall take the same value.

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 [7].

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 1. 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 ETR 162 [6]. 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 ETR 162 [6].

Table 3: Network information section

Syntax	No. 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 ETR 162 [6].

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 ETR 162 [6]. All BAT sections shall take a table_id value of 0x4A.

Table 4: Bouquet association section

Syntax	No. 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 ETR 162 [6].

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	No. 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 [7] 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 [7] 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 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	No. 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). If only one table is used this is set to the table_id of this table. The chronological order of information is maintained across successive table_id values.

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: 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: 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	No. 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	No. 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	No. 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	No. 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 [7].

6.1   Descriptor identification and location

Table 12 lists the descriptors 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	*	-	-	-	-	-	-
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	-	-	*	*	-	-	*
reserved for future use	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	0x71	-	-	*	-	-	-	-
service_availability_descriptor	0x72	-	-	*	-	-	-	-
reserved for future use	0x73 to 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).
*   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 [1]. 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 [1]. 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	No. 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 is supported. Note that 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 [12]
b_1	reserved for future use
b_2	reserved for future use
b_3	reserved for future use
b_4	reserved for future use
b_5	reserved for future use
b_6	reserved for future use
b_7	reserved 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 [33] and ISO/IEC 11172-3 [34]. 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	No. 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 [12]
b_1	Extended Ancillary Data [12]
b_2	Announcement Switching Data [12]
b_3	DAB Ancillary Data [20]
b_4	Scale Factor Error Check (ScF-CRC) [12]
b_5	reserved for future use
b_6	reserved for future use
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	No. 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.

ETSI

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	No. 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	No. 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 ETR 162 [6].

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	No. 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 (i=0;i<N;i++){		
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	No. 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_extend_of_latitude	12	uimsbf
cell_extend_of_longitude	12	uimsbf
subcell_info_loop_length	8	uimsbf
for (i=0;i<N;i++){		
cell_id_extension	8	uimsbf
subcell_latitude	16	uimsbf
subcell_longitude	16	uimsbf
subcell_extend_of_latitude	12	uimsbf
subcell_extend_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 $(90^{\circ}/2^{15})$. Southern latitudes shall be considered negative and northern latitudes positive.

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

cell_extend_of_latitude: This 12-bit field, coded as an unsigned binary number, shall specify the extend 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 extend_of_latitude field by $(90^{\circ}/2^{15})$.

cell_extend_of_longitude: This 12-bit field, coded as an unsigned binary number, shall specify the extend 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 extend_of_longitude field by $(180^{\circ}/2^{15})$.

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

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

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

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

subcell_extend_of_latitude: This 12-bit field, coded as an unsigned binary number, shall specify the extend 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 extend_of_latitude field by $(90^{\circ}/2^{15})$.

subcell_extend_of_longitude: This 12-bit field, coded as an unsigned binary number, shall specify the extend 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 extend_of_longitude field by $(180^{\circ}/2^{15})$.

6.2.8   Component descriptor

The component descriptor identifies the type of component stream and may be used to provide a text description of the elementary stream (see table 25).

Table 25: Component descriptor

Syntax	No. 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 [3]. 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 [5] 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	video, 4:3 aspect ratio, 25 Hz
0x01	0x02	video, 16:9 aspect ratio with pan vectors, 25 Hz
0x01	0x03	video, 16:9 aspect ratio without pan vectors, 25 Hz
0x01	0x04	video, > 16:9 aspect ratio, 25 Hz
0x01	0x05	video, 4:3 aspect ratio, 30 Hz
0x01	0x06	video, 16:9 aspect ratio with pan vectors, 30 Hz
0x01	0x07	video, 16:9 aspect ratio without pan vectors, 30 Hz
0x01	0x05	video, > 16:9 aspect ratio, 30 Hz
0x01	0x09	high definition video, 4:3 aspect ratio, 25 Hz
0x01	0x0A	high definition video, 16:9 aspect ratio with pan vectors, 25 Hz
0x01	0x0B	high definition video, 16:9 aspect ratio without pan vectors, 25 Hz
0x01	0x0C	high definition video, > 16:9 aspect ratio, 25 Hz
0x01	0x0D	high definition video, 4:3 aspect ratio, 30 Hz
0x01	0x0E	high definition video, 16:9 aspect ratio with pan vectors, 30 Hz
0x01	0x0F	high definition video, 16:9 aspect ratio without pan vectors, 30 Hz
0x01	0x10	high definition video, > 16:9 aspect ratio, 30 Hz
0x01	0x11 to 0xAF	reserved for future use
0x01	0xB0 to 0xFE	user defined
0x01	0xFF	reserved for future use
0x02	0x00	reserved for future use
0x02	0x01	audio, single mono channel
0x02	0x02	audio, dual mono channel
0x02	0x03	audio, stereo (2 channel)
0x02	0x04	audio, multi-lingual, multi-channel
0x02	0x05	audio, surround sound
0x02	0x06 to 0x3F	reserved for future use
0x02	0x40	audio description for the visually impaired
0x02	0x41	audio for the hard of hearing
0x02	0x42 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 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 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 table D.1)
0x04	0x80 to 0xFF	reserved for future use
0x05 to 0x0B	0x00 to 0xFF	reserved for future use
0x0C to 0x0F	0x00 to 0xFF	user defined

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	No. 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_nibble	4	uimsbf
user_nibble	4	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_nibble: This 4-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

Content_nibble_level_1	Content_nibble_level_2	Description
		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
		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

Content_nibble_level_1	Content_nibble_level_2	Description
		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 & health
0xA	0x5	cooking
0xA	0x6	advertisement/shopping
0xA	0x7	gardening
0xA	0x8 to 0xE	reserved for future use
0xA	0xF	user defined
		Special Characteristics:
0xB	0x0	original language
0xB	0x1	black & 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	No. 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:

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

country_code: This 24-bit field identifies a country using the 3-character code as specified in ISO 3166 [2]. Each character is coded into 8-bits according to ISO/IEC 8859-1 [5] 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 ETR 162 [6].

> 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	No. 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 ETR 162 [6].

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 [3] 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 [5] 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	No. 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 ETR 162 [6].

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 that the id_selector_bytes may differ from the selector_bytes of the corresponding data_broadcast_descriptor.

6.2.13   Delivery system descriptors

The delivery system descriptors all have the same overall length of 13 bytes. This facilitates the interchange of these descriptors when a TS is transcoded from one delivery system to another, e.g. satellite to cable.

6.2.13.1   Cable delivery system descriptor

See table 32.

Table 32: Cable delivery system descriptor

Syntax	No. 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
1111	No conv. coding
0111 to 1110	reserved for future use

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	No. 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
modulation	5	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

modulation: This is a 5-bit field. It specifies the modulation scheme used on a satellite delivery system according to table 38.

Table 38: Modulation scheme for satellite

Modulation bit 4 3210	Description
0 0000	not defined
0 0001	QPSK
0 0010	8PSK
0 0011	16-QAM
0 0100 to 1 1111	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.

6.2.13.3   Terrestrial delivery system descriptor

See table 39.

Table 39: Terrestrial delivery system descriptor

Syntax	No. of bits	Identifier
terrestrial_delivery_system_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
centre_frequency	32	bslbf
bandwidth	3	bslbf
reserved_future_use	5	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 uimsbf field giving the binary coded 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 what is the bandwidth in use.

Table 40: Signalling format for the bandwidth

Bandwidth	Bandwidth value
000	8 MHz
001	7 MHz
010	6 MHz
011 to 111	reserved for future use

constellation: This is a 2-bit field. It specifies the constellation pattern used on a terrestrial delivery system according to table 41.

Table 41: 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.

Table 42: Signalling format for the α values

Hierarchy_information	α value
000	non-hierarchical
001	$\alpha = 1$
010	$\alpha = 2$
011	$\alpha = 4$
100 to 111	reserved for future use

code_rate: The code_rate is a 3-bit field specifying the inner FEC scheme used according to table 43. Non-hierarchical channel coding and modulation requires signalling of one code rate. In this case, 3 bits specifying code_rate according to table 44 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 43: 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.

Table 44: 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 45: Signalling format for transmission mode

transmission_mode	Description
00	2k mode
01	8k mode
10 to 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 DSNG (Digital Satellite News Gathering) 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 [15] and annex D).

For DSNG applications at least one DSNG descriptor shall be present in the TSDT.

Table 46: DSNG descriptor

Syntax	No. 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 [15] and annex D.

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 47: Extended event descriptor

Syntax	No. 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 [3]. 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 [5] 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   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 48: Frequency list descriptor

Syntax	No. 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 49.

Table 49: 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.

6.2.17   Linkage descriptor

The linkage descriptor (see table 50) 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.

Table 50: Linkage descriptor

Syntax	No. 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){		
for (i=0;i<N;i++){		
private_data_byte	8	bslbf
}		
}		
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
}		
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 51).

Table 51: 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 [21]
0x0A	TS containing SSU BAT or NIT [21]
0x0B to 0x7F	reserved for future use
0x80 to 0xFE	user defined
0xFF	reserved for future use

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

hand-over_type: This is a 4-bit field specifying the type of hand-over (see table 52).

Table 52: 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 53).

Table 53: Origin type coding

Origin_type	Description
0x00	NIT
0x01	SDT

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

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

6.2.18   Local time offset descriptor

The local time offset descriptor (see table 54) may be used in the TOT to describe country specific dynamic changes of the local time offset relative to UTC.

ETSI

Table 54: Local time offset descriptor

Syntax	No. 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 [2].
Each character is coded into 8-bits according to ISO/IEC 8859-1 [5] 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 ETR 162 [6]. 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 55: 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 - 11 1111	reserved

local_time_offset_polarity: This 1-bit information indicates the polarity of the following local_offset_time.
If this bit is set to "0" the polarity is positive and the local time is advanced to UTC. (Usually east direction from
Greenwich). 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 current offset time from UTC in the range between -12 hours and
+13 hours at the area which is indicated by the combination of country_code and country_region_id in advance.
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 next offset time after the change from UTC in the range between
-12 hours and +13 hours at the area which is indicated by the combination of country_code and country_region_id in
advance. These 16-bits are coded as 4-digits in 4-bit BCD in the order hour tens, hour, minute tens and minutes.

6.2.19   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 56.

Table 56: Mosaic descriptor

Syntax	No. 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;i<elementary_cell_field_length;i++) {		
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 57 for coding.

Table 57: 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 58 for coding.

Table 58: 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 59.

Table 59: 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 60 for coding.

Table 60: 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.

6.2.20   Multilingual bouquet name descriptor

The multilingual bouquet name descriptor (see table 61) provides the bouquet name in text form in one or more languages.

Table 61: Multilingual bouquet name descriptor

Syntax	No. 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 [3] 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 [5] 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.

6.2.21   Multilingual component descriptor

The multilingual component descriptor (see table 62) provides a text description of a component in one or more languages. The component is identified by its component tag value.

Table 62: Multilingual component descriptor

Syntax	No. 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 [3]. 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 [5] 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.

6.2.22   Multilingual network name descriptor

The multilingual network name descriptor (see table 63) provides the network name in text form in one or more languages.

Table 63: Multilingual network name descriptor

Syntax	No. 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 [3] 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 [5] 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.

6.2.23   Multilingual service name descriptor

The multilingual service name descriptor (see table 64) provides the names of the service provider and service in text form in one or more languages.

Table 64: Multilingual service name descriptor

Syntax	No. 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 [3] 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 [5] 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.

6.2.24   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 65) 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 6.2.29.
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 65: NVOD reference descriptor

Syntax	No. 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.

6.2.25 Network name descriptor

The network name descriptor provides the network name in text form (see table 66).

Table 66: Network name descriptor

Syntax	No. 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.

6.2.26 Parental rating descriptor

This descriptor (see table 67) gives a rating based on age and allows for extensions based on other rating criteria.

Table 67: Parental rating descriptor

Syntax	No. 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 [2]. Each character is coded into 8-bits according to ISO/IEC 8859-1 [5] 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 ETR 162 [6].

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

Table 68: 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.

6.2.27   Partial Transport Stream (TS) descriptor

See clause 7.2.1.

6.2.28   PDC descriptor

The PDC_descriptor extends the DVB system with the functionalities of PDC [14]. The descriptor carries the Programme Identification Label (PIL) as defined in EN 300 231 [14]. The PIL contains date and time of the first published start time of a certain event.

Table 69: PDC_descriptor

Syntax	No. 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 [14]. 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.

6.2.29   Private data specifier descriptor

This descriptor is used to identify the specifier of any private descriptors or private fields within descriptors.

Table 70: Private data specifier descriptor

Syntax	No. 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 ETR 162 [6].

6.2.30   Short smoothing buffer descriptor

A smoothing_buffer_descriptor is specified in ISO/IEC 13818-1 [1] 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 [1] (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 71: Short smoothing buffer descriptor

Syntax	No. 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 72.

Table 72: 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 73.

Table 73: 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 - 32	$((\text{value}) - 16) \times 0,5$ (7,0, 7,5, 8,0 Mbit/s)
33 - 37	$((\text{value}) - 24)$ (9, 10, 11, 12, 13 Mbit/s)
38	13,5
39 - 43	$((\text{value}) - 25)$ (14, 15, 16, 17, 18 Mbit/s)
44 - 47	$((\text{value}) - 34) \times 2$ (20, 22, 24, 26 Mbit/s)
48	27
49 - 55	$((\text{value}) - 35) \times 2$ (28, 30, 32 ... 40 Mbit/s)
56	44
57	48
58	54
59	72
60	108
61 - 63	DVB_reserved

6.2.31   Service descriptor

The service descriptor (see table 74) provides the names of the service provider and the service in text form together with the service_type.

Table 74: Service descriptor

Syntax	No. 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. It shall be coded according to table 75.

Table 75: Service type coding

Service_type	Description
0x00	reserved for future use
0x01	digital television service
0x02	digital radio sound service
0x03	Teletext service
0x04	NVOD reference service
0x05	NVOD time-shifted service
0x06	mosaic service
0x07	PAL coded signal
0x08	SECAM coded signal
0x09	D/D2-MAC
0x0A	FM Radio
0x0B	NTSC coded signal
0x0C	data broadcast service
0x0D	reserved for Common Interface Usage [17]
0x0E	RCS Map (see EN 301 790 [35])
0x0F	RCS FLS (see EN 301 790 [35])
0x10	DVB MHP service
0x11 to 0x7F	reserved for future use
0x80 to 0xFE	user defined
0xFF	reserved for future use

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.

6.2.32 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 76: Service availability descriptor

Syntax	No. 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.

6.2.33 Service list descriptor

The service list descriptor (see table 77) provides a means of listing the services by service_id and service type.

Table 77: Service list descriptor

Syntax	No. 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 (NVOD reference service) the service_id does not have a corresponding program_number.

service_type: This is an 8-bit field specifying the type of the service. It shall be coded according to table 76.

6.2.34   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 78: Service move descriptor

Syntax	No. 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. If the service remains within the same original network, then the new_service_id is the same as the previous service_id.

6.2.35   Short event descriptor

The short event descriptor provides the name of the event and a short description of the event in text form (table 79).

Table 79: Short event descriptor

Syntax	No. 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 [3] 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 [5] 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.

6.2.36   Stream identifier descriptor

The stream identifier descriptor (see table 80) 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 80: Stream identifier descriptor

Syntax	No. 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.

6.2.37   Stuffing descriptor

The stuffing descriptor provides a means of invalidating previously coded descriptors or inserting dummy descriptors for table stuffing (see table 81).

Table 81: Stuffing descriptor

Syntax	No. 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.

6.2.38   Subtitling descriptor

In the ISO/IEC 13818-1 [1] 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 82.

Table 82: Subtitling descriptor

Syntax	No. 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 [3] 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 [5] 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 16 "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.

6.2.39   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 "Implementation guidelines for the use of telecommunications interfaces in Digital Video Broadcasting systems' (see bibliography).

The telephone descriptor syntax is specified in table 83.

Table 83: Telephone descriptor

Syntax	No. 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 [5] 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 [5] 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 [5] 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 [5] 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 [5] gives one alphanumeric character of the core number.

6.2.40   Teletext descriptor

The Teletext descriptor (see table 84) 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 84: Teletext descriptor

Syntax	No. 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 [3] 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 [5] 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 85.

Table 85: 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 [4].

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 [4].

6.2.41   Time shifted event descriptor

The time shifted event descriptor (see table 86) is used in place of the short_event_descriptor to indicate an event which is a time shifted copy of another event.

Table 86: Time shifted event descriptor

Syntax	No. 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.

6.2.42   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 87).

Table 87: Time shifted service descriptor

Syntax	No. 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.

6.2.43   Transport stream descriptor

The transport stream descriptor, being transmitted in the TSDT (see [1]) only, may be used to indicate the compliance of a transport stream with an MPEG based system, e.g. DVB.

Table 88: Transport stream descriptor

Syntax	No. 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").

6.2.44   VBI data descriptor

The VBI data descriptor shall be used in the PSI PMT of a stream which carries VBI data as defined in [16]. 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 89: VBI_data_descriptor

Syntax	No. 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 90. 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 EN 301 775 [16], clauses 4.5 to 4.9). 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 90: 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 .. 0xFF	reserved for future use

6.2.45   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 91: VBI teletext descriptor

Syntax	No. 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 6.2.40. 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.

7   Storage Media Interoperability (SMI) measures

IEC 61883 [11] describes methods for delivering TS over the IEEE 1394 [13] "High Performance Serial Bus" 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 [1].
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 TR 101 154 [12] 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 below. 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 92) is to be inserted at transition points at which SI information may be discontinuous.

Table 92: Discontinuity information section

Syntax	No. 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 93: Selection information section

Syntax	No. 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 [7].

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 [1] 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 94: Partial Transport Stream (TS) descriptor

Syntax	No. 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 a default character set is assumed.

A.1   Control codes

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	Description
0x80 to 0x85	reserved for future use
0x86	character emphasis on
0x87	character emphasis off
0x88 to 0x89	reserved for future use
0x8A	CR/LF
0x8B to 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 10646-1 [8]

Control code	Description
0xE080 to 0xE085	reserved for future use
0xE086	character emphasis on
0xE087	character emphasis off
0xE088 to 0xE089	reserved for future use
0xE08A	CR/LF
0xE08B to 0xE09F	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. The selection of character table is indicated as follows:

-    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;

-    if the first byte of the text field has a value in the range "0x01" to "0x0F" then the remaining bytes in the text item are coded in accordance with the character coding tables which are given in table A.3;

Table A.3: Character coding fables

First byte value	Character code table	Table description	Reproduced in figure
0x01	ISO/IEC 8859-5 [32]	Latin/Cyrillic alphabet	A.2
0x02	ISO/IEC 8859-6 [31]	Latin/Arabic alphabet	A.3
0x03	ISO/IEC 8859-7 [30]	Latin/Greek alphabet	A.4
0x04	ISO/IEC 8859-8 [29]	Latin/Hebrew alphabet	A.5
0x05	ISO/IEC 8859-9 [28]	Latin alphabet No. 5	A.6
0x06	ISO/IEC 8859-10 [27]	Latin alphabet No. 6	A.7
0x07	ISO/IEC 8859-11 [26]	Latin/Thai (draft only)	A.8
0x08	ISO/IEC 8859-12	possibly reserved for Indian	
0x09	ISO/IEC 8859-13 [25]	Latin alphabet No. 7	A.9
0x0A	ISO/IEC 8859-14 [24]	Latin alphabet No. 8 (Celtic)	A.10
0x0B	ISO/IEC 8859-15 [23]	Latin alphabet No. 9	A.11
0x0C to 0x0F	reserved for future use		

- 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 by ISO Standard 8859, parts 1 to 9;

- if the first byte of the text field has a value "0x11" then the remaining bytes in the text item are coded in pairs in accordance with the Basic Multilingual Plane of ISO/IEC 10646-1 [8];

- if the first byte of the text field has a value "0x12" then the remaining bytes in the text item are coded in accordance with the Korean Character Set KSC5601-1987 [18];

- if the first byte of the text field has a value "0x13" then the remaining bytes in the text item are coded in accordance with the Simplified Chinese Character Set GB-2312-1980;

- if the first byte of the text field has a value "0x14" then the remaining bytes in the text item are coded in accordance with the Big5 subset of ISO/IEC 10646-1 [8] for use with Traditional Chinese.

Values for the first byte of "0x15" to "0x1F" are reserved for future use.

Second nibble → First nibble →

Second \ First	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	`	p			NBSP	°		—	Ω	Κ
1			!	1	A	Q	a	q			¡	±	`	¹	Æ	æ
2			"	2	B	R	b	r			¢	²	´	®	Đ	đ
3			#	3	C	S	c	s			£	³	^	©	ª	ð
4			$	4	D	T	d	t			€	×	~	™	Ħ	ħ
5			%	5	E	U	e	u			¥	µ	¯	♪		ı
6			&	6	F	V	f	v				¶	˘	¬	IJ	ij
7			'	7	G	W	g	w			§	·	˙	¦	Ŀ	ŀ
8			(	8	H	X	h	x			¤	÷	¨		Ł	ł
9			)	9	I	Y	i	y			'	′			Ø	ø
A			*	:	J	Z	j	z			"	″	˚		Œ	œ
B			+	;	K	[	k	{			«	»	¸		º	ß
C			,	<	L	\	l	|			←	¼		⅛	Þ	þ
D			-	=	M	]	m	}			↑	½	˝	⅜	Ŧ	ŧ
E			.	>	N	^	n	~			→	¾	˛	⅝	Ŋ	ŋ
F			/	?	O	_	o	o			↓	¿	ˇ	⅞	ŉ	SHY

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 [9] 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

Second nibble

First nibble →

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	0	@	P	`	p			NBSP	А	Р	а	р	№
1			!	1	A	Q	a	q			Ё	Б	С	б	с	ё
2			"	2	B	R	b	r			Ђ	В	Т	в	т	ђ
3			#	3	C	S	c	s			Ѓ	Г	У	г	у	ѓ
4			$	4	D	T	d	t			Є	Д	Ф	д	ф	є
5			%	5	E	U	e	u			Ѕ	Е	Х	е	х	ѕ
6			&	6	F	V	f	v			І	Ж	Ц	ж	ц	і
7			'	7	G	W	g	w			Ї	З	Ч	з	ч	ї
8			(	8	H	X	h	x			Ј	И	Ш	и	ш	ј
9			)	9	I	Y	i	y			Љ	Й	Щ	й	щ	љ
A			*	:	J	Z	j	z			Њ	К	Ъ	к	ъ	њ
B			+	;	K	[	k	{			Ћ	Л	Ы	л	ы	ћ
C			,	<	L	\	l	|			Ќ	М	Ь	м	ь	ќ
D			-	=	M	]	m	}			SHY	Н	Э	н	э	§
E			.	>	N	^	n	~			Ў	О	Ю	о	ю	ў
F			/	?	O	_	o				Џ	П	Я	п	я	џ

NOTE 1: For the Ruthenian language, the characters in code positions Ah/5h (S) and Fh/5h (s) are replaced by Ґ and ґ, respectively.

NOTE 2: Table reproduced from ISO/IEC 8859-5 [32].

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

NOTE: Table reproduced from ISO/IEC 8859-6 [31].

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

Second nibble ↓   First nibble →

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	0	@	P	`	p			NBSP	°	$\ddot{\iota}$	Π	$\ddot{\upsilon}$	π
1			!	1	A	Q	a	q			'	±	A	P	α	ρ
2			"	2	B	R	b	r			,	²	B		β	ς
3			#	3	C	S	c	s			£	³	Γ	Σ	γ	σ
4			$	4	D	T	d	t				΄	Δ	T	δ	τ
5			%	5	E	U	e	u				΅	E	Υ	ε	υ
6			&	6	F	V	f	v			¦	Ά	Z	Φ	ζ	φ
7			'	7	G	W	g	w			§	·	H	X	η	χ
8			(	8	H	X	h	x			¨	Έ	Θ	Ψ	θ	ψ
9			)	9	I	Y	i	y			©	Ή	I	Ω	ι	ω
A			*	:	J	Z	j	z				Ί	K	Ϊ	κ	$\ddot{\iota}$
B			+	;	K	[	k	{			«	»	Λ	Ϋ	λ	$\ddot{\upsilon}$
C			,	<	L	\	l	|			¬	Ό	M	$\acute{\alpha}$	μ	$\acute{o}$
D			-	=	M	]	m	}			SHY	½	N	$\acute{\epsilon}$	ν	$\acute{\upsilon}$
E			.	>	N	^	n	~				Ύ	Ξ	$\acute{\eta}$	ξ	$\acute{\omega}$
F			/	?	O	_	o				—	Ώ	O	$\acute{\iota}$	o	

NOTE: Table reproduced from ISO/IEC 8859-7 [30].

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

NOTE: Table reproduced from ISO/IEC 8859-8 [29].

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

Second nibble

First nibble ———→

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	0	@	P	`	p			NBSP	°	À	Ğ	à	ğ
1			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
2			"	2	B	R	b	r			¢	²	Â	Ò	â	ò
3			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
4			$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
5			%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
6			&	6	F	V	f	v			¦	¶	Æ	Ö	æ	ö
7			'	7	G	W	g	w			§	·	Ç	×	ç	÷
8			(	8	H	X	h	x			¨	¸	È	Ø	è	ø
9			)	9	I	Y	i	y			©	¹	É	Ù	é	ù
A			*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
B			+	;	K	[	k	{			«	»	Ë	Û	ë	û
C			,	<	L	\	l	|			¬	¼	Ì	Ü	ì	ü
D			-	=	M	]	m	}			SHY	½	Í	İ	í	ı
E			.	>	N	^	n	~			®	¾	Î	Ş	î	ş
F			/	?	O	_	o				¯	¿	Ï	ß	ï	ÿ

NOTE: Table reproduced from ISO/IEC 8859-9 [28].

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

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	°	Ā	Đ	ā	ð	0
0 0 0 1 **01**			!	1	A	Q	a	q			Ą	ą	Á	Ņ	á	ŋ	1
0 0 1 0 **02**			"	2	B	R	b	r			Ē	ē	Â	Ō	â	ō	2
0 0 1 1 **03**			#	3	C	S	c	s			Ģ	ģ	Ã	Ó	ã	ó	3
0 1 0 0 **04**			$	4	D	T	d	t			Ī	ī	Ä	Ô	ä	ô	4
0 1 0 1 **05**			%	5	E	U	e	u			Ĩ	ĩ	Å	Õ	å	õ	5
0 1 1 0 **06**			&	6	F	V	f	v			Ķ	ķ	Æ	Ö	æ	ö	6
0 1 1 1 **07**			'	7	G	W	g	w			§	·	Į	Ũ	į	ũ	7
1 0 0 0 **08**			(	8	H	X	h	x			Ļ	ļ	Č	Ø	č	ø	8
1 0 0 1 **09**			)	9	I	Y	i	y			Đ	đ	É	Ų	é	ų	9
1 0 1 0 **10**			*	:	J	Z	j	z			Š	š	Ę	Ú	ę	ú	A
1 0 1 1 **11**			+	;	K	[	k	{			Ŧ	ŧ	Ë	Û	ë	û	B
1 1 0 0 **12**			,	<	L	\	l	|			Ž	ž	Ė	Ü	ė	ü	C
1 1 0 1 **13**			–	=	M	]	m	}			SHY	—	Í	Ý	í	ý	D
1 1 1 0 **14**			.	>	N	^	n	~			Ū	ū	Î	Þ	î	þ	E
1 1 1 1 **15**			/	?	O	_	o				Ŋ	ŋ	Ï	ß	ï	ĸ	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	hex

NOTE: Table reproduced from ISO/IEC 8859-10 [27].

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

b8 b7 b6 b5 / b4 b3 b2 b1	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	hex
0 0 0 0 00			SP	0	@	P	`	p			NBSP	ฐ	ภ	ะ	เ	๐	0
0 0 0 1 01			!	1	A	Q	a	q			ก	ฑ	ม	ั	แ	๑	1
0 0 1 0 02			"	2	B	R	b	r			ข	ฒ	ย	า	โ	๒	2
0 0 1 1 03			#	3	C	S	c	s			ฃ	ณ	ร	ำ	ใ	๓	3
0 1 0 0 04			$	4	D	T	d	t			ค	ด	ฤ	ิ	ไ	๔	4
0 1 0 1 05			%	5	E	U	e	u			ฅ	ต	ล	ี	ๅ	๕	5
0 1 1 0 06			&	6	F	V	f	v			ฆ	ถ	ฦ	ึ	ๆ	๖	6
0 1 1 1 07			'	7	G	W	g	w			ง	ท	ว	ื	๋	๗	7
1 0 0 0 08			(	8	H	X	h	x			จ	ธ	ศ	ุ	็	๘	8
1 0 0 1 09			)	9	I	Y	i	y			ฉ	น	ษ	ู	์	๙	9
1 0 1 0 10			*	:	J	Z	j	z			ช	บ	ส	ฺ	๎	๚	A
1 0 1 1 11			+	;	K	[	k	{			ซ	ป	ห		๊	๛	B
1 1 0 0 12			,	<	L	\	l	|			ฌ	ผ	ฬ		๋		C
1 1 0 1 13			–	=	M	]	m	}			ญ	ฝ	อ		์		D
1 1 1 0 14			.	>	N	^	n	~			ฎ	พ	ฮ		๎		E
1 1 1 1 15			/	?	O	_	o				ฏ	ฟ	ฯ	฿	๏		F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	hex

NOTE: Table reproduced from ISO/IEC 8859-11 [26].

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	ŋ	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 [25].

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

NOTE: Table reproduced from ISO/IEC 8859-14 [24].

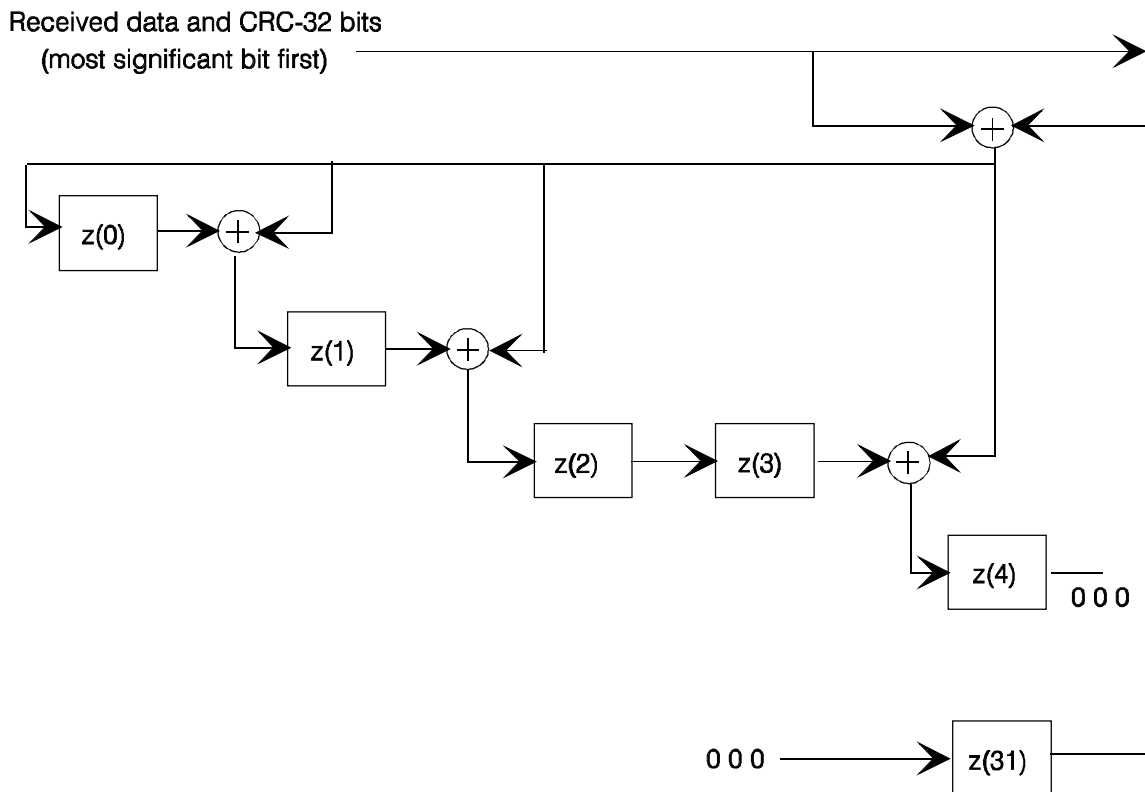**Figure A.10: Character code table 0A – Latin alphabet number 8 (Celtic)**

NOTE: Table reproduced from ISO/IEC 8859-15 [23].

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.
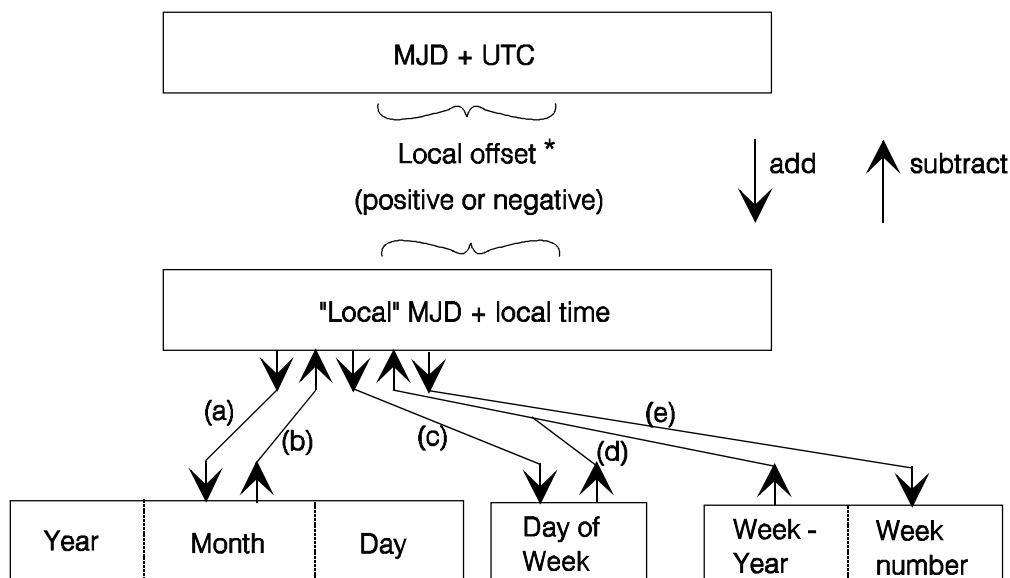
* Offsets are positive for Longitudes East of Greenwich and negative for longitudes West of Greenwich.

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:

MJD	Modified Julian Date
UTC	Universal Time, Co-ordinated
Y	Year from 1900 (e.g. for 2003, Y = 103)
M	Month from January (= 1) to December (= 12)
D	Day of month from 1 to 31
WY	"Week number" Year from 1900
MN	Week number according to ISO 2015 [22]
WD	Day of week from Monday (= 1) to Sunday (= 7)
K, L ,M' , W, Y'	Intermediate variables
x	Multiplication
int	Integer part, ignoring remainder
mod 7	Remainder (0-6) after dividing integer by 7

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)   WN = 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 audio in DVB systems

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

Since the transmission of AC-3 audio streams is optional in DVB systems this annex has an informative status. However, if AC-3 audio streams are transmitted in a DVB system the specifications in this annex shall 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 stream.

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

Component_type byte values (permitted settings)								Description
reserved status 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	reserved
0	X	X	X	X	X	X	X	Interpret b0-b6 as indicated below
	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	0	1	reserved
					1	1	0	reserved
					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 ITU-R Recommendation BS.1196-E - Annex 2 [19]. 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 [19].

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) shall be 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 D.1).

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.3).

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, "01000" (= 8) 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 0-7 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 (0-7) 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: These optional bytes are reserved for future use.

Table D.2: AC-3 descriptor syntax

Syntax	No. of Bits	Identifier
AC-3_ descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	N x 8	uimsbf
component_type_flag	1	bslbf
bsid_flag	1	bslbf
mainid_flag	1	bslbf
asvc_flag	1	bslbf
reserved	4	bslbf
If (component_type_flag)==1{		
AC-3_type	8	uimsbf
}		
If (bsid_flag)==1{		
bsid	8	uimsbf
{		
If (mainid_flag)==1{		
mainid	8	uimsbf
}		
If (asvc_flag)==1{		
asvc	8	bslbf
}		
For(I=0;I<N;I++){		
additional_info[I]	N x 8	uimsbf
}		
}		

Annex E (informative):
Bibliography

- Implementation guidelines for use of telecommunications interfaces in the Digital Broadcasting systems (DVB Project Office).

- ISO/IEC 8859-12: There is currently no draft numbered ISO-8859-12. This number might be reserved for ISCII Indian.

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	January 2003	One-step Approval Procedure OAP 20030523: 2003-01-22 to 2003-05-23
V1.5.1	May 2003	Publication