INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.94

(11/98)

SERIES J: TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER MULTIMEDIA SIGNALS

Ancillary digital services for television transmission

Service information for digital broadcasting in cable television systems

ITU-T Recommendation J.94

(Previously CCITT Recommendation)

ITU-T J-SERIES RECOMMENDATIONS

TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER MULTIMEDIA SIGNALS

General Recommendations	J.1–J.9
General specifications for analogue sound-programme transmission	J.10–J.19
Performance characteristics of analogue sound-programme circuits	J.20–J.29
Equipment and lines used for analogue sound-programme circuits	J.30–J.39
Digital encoders for analogue sound-programme signals	J.40–J.49
Digital transmission of sound-programme signals	J.50–J.59
Circuits for analogue television transmission	J.60–J.69
Analogue television transmission over metallic lines and interconnection with radio-relay links	J.70–J.79
Digital transmission of television signals	J.80–J.89
Ancillary digital services for television transmission	**J.90–J.99**
Operational requirements and methods for television transmission	J.100–J.109
Interactive systems for digital television distribution	J.110–J.129
Transport of MPEG-2 signals on packetised networks	J.130–J.139
Measurement of the quality of service	J.140–J.149
Digital television distribution through local subscriber networks	J.150–J.159

For further details, please refer to ITU-T List of Recommendations.

ITU-T RECOMMENDATION J.94

SERVICE INFORMATION FOR DIGITAL BROADCASTING IN CABLE TELEVISION SYSTEMS

Summary

This Recommendation specifies Service Information (SI) describing the services residing within streams constructed in accordance with ITU-T Rec. H.222.0 | ISO/IEC 13818-1 (MPEG-2 Systems). This Recommendation defines the standard protocol for transmission of the relevant SI data tables carried in the MPEG-2 Transport Stream multiplex.

Source

ITU-T Recommendation J.94 was prepared by ITU-T Study Group 9 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 19th of November 1998.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation the term *recognized operating agency (ROA)* includes any individual, company, corporation or governmental organization that operates a public correspondence service. The terms *Administration*, *ROA* and *public correspondence* are defined in the *Constitution of the ITU (Geneva, 1992).*

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Introduction

The development of new digital technology has reached the point at which it is evident that they enable digital systems to offer significant advantages, in comparison with conventional analogue techniques, in terms of vision and sound quality, spectrum and power efficiency, service flexibility, multimedia convergence and potentially lower equipment costs. Moreover, the use of cable distribution for the delivery of video and audio signals to individual viewers and listeners is continually growing, and has already become the dominant form of distribution in many parts of the world. It is also evident that these potential benefits can best be achieved through the economies of scale resulting from the widespread use of digital systems designed to be easily implementable on existing infrastructure and which take advantage of the many possible synergies with related audiovisual systems.

This Recommendation has three annexes, that provide the specifications for the service information for the three digital television cable systems submitted to the ITU-T.

This reflects the fact that standardization of digital cable television systems is being addressed for the first time by the ITU-T and that a number of systems had been developed and provisionally implemented when this standardization effort was undertaken by the ITU.

Administrations and private operators planning the introduction of digital cable television services are encouraged to consider the use of one of the systems described in Annexes A, B and C, and to seek opportunities for further convergence, rather than developing a different system based on the same technologies.

Recommendation J.94

SERVICE INFORMATION FOR DIGITAL BROADCASTING IN CABLE TELEVISION SYSTEMS

(Geneva, 1998)

1 Scope

The scope of this Recommendation defines the Service Information that conveys the relevant description of the services contained in a multiplex of audio, video, and data that is distributed by cable networks (e.g. CATV systems). Recommendation J.83 defines the transmission characteristics for digital multi-programme signals distributed through cable networks.

NOTE – The service information is specified to be contained within the MPEG-2 transport layer as Program Specific Information (PSI). This mechanism provides some ancillary data capacity in the forward channel, which can be used to accommodate the needs of other services such as program guides (a description of the provision and characteristics of these services is outside the scope of this Recommendation).

Being highly flexible, the MPEG-2 transport layer can be configured to deliver any desired mix of television, sound and data signals (with sound either related or unrelated to the video signal content, and at various possible levels of quality).

This Recommendation is intended to ensure that the designers and operators of cable distribution (e.g. CATV) networks carrying multi-programme signals will have the information they need to be able to establish and maintain fully satisfactory networks. It also provides the information needed by the designers and manufacturers of equipment (including receivers) for digital multi-programme signals distributed by cable networks.

2 References

The following UIT-T Recommendations and other references contain provisions which, through reference in the text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

[1] ITU-T Recommendation H.222.0 (1995) | ISO/IEC 13818-1:1996, *Information technology – Generic coding of moving pictures and associated audio information*: Systems. (Referred to in Annex A.)

[2] ISO 3166:1997, *Codes for the representation of names of countries and their subdivisions*. (Referred to in Annex A.)

[3] ISO 639-2:1998, *Codes for the representation of names of languages – Part 2: Alpha-3 code*. (Referred to in Annex A.)

[4] EBU SPB 492 (1992), *Teletext specification (625 line Television Systems).* (Referred to in Annex A.)

[5] ISO/IEC 8859 (All parts), *Information technology – 8-bit single-byte coded graphic character sets, Latin alphabets*. (Referred to in Annex A.)

[6] ETR 162, *Digital Video Broadcasting (DVB); Allocation of Service Information (SI) codes for DVB systems*. (Referred to in Annex A.)

[7] ETR 211, *Digital Video Broadcasting (DVB); Guidelines on implementation and usage of Service Information (SI)*. (Referred to in Annex A.)

[8] ISO/IEC 10646-1:1993, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*. (Referred to in Annex A.)

[9] ISO/IEC 6937:1994, *Information technology – Coded graphic character set for text communication – Latin alphabet*. (Referred to in Annex A.)

[10] ETR 289, *Digital Video Broadcasting (DVB); Support for use of scrambling and Conditional Access (CA) within digital broadcasting systems*. (Referred to in Annex A.)

[11] IEC Publication 61883 (1998), *Consumer audio/video equipment – Digital interface*. (Parts 1 and 4.) (Referred to in Annex A.)

[12] ETR 154, *Digital Video Broadcasting (DVB); Implementation guidelines for the use of MPEG-2 Systems, Video and Audio in satellite, cable and terrestrial broadcasting applications*. (Referred to in Annex A.)

[13] IEEE 1394, *High Performance Serial Bus*. (Referred to in Annex A.)

[14] ITU-T Recommendation H.222.0 (1995) | ISO/IEC 13818-1:1996, *Information technology – Generic coding of moving pictures and associated audio information: Systems*.

[15] ETS 300 468 (1997), *Digital broadcasting systems for television, sound and data services; Specification for Service Information (SI) in Digital Video Broadcasting (DVB) systems.*

3 Terms and definitions

This Recommendation defines the following terms:

3.1 bouquet: A collection of services marketed as a single entity.

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

3.3 component (ELEMENTARY Stream): One or more entities which together make up an event, e.g. video, audio, teletext.

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

3.5 delivery system: The 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.

3.6 descriptor: A data structure of the format: descriptor_tag, descriptor_length, and a variable amount of data. The tag and length fields are each 8 bits. The length specifies the length of data that begins immediately following the descriptor_length field itself. A descriptor whose descriptor_tag identifies a type not recognized by a particular decoder shall be ignored by that decoder. Descriptors can be included in certain specified places within PSIP tables, subject to certain restrictions. Descriptors may be used to extend data represented as fixed fields within the tables. They make the protocol very flexible since they can be included only as needed. New descriptor types can be standardized and included without affecting receivers that have not been designed to recognize and process the new types.

3.7 digital channel: A set of one or more digital elementary streams. See virtual channel.

3.8 Entitlement Management Messages (EMM): Are private Conditional Access information which specify the authorization levels or the services of specific decoders. They may be addressed to individual decoder or groups of decoders.

3.9 event: A 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.

3.10 forbidden: The term "forbidden" when used in the clauses defining the coded bitstream, indicates that the value shall never be used.

3.11 instance: See table instance.

3.12 logical channel: See virtual channel.

3.13 MPEG-2: Refers to ISO/IEC standard 13818 (All parts). Systems coding is defined in Part 1. Video coding is defined in Part 2. Audio coding is defined in Part 3.

3.14 **message**: The more general term *message* is used interchangeably with *section*, especially to refer to non-table-oriented data structures such as, for example, the SYSTEM TIME message. Likewise, the term *message* is used to refer to a data structure that may deliver portions of various types of tables. The NETWORK INFORMATION message, for example, defines portions of several types of network tables.

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

3.16 **network**: A collection of MPEG-2 Transport Stream (TS) multiplexes transmitted on a single delivery system, e.g. all digital channels on a specific cable system.

3.17 **original_network_id**: A unique identifier of a network.

3.18 **physical channel**: A generic term to refer to the each of the 6-8 MHz frequency bands where television signals are embedded for transmission. Also known as the Physical Transmission Channel (PTC). One analog virtual channel fits in one PTC but multiple digital virtual channels typically coexist in one PTC.

3.19 **physical transmission channel**: See physical channel.

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

3.21 **program element**: A generic term for one of the elementary streams or other data streams that may be included in a program. For example: audio, video, data, etc.

3.22 **reserved**: The term "reserved" when used in the clause defining the coded bitstream, indicates that the value may be used in the future for ISO defined extensions. Unless otherwise specified within this Recommendation, all "reserved" bits shall be set to "1".

3.23 **reserved_future_use**: The term "reserved_future_use", when used in the clause defining the coded bitstream, indicates that the value may be used in the future for ETSI defined extensions. Unless otherwise specified within this Recommendation all "reserved_future_use" bits shall be set to "1".

3.24 **section**: A section is a syntactic structure used for mapping all service information defined in this Recommendation into ITU-T Rec. H.222.0 | ISO/IEC 13818-1 [1] TS packets.

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

3.26 **service_id**: A unique identifier of a service within a TS.

3.27 **Service Information (SI)**: Digital data describing the delivery system, content and scheduling/timing of broadcast data streams, etc. It includes MPEG-2 PSI together with independently defined extensions.

3.28 **stream**: An ordered series of bytes. The usual context for the term *stream* is the series of bytes extracted from Transport Stream packet payloads which have a common unique PID value (e.g. video PES packets or Program Map Table sections).

3.29 **sub_table**: A sub_table is 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.

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".

3.30 **table**: A table is comprised of a number of sub_tables with the same value of table_id.

3.31 **table instance**: Tables are identified by the table_id field. However, in cases such as the RRT and EIT, several instances of a table are defined simultaneously. All instances have the same PID and table_id but different table_id_extension.

3.32 **Transport Stream (TS)**: A TS is a data structure defined in ITU-T Rec. H.222.0 | ISO/IEC 13818-1 [1]. It is the basis of the DVB standards.

3.33 **transport_stream_id**: A unique identifier of a TS within an original network.

3.34 **virtual channel**: A virtual channel is the designation, usually a number, that is recognized by the user as the single entity that will provide access to an analog TV program or a set of one or more digital elementary streams. It is called "virtual" because its identification (name and number) may be defined independently from its physical location. Examples of virtual channels are: digital radio (audio only), a typical analog TV channel, a typical digital TV channel (composed of one audio and one video stream), multi-visual digital channels (composed of several video streams and one or more audio tracks), or a data broadcast channel (composed of one or more data streams). In the case of an analog TV channel, the virtual channel designation will link to a specific physical transmission channel. In the case of a digital TV channel, the virtual channel designation will link both to the physical transmission channel and to the particular video and audio streams within that physical transmission channel.

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

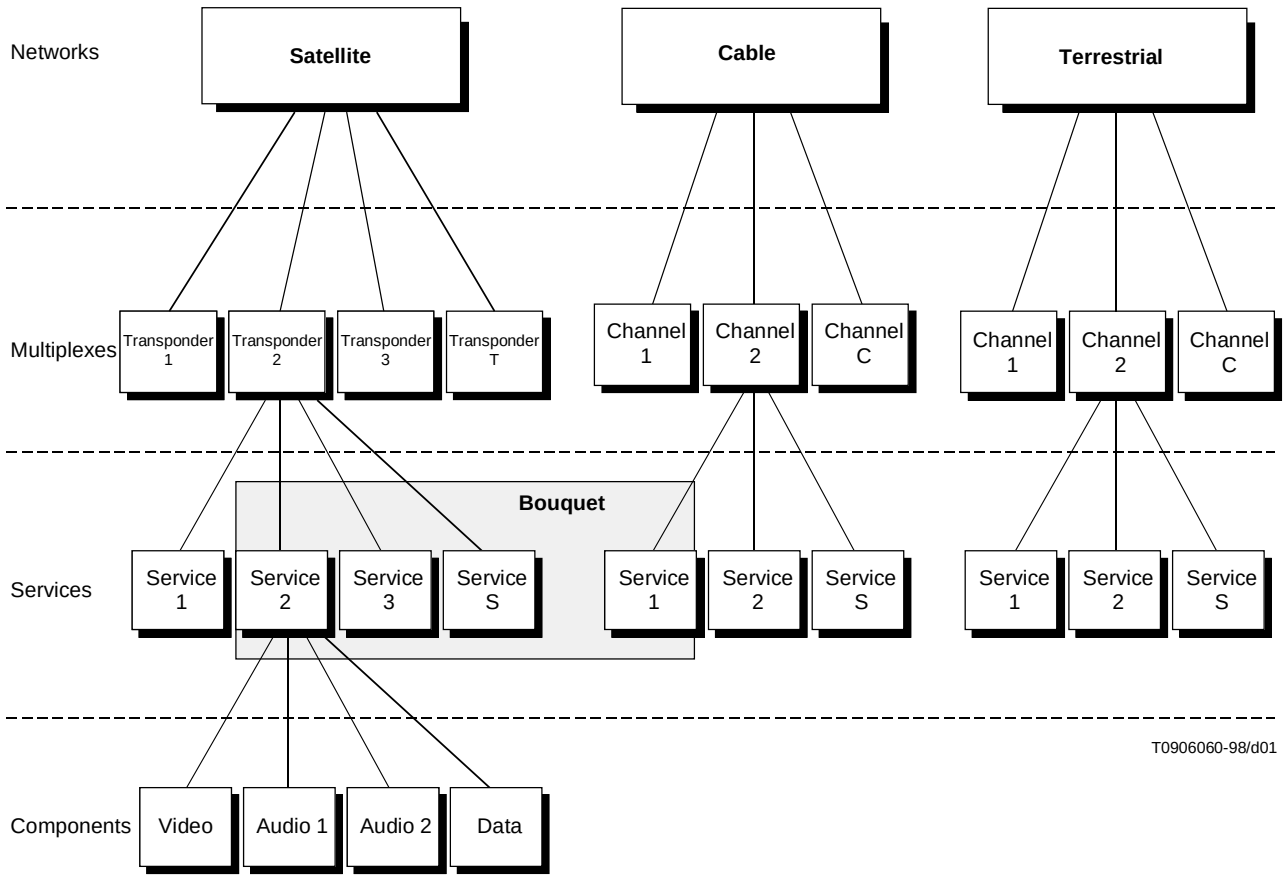


Figure 1/J.94 – Digital broadcasting, service delivery model

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations:

ATSC Advanced Television Systems Committee

BAT Bouquet Association Table

BCD Binary Coded Decimal

bslbf bit string, left bit first

CA Conditional Access

CAT Conditional Access Table

CDT Carrier Definition Table

CLUT Colour Look-Up Table

CRC Cyclic Redundancy Check

CVCT Cable Virtual Channel Table

DIT Discontinuity Information Table

DTV Digital Television

DVB Digital Video Broadcasting

EBU European Broadcasting Union

ECM Entitlement Control Message

EIT Event Information Table

EMM Entitlement Management Message

EPG Electronic Programme Guide

ETM Extended Text Message

ETS European Telecommunication Standard

ETT Extended Text Table

FEC Forward Error Correction

GA Grand Alliance

GMT Greenwich Mean Time

GPS Global Positioning System

IEC International Electrotechnical Commission

IRD Integrated Receiver-Decoder

ISO International Organization for Standardization

LSB Least Significant Bit

MCPT Multiple Carriers per Transponder

MGT Master Guide Table

MJD Modified Julian Date

MMT Modulation Mode Table

MPAA Motion Picture Association of America

MPEG Moving Pictures Expert Group

NIT Network Information Table

NVOD Near Video-on-Demand

PAT Program Association Table

PCR Program Clock Reference

PES Packetized Elementary Stream

PID Packet identifier

PMT Program Map Table

PSI Program Specific Information

PSIP Program and Service Information Protocol

PSTN Public Switched Telephone Network

PTC Physical Transmission Channel

PTS Presentation Time Stamp

QAM Quadrature Amplitude Modulation

QPSK Quaternary Phase Shift Keying

rpchof remainder polynomial coefficients, highest order first

RRT Rating Region Table

RS Reed-Solomon

RST Running Status Table

SCTE Society of Cable Telecommunications Engineers

SDT Service Description Table

SECAM Sequential colour with memory (*Séquentiel Couleur avec Mémoire*)

SI Service Information

SIT Satellite Information Table

SMI Storage Media Interoperability

ST Stuffing Table

STD System Target Decoder

STT System Time Table

TAI International Atomic Time[1]

TDT Time and Date Table (referred to in Annex A)

TNT Transponder Name Table

TOT Time Offset Table

TS Transport Stream

TVCT Terrestrial Virtual Channel Table

uimsbf unsigned integer most significant bit first

UTC Universal Time coordinated

VCN Virtual Channel Number

VCT Virtual Channel Table Used in reference to either TVCT or CVCT

---

1 French acronym used.

Annex A

Service Information for digital multi-programme System A

A.1 Scope

This annex is derived from work done in Europe and is based upon the European Telecommunication Standard (ETS) 300 468. It specifies the Service Information (SI) data which forms a part of Digital Video Broadcasting (DBV) 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 ITU-T Rec. H.222.0 | ISO/IEC 13818-1 [1] as Program Specific Information (PSI). This annex specifies additional data which complement 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 this annex, 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 SI specification, but the data contained within the SI specified here may be used as the basis for an EPG.

Rules of operation for the implementation of this annex are specified in ETR 211 [7].

A.2 References

For references, see clause 2.

A.3 Definitions and abbreviations

The terms and definitions, and abbreviations can be found in clauses 3 and 4 respectively.

A.4 Service Information (SI) description

ITU-T Rec. H.222.0 | 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 this annex) 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 this annex in compliance with ITU-T Rec. H.222.0 | ISO/IEC 13818-1 [1], but the data format is outside the scope of ITU-T Rec. H.222.0 | 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 this annex.

In addition to the PSI, data are needed to provide identification of services and events for the user. The coding of this data is defined in this annex. 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 this annex can also provide information on services and events carried by different multiplexes, and even on other networks. These data are 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. See Figure A.1.

A.5 Service Information (SI) tables

A.5.1 SI table mechanism

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

The tables listed in A.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 A.5.1.5).

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

These SI syntactic structures conform to the private section syntax defined in ITU-T Rec. H.222.0 | ISO/IEC 13818-1 [1].

A.5.1.1 Explanation

Sections may be variable in length. The sections within each table are limited to 1024 bytes in length, except for sections within the EIT which are limited to 4096 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 A.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 A.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 this annex, section numbering applies to sub_tables.

d) Version_number:

When the characteristics of the TS described in the SI given in this annex 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 this annex, 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.

A.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, refer to 2.4.4 and Annex C of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 [1].

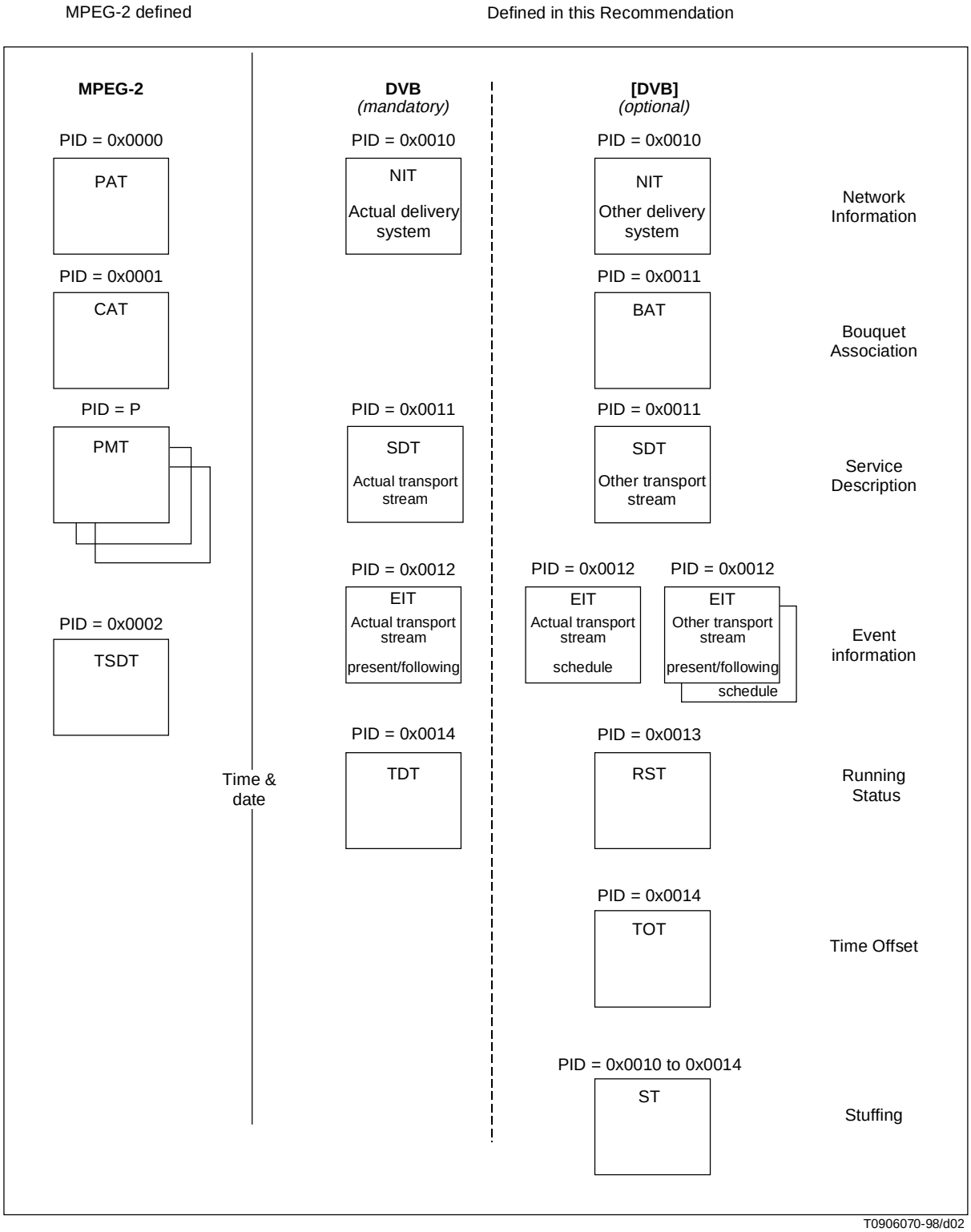


Figure A.1/J.94 – General organization of the Service Information (SI)

A.5.1.3 Coding of PID and table_id fields

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

Table A.1/J.94 – 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 0x001D
DIT	0x001E
SIT	0x001F

Table A.2 lists the values which shall be used for table_id for the service information, defined in this annex.

A.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 this annex several times, even when changes do not occur in the configuration.

For SI specified within this annex, 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.

A.5.1.5 Scrambling

With the exception of the EIT carrying schedule information, all tables specified in this annex shall not be scrambled. One method for scrambling the EIT schedule table is given in Appendix A.II, 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 ITU-T Rec. H.222.0 | 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.

A.5.2 Table definitions

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

NOTE – The symbols and abbreviations, and the method of describing syntax used in this annex are the same as those defined in 2.2 and 2.3 of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 [1].

Table A.2/J.94 – 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

A.5.2.1 Network Information Table (NIT)

The NIT (see Table A.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 ETR 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 A.2).

The NIT shall be segmented into network_information_sections using the syntax of Table A.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 value 0x40 with the network_id field taking 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 A.3/J.94 – 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 A.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 1021 so that the entire section has a maximum length of 1024 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 A.B after processing the entire section.

A.5.2.2 Bouquet Association Table (BAT)

The BAT (see Table A.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 A.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 A.4/J.94 – 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 A.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 1021 so that the entire section has a maximum length of 1024 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 A.B after processing the entire private section.

A.5.2.3 Service Description Table (SDT)

Each sub_table of the SDT (see Table A.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 A.2).

The SDT shall be segmented into service_description_sections using the syntax of Table A.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, and any sections of an SDT which refer to a TS other than the actual TS shall take a table_id value of 0x46.

Table A.5/J.94 – 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 A.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 1021 so that the entire section has a maximum length of 1024 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 ETR 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 ETR 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 A.6.

Table A.6/J.94 – 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 A.B after processing the entire section.

A.5.2.4 Event Information Table (EIT)

The EIT (see Table A.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 A.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".

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 A.7. Any sections forming part of an EIT shall be transmitted in TS packets with a PID value of 0x0012.

Table A.7/J.94 – 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 A.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 4093 so that the entire section has a maximum length of 4096 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 A.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 Coordinated (UTC) and Modified Julian Date (MJD) (see Appendix A.I). This field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 digits in 4-bit Binary Coded Decimal (BCD). If the start time is undefined (e.g. for an event in a NVOD reference service), all bits of the field are set to "1".

Example 1 – 93/10/13 12:45:00 is coded as "0xC079124500".

duration: A 24-bit field containing the duration of the event in hours, minutes and seconds.

format: 6 digits, 4-bit BCD = 24 bit.

Example 2 – 01:45:30 is coded as "0x014530".

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

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

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

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex A.B after processing the entire private section.

A.5.2.5 Time and Date Table (TDT)

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

The TDT shall consist of a single section using the syntax of Table A.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 A.8/J.94 – 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 A.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 Appendix A.I). 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".

A.5.2.6 Time Offset Table (TOT)

The TOT (see Table A.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 A.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 A.9/J.94 – 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 A.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 Appendix A.I). This field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 digits in 4-bit BCD.

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

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

CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex A.B after processing the entire private section.

A.5.2.7 Running Status Table (RST)

The RST (see Table A.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.

Table A.10/J.94 – 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
}		
}		

The RST shall be segmented into running_status_sections using the syntax of Table A.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.

Semantics for the running status section

table_id: See Table A.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 1021 so that the entire section has a maximum length of 1024 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 A.6.

A.5.2.8 Stuffing Table (ST)

The purpose of this section (see Table A.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 A.11/J.94 – 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 A.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 4093 so that the entire section has a maximum length of 4096 bytes.

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

A.5.2.9 Discontinuity Information Table (DIT)

See A.7.1.1

A.5.2.10 Selection Information Table (SIT)

See A.7.1.2

A.6 Descriptors

This subclause describes the different descriptors that can be used within the SI (for further information, refer to ETR 211 [7]).

A.6.1 Descriptor identification and location

Table A.12 lists the descriptors defined within this annex, 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 A.12/J.94 – Possible locations of descriptors

Descriptor	Tag value	NIT	BAT	SDT	EIT	TOT	PMT	SIT (Note 1)
network_name_descriptor	0x40	*	–	–	–	–	–	–
service_list_descriptor	0x41	*	*	–	–	–	–	–
stuffing_descriptor	0x42	*	*	*	*	–	–	*
satellite_delivery_system_descriptor	0x43	*	–	–	–	–	–	–
cable_delivery_system_descriptor	0x44	*	–	–	–	–	–	–
Reserved for future use	0x45	–	–	–	–	–	–	–
Reserved for future use	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	0x63	–	–	–	–	–	–	*
data_broadcast_descriptor	0x64	–	–	*	*	–	–	*
CA_system_descriptor (Note 2)	0x65	–	–	–	–	–	*	–
data_broadcast_id_descriptor	0x66	–	–	–	–	–	*	–
Reserved for future use	0x67 to 0x7F							
User-defined	0x80 to 0xFE							
Forbidden	0xFF							

***** Possible location.

NOTE 1 – Only found in Partial Transport Streams.

NOTE 2 – Reserved for DAVIC/DVB use: DAVIC shall define its use.

A.6.2 Descriptor coding

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

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

descriptor_tag: The descriptor tag is an 8-bit field which identifies each descriptor. Those values with MPEG-2 normative meaning are described in ITU-T Rec. H.222.0 | ISO/IEC 13818-1 [1]. The values of descriptor_tag are defined in Table A.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.

A.6.2.1 Bouquet name descriptor

The bouquet name descriptor provides the bouquet name in text form, see Table A.13.

Table A.13/J.94 – 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.A.

A.6.2.2 CA identifier descriptor

The CA identifier descriptor (see Table A.14) 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 A.14/J.94 – 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].

A.6.2.3 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 A.15).

Table A.15/J.94 – 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 A.16.

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 A.16.

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 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 specify a text description of the component stream.

Text information is coded using the character sets and methods described in Annex A.A.

Table A.16/J.94 – 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
0x01	0x02	Video, 16:9 aspect ratio with pan vectors
0x01	0x03	Video, 16:9 aspect ratio without pan vectors
0x01	0x04	Video, >16:9 aspect ratio
0x01	0x05 to 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, multilingual, multichannel
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 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 0xFF	Reserved for future use
0x04 to 0x0B	0x00 to 0xFF	Reserved for future use
0x0C to 0x0F	0x00 to 0xFF	User-defined

A.6.2.4 Content descriptor

The intention of the content descriptor (see Table A.17) is to provide classification information for an event.

Table A.17/J.94 – 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 A.18.

content_nibble_level_2: This 4-bit field represents the second level of a content identifier. This field shall be coded according to Table A.18.

user_nibble: This 4-bit field is defined by the broadcaster.

Table A.18/J.94 – 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

Table A.18/J.94 – content_nibble level 1 and 2 assignments *(continued)*

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

Table A.18/J.94 – content_nibble level 1 and 2 assignments *(continued)*

content_nibble_level_1	content_nibble_level_2	Description
		Arts/Culture (without music):
0x7	0x0	Arts/culture (without music, general)
0x7	0x1	Performing arts
0x7	0x2	Fine arts
0x7	0x3	Religion
0x7	0x4	Popular culture/traditional arts
0x7	0x5	Literature
0x7	0x6	Film/cinema
0x7	0x7	Experimental film/video
0x7	0x8	Broadcasting/press
0x7	0x9	New media
0x7	0xA	Arts/culture magazines
0x7	0xB	Fashion
0x7	0xC to 0xE	Reserved for future use
0x7	0xF	User-defined
		Social/Political issues/Economics:
0x8	0x0	Social/political issues/economics (general)
0x8	0x1	Magazines/reports/documentary
0x8	0x2	Economics/social advisory
0x8	0x3	Remarkable people
0x8	0x4 to 0xE	Reserved for future use
0x8	0xF	User-defined
		Education/ Science/Factual topics:
0x9	0x0	Education/science/factual topics (general)
0x9	0x1	Nature/animals/environment
0x9	0x2	Technology/natural sciences
0x9	0x3	Medicine/physiology/psychology
0x9	0x4	Foreign countries/expeditions
0x9	0x5	Social/spiritual sciences
0x9	0x6	Further education
0x9	0x7	Languages
0x9	0x8 to 0xE	Reserved for future use
0x9	0xF	User-defined
		Leisure hobbies:
0xA	0x0	Leisure hobbies (general)
0xA	0x1	Tourism/travel
0xA	0x2	Handicraft
0xA	0x3	Motoring
0xA	0x4	Fitness & health
0xA	0x5	Cooking
0xA	0x6	Advertisement/shopping
0xA	0x7	Gardening
0xA	0x8 to 0xE	Reserved for future use
0xA	0xF	User-defined

Table A.18/J.94 – content_nibble level 1 and 2 assignments *(concluded)*

content_nibble_level_1	content_nibble_level_2	Description
		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

A.6.2.5 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 A.19).

Table A.19/J.94 – 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

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

country_code: This 24-bit field identifies a country using the 3-character code as specified in ISO 3166 [2].

Each character is coded into 8-bits according to ISO 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'.

A.6.2.6 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 A.20).

Table A.20/J.94 – 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 are broadcasted.

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 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 specify the text description of the data component.

Text information is coded using the character sets and methods described in Annex A.A.

A.6.2.7 Data broadcast id descriptor

The data broadcast id descriptor identifies the type of the data component (see Table A.21). It is a short form of the broadcast descriptor and it may be placed in the component loop of the PSI PMT table.

Table A.21/J.94 – 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
}		

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].

A.6.2.8 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.

A.6.2.8.1 Cable delivery system descriptor

See Table A.22.

Table A.22/J.94 – 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 A.23.

Table A.23/J.94 – 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 A.24.

Table A.24/J.94 – 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 A.25.

Table A.25/J.94 – 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
1111	No conv. Coding
0110 to 1110	Reserved for future use

A.6.2.8.2 Satellite delivery system descriptor

See Table A.26.

Table A.26/J.94 – 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 degrees).

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 A.27).

Table A.27/J.94 – 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 A.28.

Table A.28/J.94 – Modulation scheme for satellite

Modulation bit 4 3210	Description
0 0000	Not defined
0 0001	QPSK
0 0010 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 A.25.

A.6.2.8.3 Terrestrial delivery system descriptor

See Table A.29.

Table A.29/J.94 – 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. See Table A.30.

Table A.30/J.94 – Signalling format for the bandwidth

bandwidth	Bandwidth value
000	8 MHz
001	7 MHz
010 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 A.31.

Table A.31/J.94 – Signalling format for the possible constellation patterns

constellation	Constellation characteristics
00	QPSK
01	16-QAM
10	64-QAM
11	Reserved for future use

hierarchy_information: The hierarchy_information specifies whether the transmission is hierarchical and, if so, what the α value is. See Table A.32.

Table A.32/J.94 – Signalling format for the α values

hierarchy_information	α value
000	Non-hierarchical
001	α = 1
010	α = 2
011	α = 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 A.33. Non-hierarchical channel coding and modulation requires signalling of one code rate. In this case, 3 bits specifying code_rate according to Table A.34 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 A.33/J.94 – Signalling format for each of the code rates

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

guard_interval: The guard_interval is a 2-bit field specifying the guard interval values. See Table A.34.

Table A.34/J.94 – 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. See Table A.35.

Table A.35/J.94 – 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:

– 0: no other frequency in use.

– 1: one or more other frequencies in use.

A.6.2.9 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. See Table A.36.

Table A.36/J.94 – 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 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.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.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.A.

A.6.2.10 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. See Table A.37.

Table A.37/J.94 – 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 A.38.

Table A.38/J.94 – 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.

A.6.2.11 Linkage descriptor

The linkage descriptor (see Table A.39) identifies a service that can be presented if the consumer requests for additional information related to a specific entity described by the SI system. The location of the linkage descriptor in the syntax indicates the entity for which additional information is available. For example a linkage descriptor located within the NIT shall point to a service providing additional information on the network, a linkage descriptor in the BAT shall provide a link to a service informing about the bouquet, etc.

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

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

Table A.39/J.94 – 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
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 A.40).

Table A.40/J.94 – 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 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.

A.6.2.12 Local time offset descriptor

The local time offset descriptor (see Table A.41) may be used in the TOT to describe country specific dynamic changes of the local time offset relative to UTC.

Table A.41/J.94 – 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 8859-1 [5] and inserted in order into the 24-bit field.

In the case where 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. See Table A.42.

Table A.42/J.94 – 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 (most westerly region)
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 +12 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 Appendix A.I) 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 –12hours and +12hours 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.

A.6.2.13 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 A.43.

Table A.43/J.94 – 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
}		

Table A.43/J.94 – Mosaic descriptor *(end)*

Syntax	No. of bits	Identifier
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 A.44 for coding.

Table A.44/J.94 – 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 A.45 for coding.

Table A.45/J.94 – 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 A.2) 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**

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 A.2/J.94 – 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 A.46.

Table A.46/J.94 – Coding of logical_cell_presentation_info

Value	Meaning
0x00	Undefined
0x01	Video
0x02	Still picture (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 A.47 for coding.

Table A.47/J.94 – 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 A.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.

A.6.2.14 Multilingual bouquet name descriptor

The multilingual bouquet name descriptor (see Table A.48) provides the bouquet name in text form in one or more languages.

Table A.48/J.94 – 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 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.A.

A.6.2.15 Multilingual component descriptor

The multilingual component descriptor (see Table A.49) provides a text description of a component in one or more languages. The component is identified by its component tag value.

Table A.49/J.94 – 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 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 specify a text description of the component stream. Text information is coded using the character sets and methods described in Annex A.A.

A.6.2.16 Multilingual network name descriptor

The multilingual network name descriptor (see Table A.50) provides the network name in text form in one or more languages.

Table A.50/J.94 – 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 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.A.

A.6.2.17 Multilingual service name descriptor

The multilingual service name descriptor (see Table A.51) provides the names of the service provider and service in text form in one or more languages.

Table A.51/J.94 – 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 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.A.

A.6.2.18 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 A.52) 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 A.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 A.52/J.94 – 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.

A.6.2.19 Network name descriptor

The network name descriptor provides the network name in text form (see Table A.53).

Table A.53/J.94 – 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.A.

A.6.2.20 Parental rating descriptor

This descriptor (see Table A.54) gives a rating based on age and allows for extensions based on other rating criteria.

Table A.54/J.94 – 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 8859-1 [5] and inserted in order into the 24-bit field.

In the case where 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 found in ETR 162 [6].

Example – 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 A.55, giving the recommended minimum age in years of the end user.

Table A.55/J.94 – Parental rating descriptor, rating

Rating	Description
0x00	Undefined
0x01 to 0x0F	Minimum age = rating + 3 years
0x10 to 0xFF	Defined by the broadcaster

Example – 0x04 implies that end users should be at least 7 years old.

A.6.2.21 Partial Transport Stream (TS) descriptor

See A.7.2.1

A.6.2.22 Private data specifier descriptor

This descriptor is used to identify the specifier of any private descriptors or private fields within descriptors. See Table A.56.

Table A.56/J.94 – 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].

A.6.2.23 Short smoothing buffer descriptor

A smoothing_buffer_descriptor is specified in ITU-T Rec. H.222.0 | 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 ITU-T Rec. H.222.0 | 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.

See Table A.57.

Table A.57/J.94 – 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 A.58.

Table A.58/J.94 – Smoothing buffer size

Value	Buffer size (bytes)
0	DVB_reserved
1	1536
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 A.59.

Table A.59/J.94 – 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

Table A.59/J.94 – Smoothing buffer leak rate *(end)*

Value	Leak rate (Mbit/s)	
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	((value) - 16) × 0.5	(7.0, 7.5, 8.0 Mbit/s)
33-37	((value) - 24)	(9, 10, 11, 12, 13 Mbit/s)
38	13.5	
39-43	((value) - 25)	(14, 15, 16, 17, 18 Mbit/s)
44-47	((value) - 34) × 2	(20, 22, 24, 26 Mbit/s)
48	27	
49-55	((value) - 35) × 2	(28, 30, 32 ... 40 Mbit/s)
56	44	
57	48	
58	54	
59	72	
60	108	
61-63	DVB_reserved	

A.6.2.24 Service descriptor

The service descriptor (see Table A.60) provides the names of the service provider and the service in text form together with the service_type.

Table A.60/J.94 – 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 A.61.

Table A.61/J.94 – 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 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.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.

A.6.2.25 Service list descriptor

The service list descriptor (see Table A.62) provides a means of listing the services by service_id and service type.

Table A.62/J.94 – 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 A.61.

A.6.2.26 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. See Table A.63.

Table A.63/J.94 – 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.

A.6.2.27 Short event descriptor

The short event descriptor provides the name of the event and a short description of the event in text form (see Table A.64).

Table A.64/J.94 – 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 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 specify the event name.

Text information is coded using the character sets and methods described in Annex A.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.A.

A.6.2.28 Stream identifier descriptor

The stream identifier descriptor (see Table A.65) 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 A.65/J.94 – 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.

A.6.2.29 Stuffing descriptor

The stuffing descriptor provides a means of invalidating previously coded descriptors or inserting dummy descriptors for table stuffing (see Table A.66).

Table A.66/J.94 – 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.

A.6.2.30 Subtitling descriptor

In the ITU-T Rec. H.222.0 | 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 A.67.

Table A.67/J.94 – 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 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 A.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.

A.6.2.31 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 narrow-band interactive channels. Further information is given in "Implementation guidelines" for the use of telecommunications interfaces in Digital Video Broadcasting systems (see Appendix A.II, Bibliography).

The telephone descriptor syntax is specified in Table A.68.

Table A.68/J.94 – 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 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 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 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 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 8859-1 [5] gives one alphanumeric character of the core number.

A.6.2.32 Teletext descriptor

The Teletext descriptor (see Table A.69) 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 A.69/J.94 – 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 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 A.70.

Table A.70/J.94 – Teletext descriptor, teletext_type

teletext_type	Description
0x00	Reserved for future use
0x01	Initial Teletext page
0x02	Teletext subtitle page
0x03	Additional information page
0x04	Programme schedule page
0x05	Teletext subtitle page for hearing impaired people
0x06 to 0x1F	Reserved for future use

teletext_magazine_number: This is a 3-bit field which identifies the magazine number as defined in EBU SPB 492 [4].

teletext_page_number: This is an 8-bit field giving two 4-bit hex digits identifying the page number as defined in EBU SPB 492 [4].

A.6.2.33 Time-shifted event descriptor

The time-shifted event descriptor (see Table A.71) is used in place of the short_event_descriptor to indicate an event which is a time-shifted copy of another event.

Table A.71/J.94 – 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.

A.6.2.34 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 A.72).

Table A.72/J.94 – 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.

A.7 Storage Media Interoperability (SMI) measures

IEC Publication 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 subclause on Storage Media Interoperability (SMI) describes the SI and PSI required in the delivered data in these cases.

A.7.1 SMI tables

The SMI tables are encoded using the private section syntax defined in ITU-T Rec. H.222.0 | ISO/IEC 13818-1 [1].

The SIT may be up to 4096 bytes long.

The bitstream presented at a digital interface shall either be a "complete" TS conforming to ETR 154 [12] and with SI conforming to this annex or it shall be "partial" TS.

In the latter case, the SI and PSI shall conform to the following subclauses.

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 & DIT is restricted to partial TSs, they shall not be used in broadcasts.

A.7.1.1 Discontinuity Information Table (DIT)

The DIT (see Table A.73) is to be inserted at transition points at which SI information may be discontinuous.

Table A.73/J.94 – 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 discontinuity information section

table_id: See Table A.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.

A.7.1.2 Selection Information Table (SIT)

The SIT describes the service(s) and event(s) carried by the "partial" TS. See Table A.74.

Table A.74/J.94 – 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 A.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 4093 so that the entire section has a maximum length of 4096 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 ETR 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 ITU-T Rec. H.222.0 | ISO/IEC 13818-1 [1] after processing the entire section.

A.7.2 SMI descriptors

This subclause contains syntax and semantics for descriptors exclusively found in partial TSs.

A.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. See Table A.75.

Table A.75/J.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.A

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.A.1 Control codes

The codes in the range 0x80 to 0x9F are assigned to control functions as shown in Table A.A.1.

Table A.A.1/J.94 – 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.A.2.

Table A.A.2/J.94 – 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.A.2 Selection of character table

Text fields can optionally start with non-spacing, non-displayed data which specify the alternative character table to be used for the remainder of the text item. The selection of character table is indicated as follows:

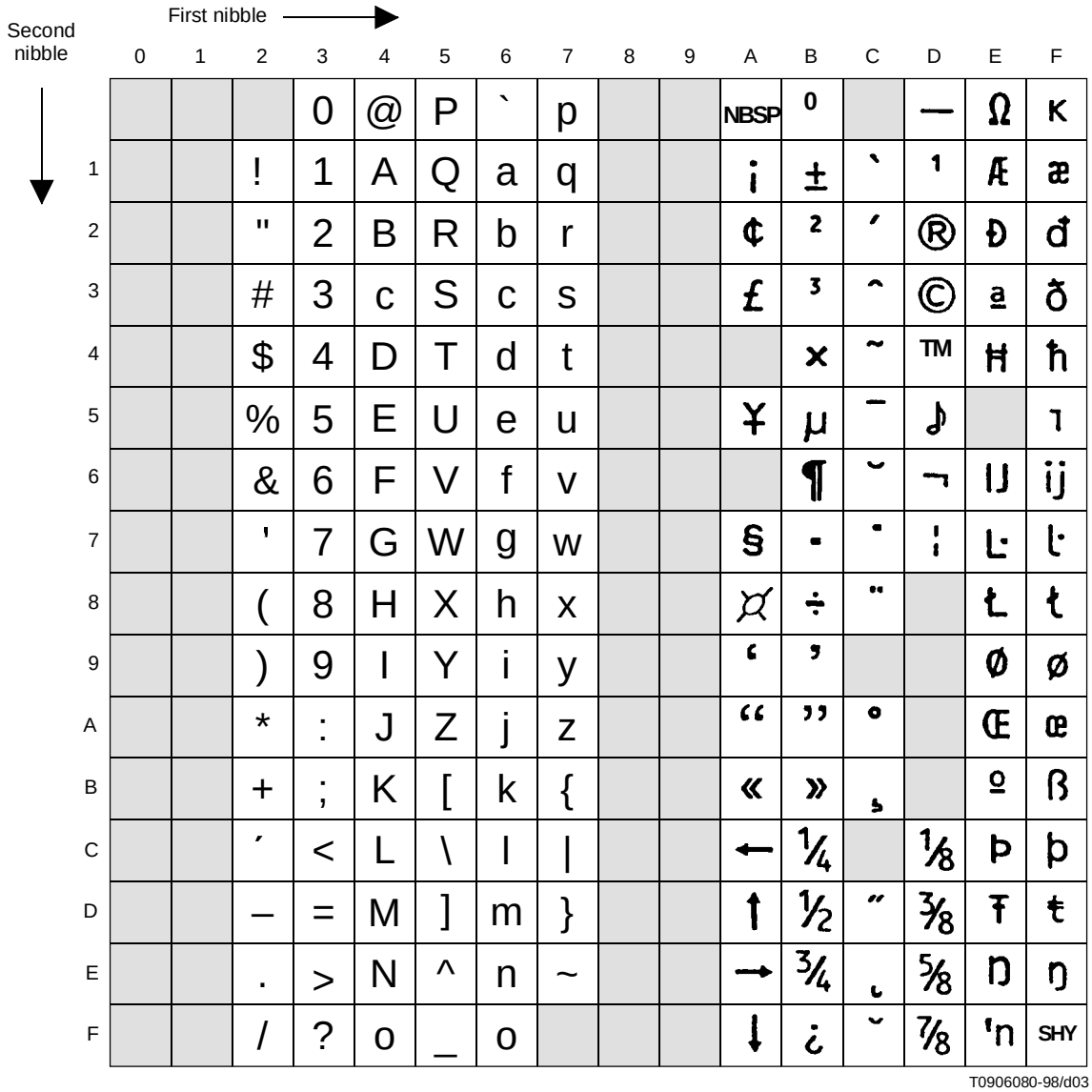
– if the first byte of the text field has a value in the range "0x20" to "0xFF", then this and all subsequent bytes in the text item are coded using the default character coding table (table 00 – Latin alphabet) of Figure A.A.1;

– if the first byte of the text field has a value in the range "0x01" to "0x05", then the remaining bytes in the text item are coded in accordance with character coding tables 01 to 05 respectively, which are given in Figures A.A.2 to A.A.6 respectively;

– 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/IEC 8859 [5], Parts 1 to 9;

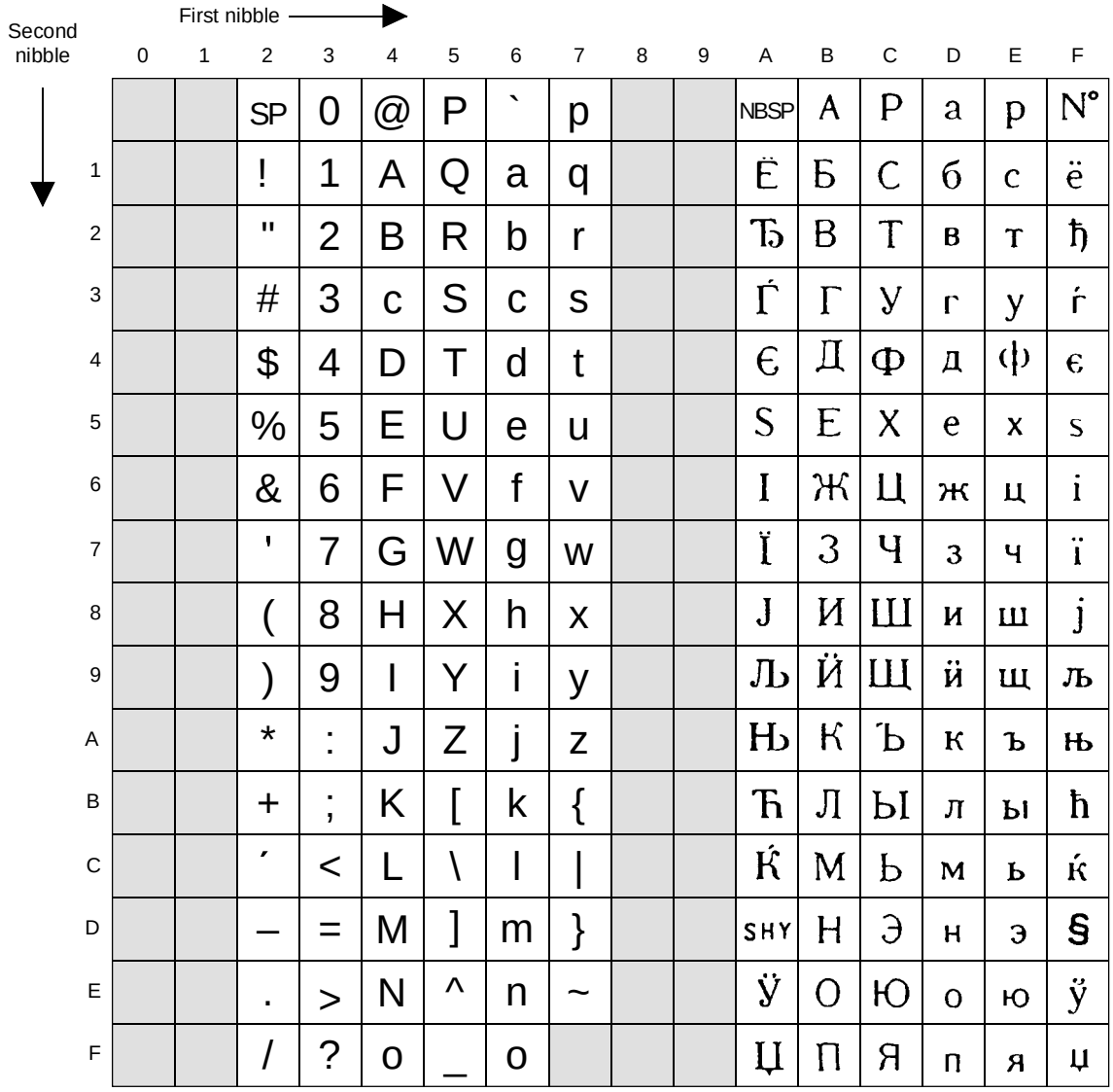
– if the first byte of the text field has a value "0x11", then the remaining bytes in the text item are coded in pairs in accordance with the Basic Multilingual Plane of ISO/IEC 10646-1 [8].

Values for the first byte of "0x00", "0x06" to "0x0F", and "0x12" to "0x1F" are reserved for future use.



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 – Table reproduced from ISO/IEC 6937 (1994) [9].
NOTE 5 – All characters in column C are non-spacing characters (diacritical marks).

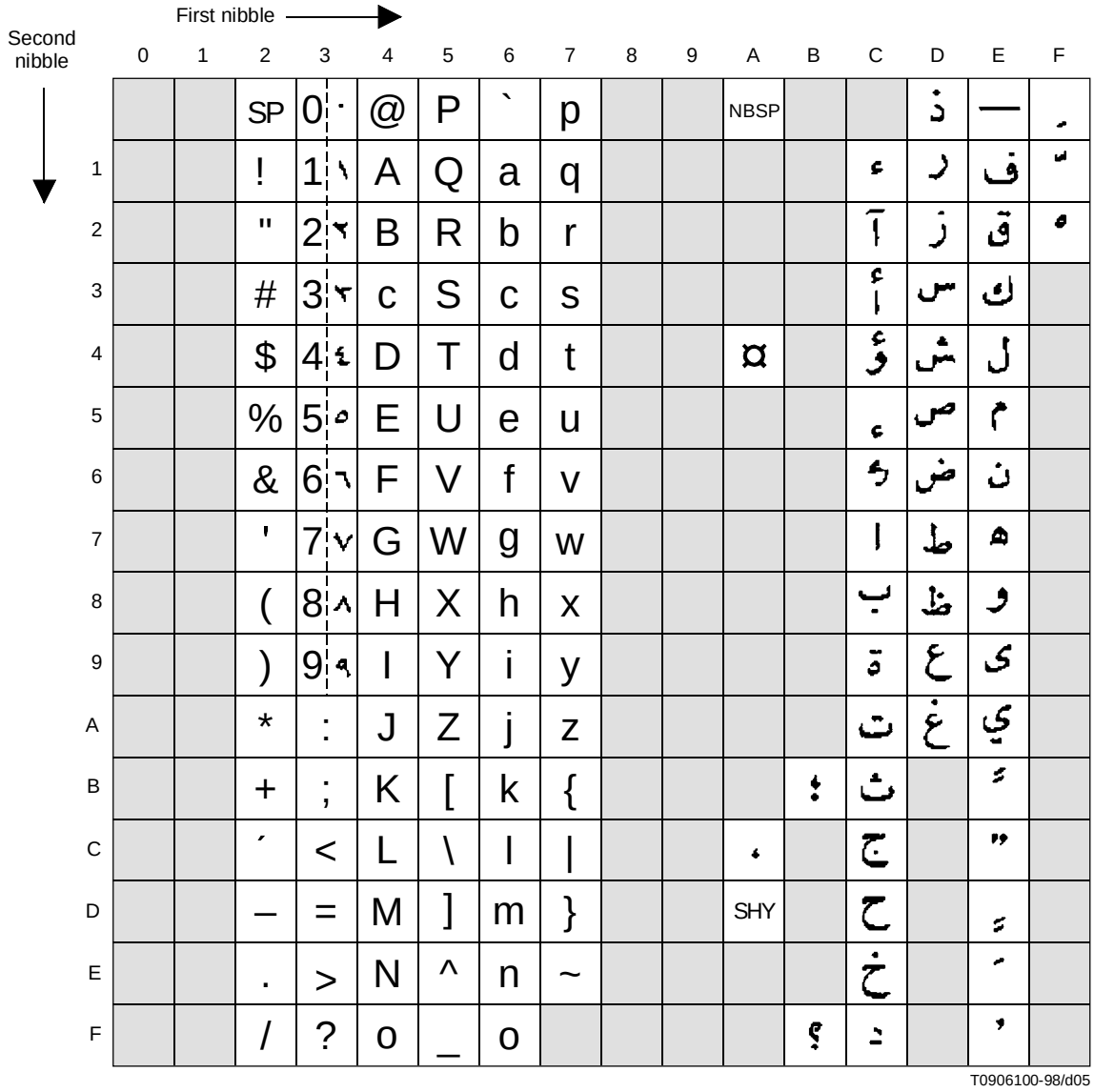
Figure A.A.1/J.94 – Character code table 00 – Latin alphabet



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

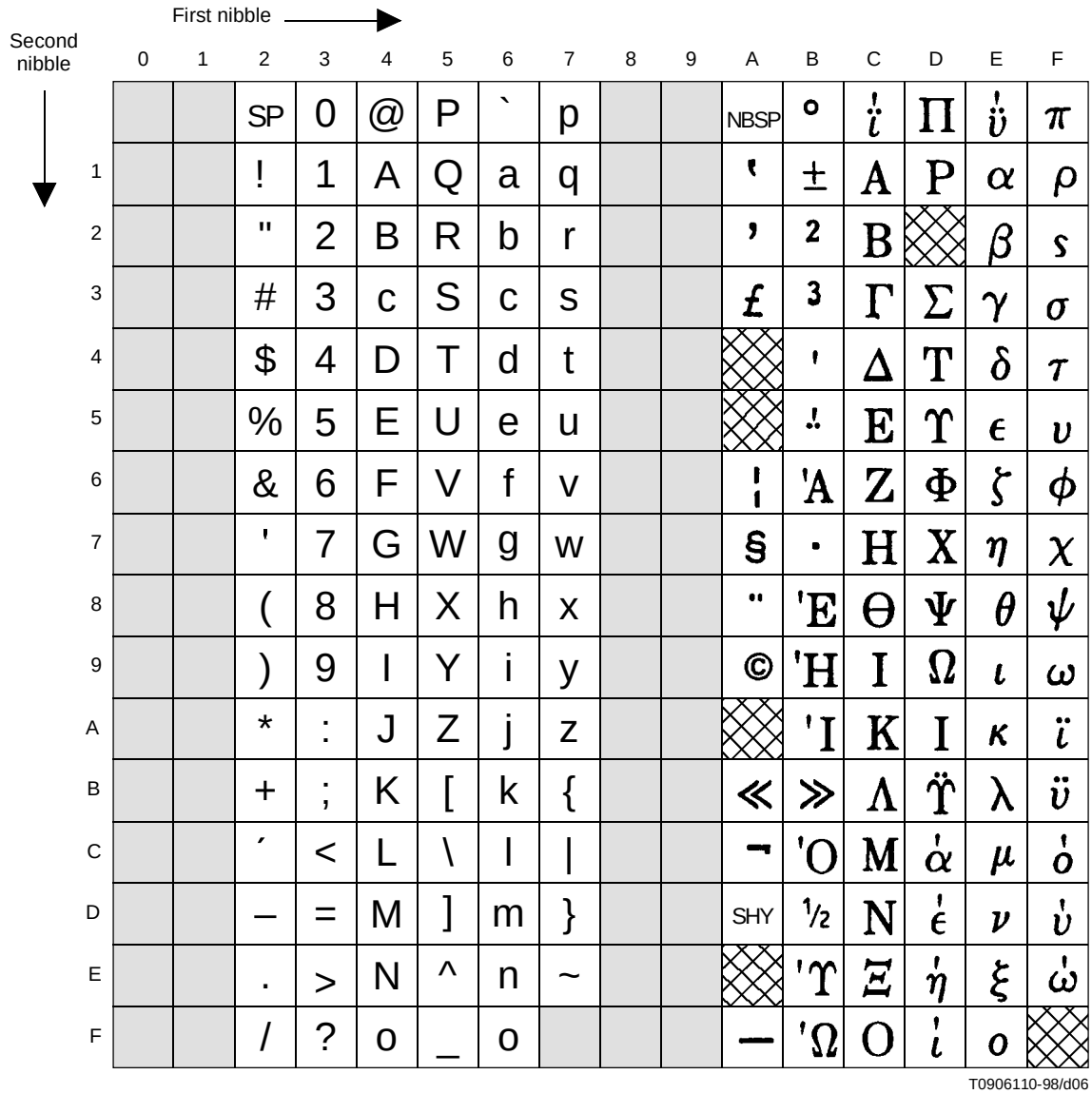
NOTE 2 – Table reproduced from ISO/IEC 8859-5 (1988) [5].

Figure A.A.2/J.94 – Character code table 01 – Latin/Cyrillic alphabet



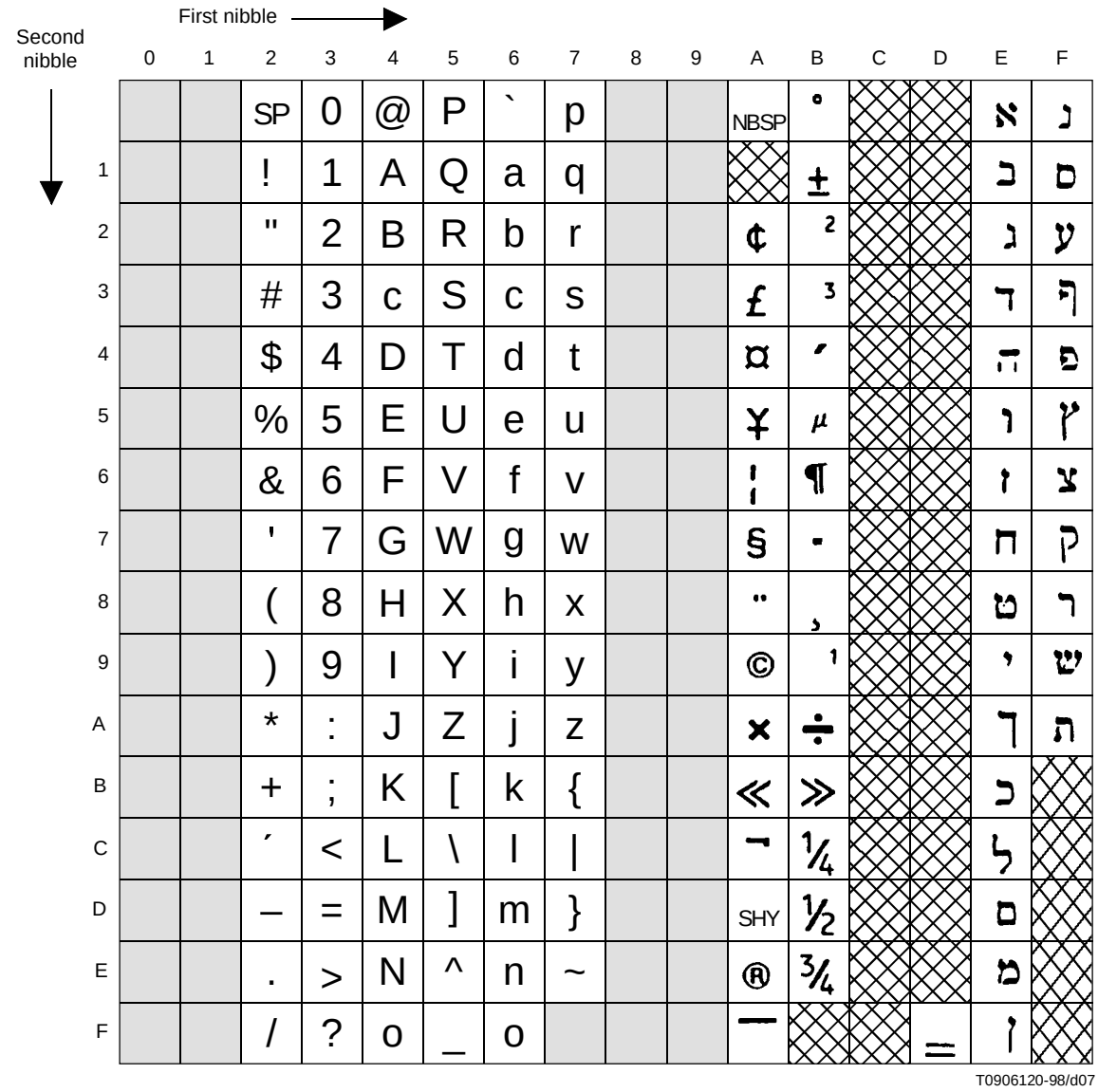
NOTE – Table reproduced from ISO 8859-6 (1987) [5].

Figure A.A.3/J.94 – Character code table 02 – Latin/Arabic alphabet



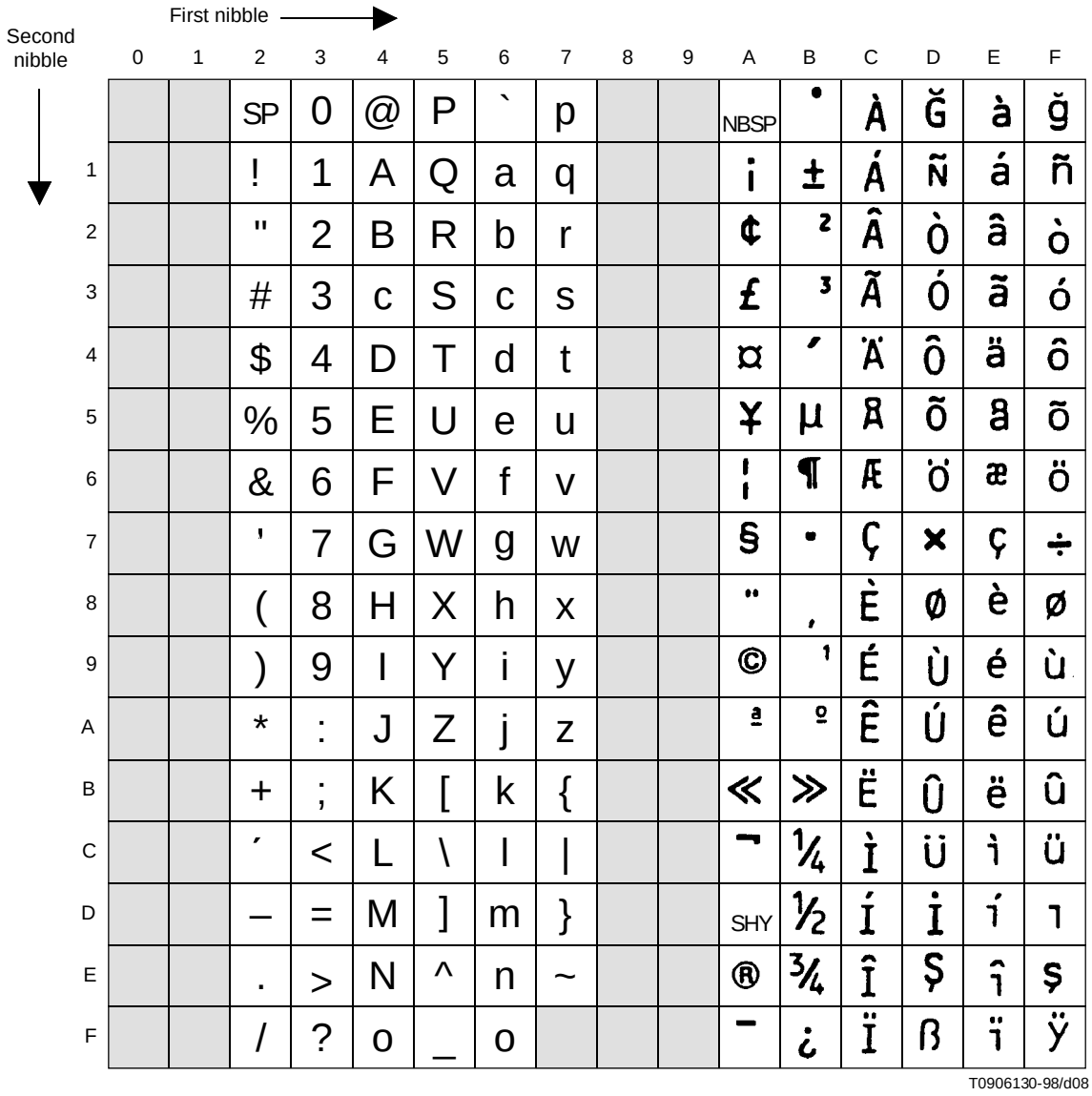
NOTE – Table reproduced from ISO 8859-7 (1987) [5].

Figure A.A.4/J.94 – Character code table 03 – Latin/Greek alphabet



NOTE – Table reproduced from ISO 8859-8 (1988) [5].

Figure A.A.5/J.94 – Character code table 04 – Latin/Hebrew alphabet



NOTE – Table reproduced from ISO/IEC 8859-9 [5].

Figure A.A.6/J.94 – Character code table – Latin alphabet No. 5

Annex A.B

CRC decoder model

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

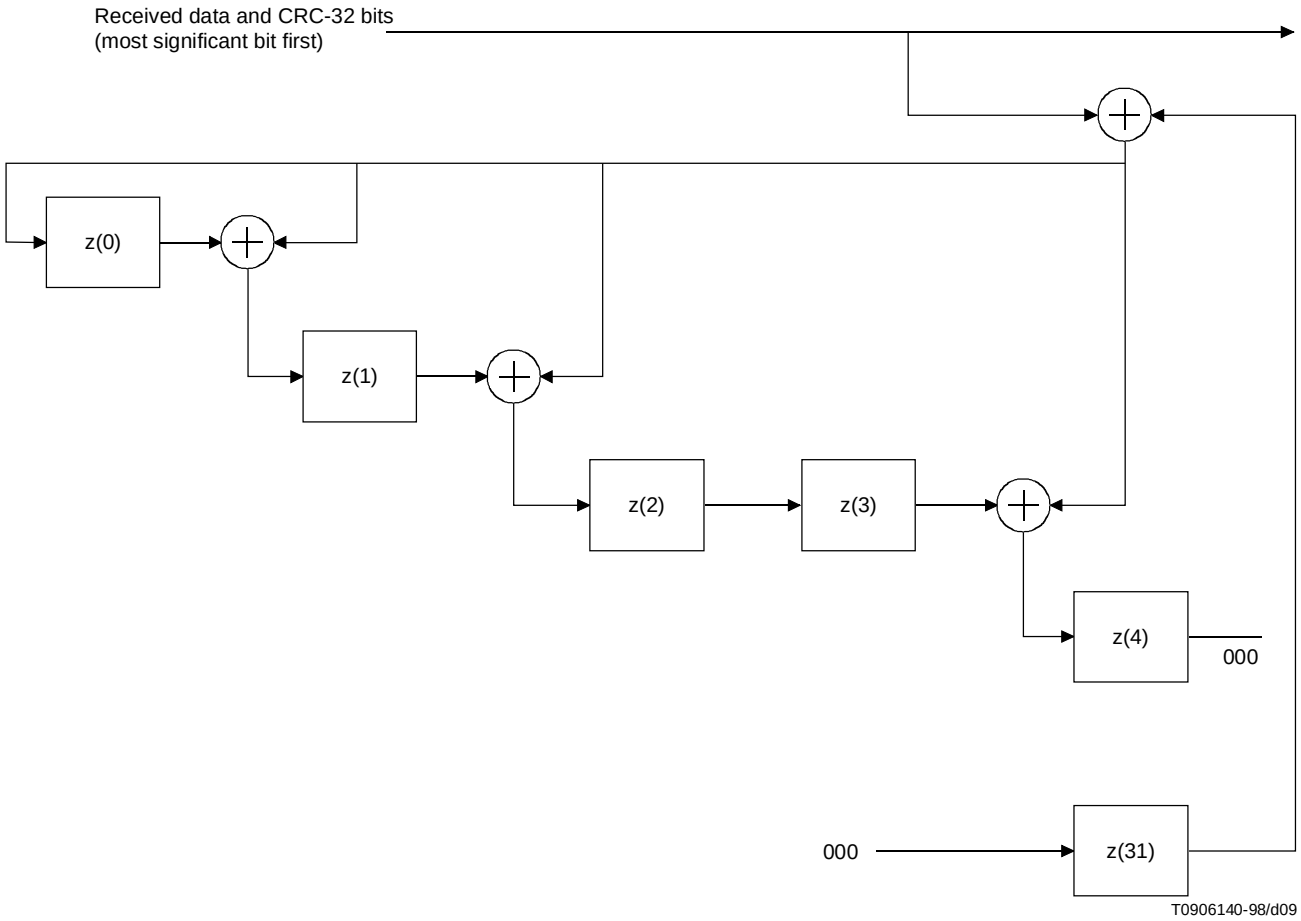


Figure A.B.1/J.94 – 32-bit CRC decoder model

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 A.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.

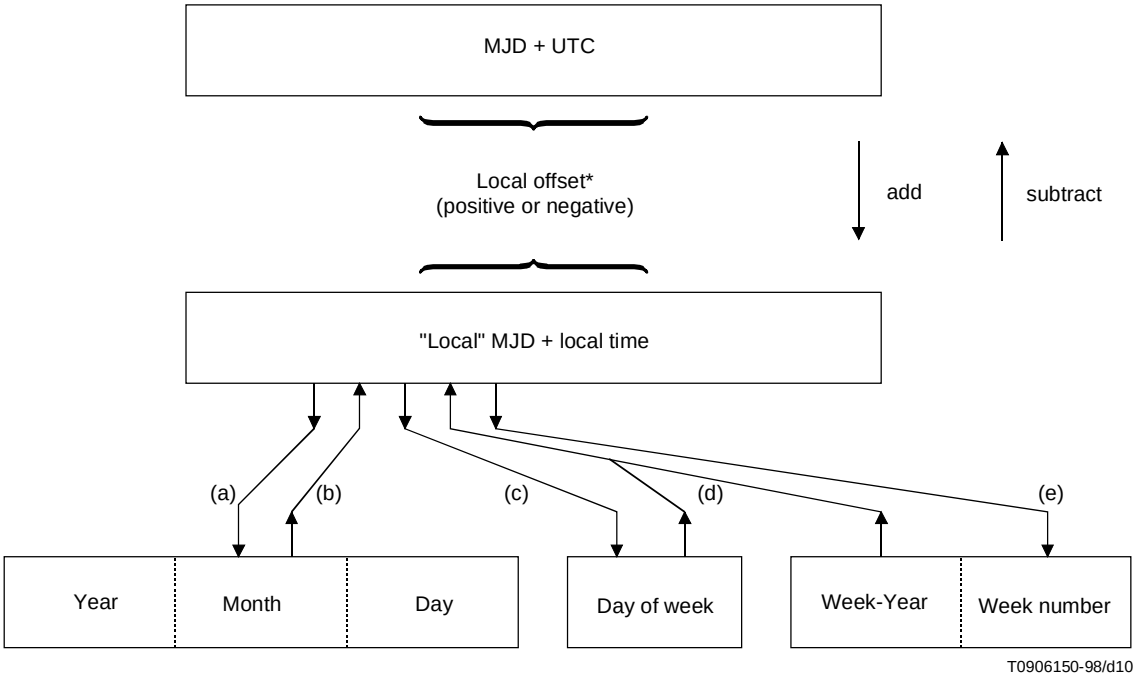
After shifting the last bit of the last CRC_32 byte into the decoder, i.e. into z(0) after the addition with the output of z(31), the output of all delay elements z(i) is read. In case of no errors, each of the outputs of z(i) has to be zero.

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

Appendix A.I

Conversion between time and date conventions

The types of conversion which may be required are summarized in Figure A.I.1.



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

Figure A.I.1/J.94 – Conversion routes between Modified Julian Date (MJD) and Universal Time Coordinated (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 formulae below:

Symbols used:

MJD	Modified Julian Date
UTC	Universal Time Coordinated
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
WN	Week number according to ISO 2015:1976
WD	Day of week from Monday (= 1) to Sunday (= 7)
K, L, M′, W, Y′	Intermediate variables
×	Multiplication
int	Integer part, ignoring remainder
mod 7	Remainder (0-6) after dividing integer by 7

a) To find Y, M, D from MJD

$Y' = \text{int} [(MJD - 15078.2) / 365.25]$

$M' = \text{int} \{ [MJD - 14956.1 - \text{int} (Y' \times 365.25)] / 30.6001 \}$

$D = MJD - 14956 - \text{int} (Y' \times 365.25) - \text{int} (M' \times 30.6001)$

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 $L = 0$

$MJD = 14956 + D + \text{int} [(Y - L) \times 365.25] + \text{int} [(M + 1 + L \times 12) \times 30.6001]$

c) To find WD from MJD

$WD = [(MJD + 2) \bmod 7] + 1$

d) To find MJD from WY, WN, WD

$MJD = 15012 + WD + 7 \times \{ WN + \text{int} [(WY \times 1461 / 28) + 0.41] \}$

e) To find WY, WN from MJD

$W = \text{int} [(MJD / 7) - 2144.64]$

$WY = \text{int} [(W \times 28 / 1461) - 0.0079]$

$WN = W - \text{int} [(WY \times 1461 / 28) + 0.41]$

Example – MJD = 45218 W = 4315

Y = (19)82 WY = (19)82

M = 9 (September) WN = 36

D = 6 WD = 1 (Monday)

NOTE – These formulae are applicable between the inclusive dates 1 March 1900 to 28 February 2100.

Appendix A.II

Bibliography

– *Implementation guidelines for use of telecommunications interfaces in the Digital Broadcasting systems*, DVB Project Office.

Annex B

Service Information for digital multi-programme System B

(For further study)

Annex C

Service Information for digital multi-programme System C

Summary

This annex describes the service information for digital broadcasting by cable television of Annex C/J.83 and basically constitutes a subset of Annex A to Recommendations J.94.

However, there are some specifications which are different from those of Annex A and also there are some specifications which are yet to be established.

C.1 SI tables

The specifications for SI tables fully comply with Annex A both in table names and in their function. See Table C.1.

Table C.1/J.94 – SI tables and their function

Table	Function
Program Association Table (PAT)	For each service in the multiplex, the PAT indicates the location (the PID values of the Transport Stream packets) of the corresponding Program Map Table (PMT). It also gives the location of the Network Information Table (NIT).
Conditional Access Table (CAT)	The CAT provides information on the Conditional Access (CA) systems used in the multiplex; the information is private (not defined within ITU-T Rec. H.222.0 | ISO/IEC 13818-1) and dependent on the CA system, but includes the location of the EMM stream, when applicable.
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.
Network information Table (NIT)	The location of the NIT is defined in ITU-T Rec. H.222.0 | ISO/IEC 13818-1, but the data format is outside the scope of ITU-T Rec. H.222.0 | ISO/IEC 13818-1. It is intended to provide information about the physical network. The syntax and semantics of the NIT are defined in this Recommendation.
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.
Service Description Table (SDT)	The SDT contains data describing the services in the system, e.g. names of services, the service provider, etc.
Event Information Table (EIT)	The EIT contains data concerning events or programs 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.
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.
Time and Date Table (TDT)	The TDT gives information relating to present time and date. This information is given in a separate table due to the frequent updating of the time information.
Stuffing Table (ST)	The ST is used to invalidate existing sections, for example at delivery system boundaries.

The PID allocation for SI and the allocation of table_id values are as shown in Tables C.2 and C.3, which are the same as those in Tables A.1 and A.2.

Table C.2/J.94 – PID allocation for SI

Table	PID value
PAT	0x0000
CAT	0x0001
NIT, ST	0x0010
SDT, BAT, ST	0x0011
EIT, ST	0x0012
RST, ST	0x0013
TDT	0x0014
NULL	0x1FFF

Table C.3/J.94 – Allocation of table_id values

Value	Table and description
0x00	PAT
0x01	CAT
0x02	PMT
0x40	NIT, network_information_section-actual_network
0x41	NIT, network_information_section-other_network
0x42	SDT, service_description_section-actual_transport_stream
0x46	SDT, service_description_section-other_transport_stream
0x4A	BAT
0x4E	EIT, event_information_section-actual_transport_stream, present/following
0x4F	EIT, event_information_section-other_transport_stream, present/following
0x50 to 0x5F	EIT, event_information_section-actual_transport_stream, before 8th day EIT, event_information_section-actual_transport_stream, on or after 8th day
0x60 to 0x6F	EIT, event_information_section-other_transport_stream, before 8th day EIT, event_information_section-other_transport_stream, on or after 8th day
0x70	TDT, time_date_section
0x71	RST, running_status_section
0x72	ST, stuffing_section
0x82 to 0x85	Reserved for conditional access system
0x90 to 0xBF	Selectable as operator setting table_id

C.2 Descriptor

C.2.1 Location and tag value

The location and tag value of each descriptor are as shown in Table C.4. The description, data structure, and syntax of each descriptor are the same as those in Table A.12. However, the coding of the data field of each descriptor is not specified.

Table C.4/J.94 – Possible locations of descriptors

Descriptor	Tag value	NIT	BAT	SDT	EIT	PMT	CAT
CA_descriptor	0x09					*	*
network_name_descriptor	0x40	*					
stuffing_descriptor	0x42	*	*	*	*		
cable_delivery_system_descriptor	0x44	*					
bouquet_name_descriptor	0x47		*	*			
service_descriptor	0x48			*			
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					*	
content_descriptor	0x54				*		
parental_rating_descriptor	0x55				*		
User-defined	0x80 to 0xBF						
Forbidden	0xFF						
area_specified_service_descriptor	0x96		*	*			
data_coding_method_descriptor	0xFD					*	
* Possible location.							

Descriptors which are used in Japan but not specified in Annex A are detailed in the following subclauses.

C.2.2 CA descriptor

The CA descriptor which is described in CAT and PMT identifies the type of conditional access and also identifies the PID in TS packet that carries the information related to conditional access. Conditional access is only available when this descriptor is used. See Table C.5.

Table C.5/J.94 – CA descriptor

Syntax	Bits	Identifier	Note
CA_descriptor(){			
descriptor_tag	8	uimsbf	
descriptor_length	8	uimsbf	
CA_system_id	16	uimsbf	
reserved	3	bslbf	"111"
CA_PID	13	uimsbf	
for (i = 0; i < N; i++) {			
private_data }	8xN	bslbf	
}			

C.2.3 Area specified service descriptor

This descriptor is used to render the services to the specified part within a given service area by transmitting either the area list of the service reception area or the one beyond the service reception area (see Table C.6). Area specified service is only available when this descriptor is used.

Table C.6/J.94 – Area specified service descriptor

Syntax	Bits	Identifier	Note
area_specified_service _descriptor(){			
descriptor_tag	8	uimsbf	
descriptor_length	8	uimsbf	
descriptor_flag	1	bslbf	(1: available, 0: not available)
reserved	7	bslbf	
for (i = 0; i < N; i++) { area_code	24	bslbf	alphanumeric 3 characters
}			
}			

C.2.4 Data coding method descriptor

The data coding method descriptor which is described in PMT identifies the data coding method for data broadcasting services. See Table C.7.

Table C.7/J.94 – Data coding method descriptor

Syntax	Bits	Identifier	Note
data_coding_method_descriptor(){			
descriptor_tag	8	uimsbf	
descriptor_length	8	uimsbf	
data_component_id	16	uimsbf	
for (i = 0; i < N; i++) {			
additional_identification_information	8xN	bslbf	
}			
}			

C.3 Character code tables

The tables corresponding to Annex A.A are under study now.

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	**Transmission of television, sound programme and other multimedia signals**
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure
Series Z	Languages and general software aspects for telecommunication systems