



***Society of Cable
Telecommunications
Engineers***

ENGINEERING COMMITTEE
Digital Video Subcommittee

AMERICAN NATIONAL STANDARD

ANSI/SCTE 65 2016

**SERVICE INFORMATION
DELIVERED OUT-OF-BAND FOR
DIGITAL CABLE TELEVISION**

NOTICE

The Society of Cable Telecommunications Engineers (SCTE) Standards are intended to serve the public interest by providing specifications, test methods and procedures that promote uniformity of product, interchangeability and ultimately the long term reliability of broadband communications facilities. These documents shall not in any way preclude any member or nonmember of SCTE from manufacturing or selling products not conforming to such documents, nor shall the existence of such standards preclude their voluntary use by those other than SCTE members, whether used domestically or internationally. SCTE assumes no obligations or liability whatsoever to any party who may adopt the Standards. Such adopting party assumes all risks associated with adoption of these Standards or Recommended Practices, and accepts full responsibility for any damage and/or claims arising from the adoption of such Standards or Recommended Practices. Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. SCTE shall not be responsible for identifying patents for which a license may be required or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention. Patent holders who believe that they hold patents which are essential to the implementation of this standard have been requested to provide information about those patents and any related licensing terms and conditions. Any such declarations made before or after publication of this document are available on the SCTE web site at <http://www.scte.org>.

All Rights Reserved

© Society of Cable Telecommunications Engineers, Inc. 2016

140 Philips Road

Exton, PA 19341

SERVICE INFORMATION DELIVERED OUT-OF-BAND FOR DIGITAL CABLE TELEVISION

TABLE OF CONTENTS

1 PURPOSE, SCOPE AND ORGANIZATION	1
1.1 Purpose	1
1.2 Scope	1
1.3 Organization	3
2 REFERENCES.....	3
3 DEFINITIONS	5
3.1 Compliance Notation	5
3.2 Definition of Terms	5
3.3 Acronyms and Abbreviations	6
3.4 Section and Data Structure Syntax Notation	7
3.4.1 Field Sizes	7
4 TABLE STRUCTURE	9
4.1 Table ID Ranges and Values	9
4.2 Extensibility	10
4.3 Reserved Fields	11
4.4 Private Table Section Syntax	11
4.4.1 Protocol Version	11
4.4.2 Format Identifier	11
4.4.3 Private Message Body	11
4.4.4 CRC	11
5 TABLE SECTION FORMATS	12
5.1 Network Information Table	12
5.1.1 Carrier Definition Subtable (CDS)	13
5.1.2 Modulation Mode Subtable (MMS)	15
5.1.3 Descriptors Count	17
5.2 Network Text Table	17
5.3 Short-form Virtual Channel Table Section	20
5.3.1 Defined Channels Map	21
5.3.2 Virtual Channel Map	22
5.3.3 Inverse Channel Map	26
5.4 System Time Table Section	27

5.5 Master Guide Table (MGT)	28
5.5.1 Restrictions on PID Values	31
5.5.2 Restrictions on Order of Occurrence of Table References	31
5.6 Long-form Virtual Channel Table	32
5.7 Rating Region Table (RRT)	38
5.8 Aggregate Event Information Tables (AEIT)	41
5.9 Aggregate Extended Text Tables (AETT)	44
6 DESCRIPTORS.....	47
6.1 Descriptor Usage	47
6.2 Stuffing Descriptor	47
6.3 AC-3 Audio Descriptor	48
6.4 Caption Service Descriptor	48
6.5 Content Advisory Descriptor	49
6.6 Revision Detection Descriptor	51
6.7 Two Part Channel Number Descriptor	51
6.8 Channel Properties Descriptor	52
6.9 Extended Channel Name Descriptor	54
6.10 Time Shifted Service Descriptor	54
6.11 Component Name Descriptor	55
6.12 Daylight Savings Time Descriptor	56
6.13 User Private Descriptors	56
7 TEXT STRING CODING.....	58
7.1 Multilingual Text String (MTS) Format	58
7.1.1 Mode Byte Definition	61
7.1.2 Format Effectors	61
7.1.3 Default Attributes	62
7.1.4 Mode Zero	62
7.1.5 Supported Characters	63
7.2 Multiple String Structure (MSS)	63
ANNEX A OPERATIONAL PROFILES FOR CABLE SERVICE INFORMATION DELIVERY	66
A.1 Operational Profiles	66
A.1.1 Profile 1 – Baseline	66
A.1.2 Profile 2 – Revision Detection	66
A.1.3 Profile 3 – Parental Advisory	66
A.1.4 Profile 4 – Standard Electronic Program Guide Data	66
A.1.5 Profile 5 – Combination	67
A.1.6 Profile 6 – PSIP Only	67
A.2 Profile Definition Tables	67

A.3 Operational Considerations for the use of profiles (Informative)	69
ANNEX B IMPLEMENTATION RECOMMENDATIONS	70
B.1 Implications for Retail Digital Cable-Ready Devices	70
B.2 Channel Number Handling	70
B.3 Processing of Dynamic Changes to Service Information	70
B.4 AEITs May Include Event Information for Inaccessible Channels	70
B.5 Splice Flag Processing	71
ANNEX C SERVICE INFORMATION OVERVIEW AND GUIDE	72
C.1 Table Hierarchy	72
C.2 SI_base PID	76
C.3 Representation of Time	85
ANNEX D PACKET RATES	89
D.1 Maximum cycle times	89
D.2 Maximum Transmission Rates	89
D.3 MINIMUM Transmission Rates	89
ANNEX E DAYLIGHT SAVINGS TIME CONTROL.....	90
ANNEX F STANDARD HUFFMAN TABLES FOR TEXT COMPRESSION	92
F.1 Character Set Definition	92
F.2 Standard Compression Type 1 Encode/Decode Tables	95
F.3 Standard Compression Type 2 Huffman Encode/Decode Tables	102

List of Figures

Figure 1.1 A Framework for the Extended Channel Service Information Stream	2
Figure C.1 Hierarchy of Table Sections -- Profiles 1 and 2	72
Figure C.2 Hierarchy of Table Sections -- Profile 3	73
Figure C.3 Hierarchy of Table Sections -- Profile 4	74
Figure C.4 Hierarchy of Table Sections -- Profile 5	75
Figure C.5 Hierarchy of Table Sections -- Profile 6	76
Figure C.6 An instance of a Rating Region Table	81
Figure C.7 Example AEIT-0	83
Figure C.8 AEIT data structure	84
Figure C.9 Structure of AETT	84

List of Tables

Table 3.1 Field Sizes Example	8
Table 4.1 Table ID Ranges and Values for Out-of-Band Transport	9
Table 4.2 Network private table section format	10
Table 5.1 Network Information Table section format	13
Table 5.2 Network Information Table Subtype	13
Table 5.3 CDS record format	14
Table 5.4 Spacing Unit	15
Table 5.5 Frequency Unit	15
Table 5.6 MMS record format	15
Table 5.7 Transmission System	16
Table 5.8 Inner Coding Mode	16
Table 5.9 Modulation Format	17
Table 5.10 Network Text Table section format	18
Table 5.11 Network Text Table Subtype	18
Table 5.12 Source Name Subtable format	19
Table 5.13 Short-form Virtual Channel Table section format	20
Table 5.14 S-VCT Table Subtypes	21
Table 5.15 DCM structure format	21
Table 5.16 VCM structure format	22
Table 5.17 Virtual channel record format	23
Table 5.18 Path Select	24
Table 5.19 Transport Type	24
Table 5.20 Channel Type	24
Table 5.21 Video Standard	26
Table 5.22 ICM structure format	26
Table 5.23 System Time Table section format	27
Table 5.24 Master Guide Table section format	29
Table 5.25 MGT Table Types	30
Table 5.26 Long-form Virtual Channel Table section format	33
Table 5.27 Major and Minor Channel Number Field Coding	35
Table 5.28 Modulation Modes	36
Table 5.29 Path Select	37
Table 5.30 Service Types	38
Table 5.31 Rating Region Table section format	39
Table 5.32 Rating Regions	40
Table 5.33 Aggregate Event Information Table format	43
Table 5.34 ETM_present	44
Table 5.35 Aggregate Extended Text Table format	45
Table 5.36 ETM ID	46
Table 6.1 Descriptor Usage	47
Table 6.2 Caption Service Descriptor format	48
Table 6.3 Content Advisory Descriptor format	50
Table 6.4 Revision Detection Descriptor format	51
Table 6.5 Two-part Channel Number Descriptor format	52
Table 6.6 Channel Properties Descriptor format	53

Table 6.7 Extended Channel Name Descriptor format	54
Table 6.8 Time Shifted Service Descriptor format	55
Table 6.9 Component Name Descriptor format	55
Table 6.10 Daylight Savings Time Descriptor format	56
Table 7.1 Text String Coding Format in Tables	58
Table 7.2 Text String Coding Format in Descriptors	58
Table 7.3 Mode Byte Encoding	60
Table 7.4 Multilingual text string format	60
Table 7.5 Format Effector Function Codes	62
Table 7.6 Encodings of Columns 8 and 9 of Mode Zero Latin Character Set	62
Table 7.7 Multiple String Structure	64
Table 7.8 Compression Types	64
Table 7.9 Modes	65
Table A.1 Usage of Table Sections in Various Profiles	68
Table A.2 Usage of Descriptors in Various Profiles	69
Table C.1 Example Master Guide Table content	80
Table C.2 Example Revised Master Guide Table content	80
Table C.3 Receiver Behavior with hidden and hide_guide attributes	85
Table D.1 Maximum cycle time for the STT, MGT, S-VCT, L-VCT and RRT	89
Table D.2 Maximum rate for each packet stream	89
Table D.3 Minimum rate for each packet stream	89
Table E.1 Basic Use of Daylight Savings Fields Through the Year	91
Table F.1 Characters with Special Definitions	92
Table F.2 Decode Table Format	93
Table F.3 Decode Tree Format	94
Table F.4 English-language Program Title Encode Table	95
Table F.5 English-language Program Title Decode Table	99
Table F.6 English-language Program Description Encode Table	102
Table F.7 English-language Program Description Decode Table	107

SERVICE INFORMATION DELIVERED OUT-OF-BAND FOR DIGITAL CABLE TELEVISION

1 PURPOSE, SCOPE AND ORGANIZATION

1.1 Purpose

This document defines a standard for Service Information (SI) delivered out-of-band on cable. This standard is designed to support “navigation devices” on cable. The current specification defines the syntax and semantics for a standard set of tables providing the data necessary for such a device to discover and access digital and analog services offered on cable.

1.2 Scope

This specification defines SI tables delivered via an out-of-band path to support service selection and navigation by digital cable set-top boxes and other “digital cable-ready” devices. The SI tables defined in this standard are formatted in accordance with the Program Specific Information (PSI) data structures defined in MPEG-2 Systems [1].

The formal definition of “digital cable-ready” has a scope broader than that of the current standard. The formal definition includes requirements related to navigation and service selection, demodulation and decoding, video format decoding, Emergency Alert handling, and other aspects. The current specification supports, primarily, the navigation and service selection function for services delivered in the clear, as well as those subject to conditional access.

This specification does not address the Electronic Program Guide application itself or any user interface which might deal with the presentation and application of the Service Information.

1.3 Digital Cable Ready Device

A digital cable-ready device can take the form of a cable set-top box, a computer, a television, or a convergence of these. Devices such as digital video recorders may also be cable-ready. A digital cable-ready device capable of processing access controlled digital services supports an interface to a conditional access module. As used here, the term “Host” refers to the capability to support an interface to a standard Point-of-Deployment (POD)¹ security module.

SI data delivered out-of-band is transported in accordance with the Extended Channel Interface defined in SCTE 28. To obtain access to a POD Extended Channel Interface, the digital cable-ready device must act as a Host to a POD security module. The Extended Channel

¹ The Point-of-Deployment module is also known as a CableCARD™ device.

interface is designed to present the needed SI data to the Host. This data can be used by the Host for channel navigation, construction of electronic program guides and other associated functions.

Figure 1.1 is a high-level block diagram illustrating the POD module to Host interface via the Extended Channel interface. The Host is responsible for providing a standard receiver/QPSK demodulator function for the POD module. The choice of transport format of bits coming across from the receiver/QPSK demodulator to the POD module is by mutual agreement between the POD and the cable head-end equipment. The transport format of data traveling between the Host and POD module on the Extended Channel interface conforms to standards defined in [6].

The POD module may perform various transport, filtering, and error checking/correction functions on the out-of-band data stream as depicted by the box labeled “Transport Processing, Filtering, and Routing.” As described in [6], the Host may request from the POD module to open one or several “flows” in which to receive PSI sections taken from the cable out-of-band data stream. Each flow is associated with a PID value, in accordance with MPEG-2 Transport Stream concepts.

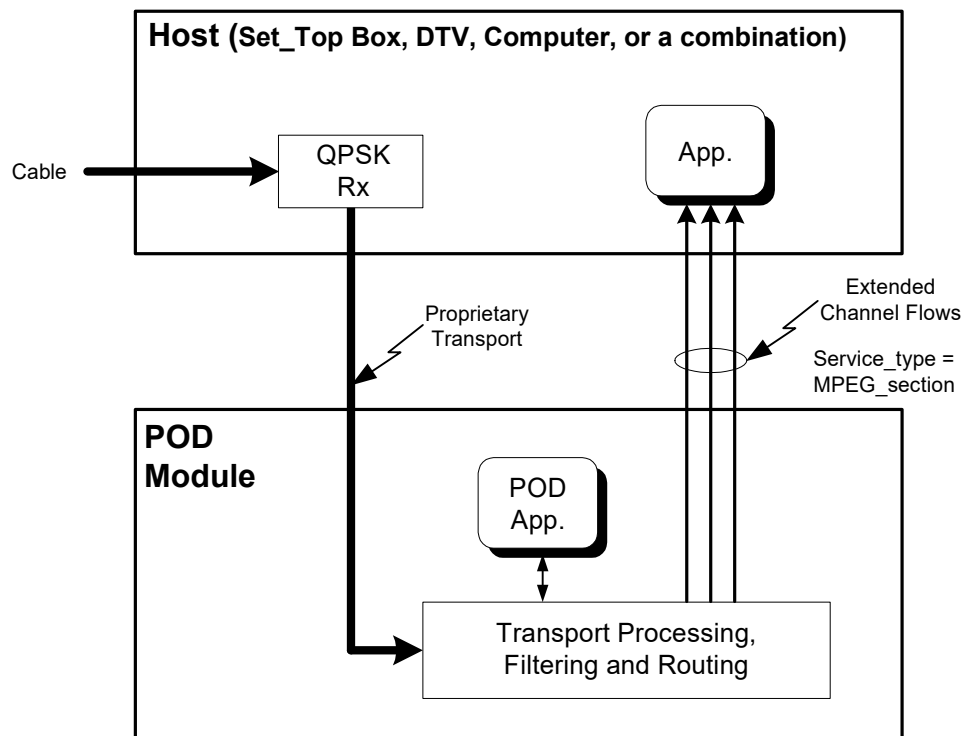


Figure 1.1 A Framework for the Extended Channel Service Information Stream

Data flowing to the Host from the POD module that is associated with Service_type=MPEG_section is required to be in the form of MPEG PSI data structures. However, data delivered into the POD from cable out-of-band may or may not be organized in a Transport Stream compliant with ISO/IEC 13818-1. In other words, PID values associated with MPEG-2 tables on the Extended Channel interface *may or may not* correspond to MPEG-2 Transport Stream packet header PID values from the cable out-of-band.

Independent of the fact that out-of-band data may reach the POD module via a proprietary method, the data structures delivered across the Extended Channel shall be formatted as MPEG-2 table sections. Like table sections carried in an MPEG-2 Transport Stream, each is associated with a PID value.

1.4 Organization

The sections of this document are organized as follows:

- **Section 1** — Provides this general introduction.
- **Section 2** — Lists applicable documents.
- **Section 3** — Provides a list of acronyms and abbreviations used in this document.
- **Section 4** — Describes the basic structure of sections.
- **Section 5** — Describes formats of sections carried in the Base PID.²
- **Section 6** — Explains descriptors applicable to the tables defined in this standard.
- **Section 7** — Describes multilingual character string coding.
- **Annex A** — Defines profiles of choice for cable operator compliance with this standard.
- **Annex B** — Discusses recommendations for receiver implementations.
- **Annex C** — Provides an overview of tables defined in this Service Information standard.
- **Annex D** — Specifies packet rates for delivery of SI data
- **Annex E** — Defines the daylight savings time control fields in the System Time Table.
- **Annex F** — Defines the standard Huffman tables used for text compression.

2 REFERENCES

The following documents are applicable to this Service Information standard:

1. ITU-T Rec. H.222.0 | ISO/IEC 13818-1 (2013), Information Technology — Generic coding of moving pictures and associated audio information — Part 1: systems.
2. ATSC A/52:2012: Digital Audio Compression (AC-3) (E-AC-3) Standard.
3. ATSC A/53: ATSC Digital Television Standard, Parts 1-6, (2009-2013).
4. ANSI/SCTE 07 2013, Digital Transmission Standard for Cable Television

² The Base PID is the PID associated with the “base” Service Information tables. In this protocol, the base_PID is fixed at 0x1FFC. Refer to Table 4.1.

5. A/65:2013: Program And System Information Protocol For Terrestrial Broadcast And Cable.
6. ANSI/SCTE 28 2012, Host-POD Interface Standard.
7. ANSI/SCTE 18 2013, Emergency Alert Messaging for Cable.
8. ISO 639, Code for the Representation of Names of Languages, 1988.
9. ISO/IEC 10646-1:1993, Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane.
10. ISO/IEC 10646:2012 Information technology - Universal Coded Character Set (UCS).
11. ISO/IEC 8859, Information Processing — 8-bit Single-Octet Coded Character Sets, Parts 1 through 10.
12. ITU-T Rec. J.83 (12/07) Digital multi-programme systems for television, sound and data services for cable distribution.
13. CEA-708-E (ANSI) Digital Television (DTV) Closed Captioning, Consumer Electronics Association, 2013.
14. ANSI-CEA-766-D, U.S. and Canadian Rating Region Tables (RRT) and Content Advisory Descriptors for Transport of Content Advisory Information Using ATSC Program and System Information Protocol (PSIP), Consumer Electronics Association, 2013.
15. CEA-608-E, Line 21 Data Services, Consumer Electronics Association, 2008.

3 DEFINITIONS

3.1 Compliance Notation

As used in this document, “*shall*” denotes a mandatory provision of the standard. “*Should*” denotes a provision that is recommended but not mandatory. “*May*” denotes a feature whose presence does not preclude compliance, that may or may not be present as optional for the implementers.

3.2 Definition of Terms

The following terms are used throughout this document:

conditional access: The control and security of subscriber access to cable or broadcast services and events in the form of video, data and voice communications.

host: A device capable of supporting a POD module by implementing the interface protocol defined in SCTE 28 [6]. SCTE 28 defines the Extended Channel data path through which the SI tables defined in this standard are passed.

navigation: The process of selection and movement among analog and digital services offered on the cable network. The service information tables defined in this protocol assist in the navigation process by providing physical service locations, channel names and numbers for user reference. Those tables supporting electronic program guides also assist the navigation process.

program element: A generic term for one of the elementary streams or other data streams that may be included in a program.

program: A collection of program elements. Program elements may be elementary streams. Program elements need not have any defined time base. Those that do have a common time base are intended for synchronized presentation. The term *program* is also used in the context of a “television program” such as a scheduled daily news broadcast. The distinction between the two usages should be understood by context.

region: as used in this document, a region is a geographical area consisting of one or more countries.

section or table section: A data structure comprising a portion of an *ISO/IEC 13818-1*-defined table, such as the Program Association Table (PAT), Conditional Access Table (CAT), or Program Map Table (PMT). The term conforms to MPEG terminology. All sections begin with the `table_ID` and end with the `CRC_32` field. Sections are carried in Transport Stream packets in which the starting point within a packet payload is indicated by the `pointer_field` mechanism defined in the *ISO/IEC 13818-1 Systems* document. The Network Information Table, for example, defines portions of several types of tables.

service: *ISO/IEC 13818-1* uses the term *program* to refer to a collection of program elements with no regard to time. In this Service Information standard, the term *service* is used in this same context to denote a collection of elementary components. Usage of the term *service* clarifies certain discussions that also involve the notion of the term *program* in its traditional meaning — for example, in the statement, “A video service carries a series of programs.” In a broader sense,

service is also intended for multimedia services of video, voice and data, as these services become prevalent.

stream: An ordered series of bytes. The usual context for the term *stream* involves specification of a particular PID (such as the “Program Map PID stream”), in which case the term indicates a series of bytes extracted from the packet multiplex from packets with the indicated PID value.

3.3 *Acronyms and Abbreviations*

The following acronyms and abbreviations are used within this specification:

AEIT	Aggregate Event Information Table
AETT	Aggregate Extended Text Table
ATSC	Advanced Television Systems Committee
BMP	Basic Multilingual Plane
bslbf	bit serial, leftmost bit first
CAT	Conditional Access Table
CC	Closed Caption
CDS	Carrier Definition Subtable
CRC	Cyclic Redundancy Check
DCM	Defined Channels Map
DTV	Digital Television
ECM	Entitlement Control Message
EMM	Entitlement Management Message
ETSI	European Telecommunications Standards Institute
GPS	Global Positioning System
ICM	Inverse Channel Map
ITU	International Telecommunications Union
L-VCT	Long-form Virtual Channel Table
LSB	Least Significant Bit
MGT	Master Guide Table
MMS	Modulation Mode Subtable
MPEG	Moving Picture Experts Group
MPAA	Motion Picture Association of America
MSB	Most Significant Bit
MSS	Multiple String Structure
MTS	Multi-lingual Text String
NTSC	National Television System Committee
NVOD	Near Video On Demand
OOB	Out-of-band
PAT	Program Association Table
PCR	Program Clock Reference
PES	Packetized Elementary Stream
PID	Packet Identifier
PMT	Program Map Table
POD	Point of Deployment

PSIP	Program and System Information Protocol
PTC	Physical Transmission Channel
PTS	Presentation Time Stamp
rpchof	remainder polynomial coefficients, highest order first
RRT	Rating Region Table
S-VCT	Short-form Virtual Channel Table
SCTE	Society of Cable Telecommunications Engineers
SI	Service Information
SNS	Source Name Subtable
TS	Transport Stream
UTC	Coordinated Universal Time ³
uimsbf	unsigned integer, most significant bit first
VCM	Virtual Channel Map

3.4 Section and Data Structure Syntax Notation

This document contains symbolic references to syntactic elements. These references are typographically distinguished by the use of a different font (e.g., `restricted`), may contain the underscore character (e.g., `sequence_end_code`) and may consist of character strings that are not English words (e.g., `dynrng`).

The formats of sections and data structures in this document are described using a C-like notational method employed in *ISO/IEC 13818-1*. Extensions to this method are described in the following sections.

3.4.1 Field Sizes

Each data structure is described in a table format wherein the size in bits of each variable within that section is listed in a column labeled “Bits.” The column adjacent to the Bits column is labeled “Bytes” and indicates the size of the item in bytes. For convenience, several bits within a particular byte or multi-byte variable may be aggregated for the count. An example follows:

³ Since unanimous agreement could not be achieved by the ITU on using either the English word order, CUT, or the French word order, TUC, a compromise to use neither was reached.

Table 3.1 Field Sizes Example

	Bits	Bytes	Format
foo_section(){			
section_syntax_indicator	1	1	
...			
if (section_syntax_indicator) {			
table_extension	16	(2)	uimsbf
reserved	2	(1)	bslbf
version_number	5		uimsbf
current_next_indicator	1		bslbf {next, current}
...			
}			
...			

In the byte count column, items that are conditional (because they are within a loop or conditional statement) are in parentheses. Nested parentheses are used if the loops or conditions are nested.

4 TABLE STRUCTURE

This section describes details of the structure of MPEG-2 tables defined in this standard.

Tables and table sections defined in this Service Information standard are structured in the same manner used for carrying *ISO/IEC 13818-1* -defined PSI tables. The MPEG-defined 32-bit CRC is required.

4.1 Table ID Ranges and Values

Table 4.1 defines table_ID ranges and values for tables defined in MPEG and in this standard.

Table 4.1 Table ID Ranges and Values for Out-of-Band Transport

Table ID Value (hex)	Tables	PID	Ref.
0x00 0x01 0x02 0x03-0x3F	ISO/IEC 13818-1 Sections:		
	Program Association Table (PAT)	0	Ref. [1]
	Conditional Access Table (CAT)	1	Ref. [1]
	TS Program Map Table (PMT)	per PAT	Ref. [1]
0x40-0x7F 0x80-0xBF	User Private Sections:		
	[User Private for other systems] [SCTE User Private]		
0xC0-0xC1	Other Standards: [Used in other standards]		
0xC2 0xC3 0xC4 0xC5 0xC6 0xC7 0xC8 0xC9 0xCA 0xCB-0xD5 0xD6 0xD7 0xD8 0xD9-0xFE	Service Information Tables:		
	Network Information Table (NIT)	0x1FFC	Sec. 5.1
	Network Text Table (NTT)	0x1FFC	Sec. 5.2
	Short-form Virtual Channel Table (S-VCT)	0x1FFC	Sec. 5.3
	System Time Table (STT)	0x1FFC	Sec. 5.4
	[Used in other standards]	-	-
	Master Guide Table (MGT)	0x1FFC	Sec. 5.5
	Reserved	-	-
	Long-form Virtual Channel Table (L-VCT)	0x1FFC	Sec. 5.6
	Rating Region Table (RRT)	0x1FFC	Sec. 5.7
	[Used in ATSC]	-	-
	Aggregate Event Information Table (AEIT)	per MGT	Sec. 5.8
	Aggregate Extended Text Table (AETT)	per MGT	Sec. 5.9
	Cable Emergency Alert Message	0x1FFC	Ref. [7]
	[Reserved for future use or by other standards]	-	-

Table sections defined in this Service Information standard, and any created as user extensions to it are considered “private” with respect to *ISO/IEC 13818-1*. Table section types 0x80 through 0xBF are user-defined (outside the scope of this Service Information standard).

The maximum total length of any table section defined in this standard is 1024 bytes, except for the MGT, L-VCT, AEIT and AETT, each of which has a maximum total length of

4096 bytes. This total includes table_ID, CRC, and all fields contained within the specific table section.

4.2 Extensibility

This Service Information standard defines a number of tables and table sections. The Service Information standard is designed to be extensible via the following mechanisms:

1. **Reserved Fields:** Fields in this Service Information standard marked `reserved` are reserved for use either when revising this standard, or when another standard is issued that builds upon this one. See Section 4.4 below.
2. **Standard Table Types:** As indicated in Table 4.1, table_ID values in the range 0xCE through 0xFE are reserved for use either when revising this Service Information standard, or when another standard is issued that builds upon this one.⁴
3. **User Private Table Types:** As indicated in Table 4.1, table_id values in the range 0x80 through 0xBF are reserved for “user private” use. The format of user private tables carried in the Network PID shall conform to the syntax described in Table 4.2.
4. **User Private Descriptors:** Privately defined descriptors may be placed at designated locations throughout the table sections described in this Service Information standard. Ownership of one or more user private descriptors is indicated by the presence of an MPEG registration_descriptor() preceding the descriptor(s).

Table 4.2 Network private table section format

	Bits	Bytes	Format
network_private_table section(){			
private_table_ID	8	1	uimsbf (0x80 <= table_ID <= 0xBF)
section_syntax_indicator	1	2	Bslbf
zero	1		Bslbf
reserved	2		Bslbf
section_length	12		Uimsbf
if (section_syntax_indicator==1) {			
table_extension	16	(2)	Uimsbf
reserved	2	(1)	Bslbf
version_number	5		Uimsbf
current_next_indicator	1		bslbf {next, current}
section_number	8	(1)	Uimsbf
last_section_number	8	(1)	Uimsbf
}			
zero	3	1	Bslbf
protocol_version	5		see Section 4.4.1
format_identifier	32	4	Uimsbf
private_message_body()	N*8	N	
CRC_32	32	4	Rpchof
}			

⁴ Note: Assignment of table_ID values in the 0xCE to 0xFE range requires coordination between ATSC and SCTE.

4.3 *Reserved Fields*

reserved — Fields in this Service Information standard marked “reserved” shall not be assigned by the user, but shall be available for future use. Hosts are expected to disregard reserved fields for which no definition exists that is known to that unit. Fields marked “reserved” shall be set to “1” until such time as they are defined and supported.

zero — Indicates the bit or bit field shall be “0”.

4.4 *Private Table Section Syntax*

Table 4.2 defines the syntax for user private table sections. The MPEG-defined CRC is required. Refer to *ISO/IEC 13818-1* for definition of MPEG-standard fields.

private_table_ID — The value of table_ID in private table sections shall be in the range 0x80 through 0xBF.

4.4.1 *Protocol Version*

protocol_version — A 5-bit unsigned integer field whose function is to allow, in the future, any defined table type to carry parameters that may be structured fundamentally differently from those defined in the current protocol. At present, all defined table section types in this protocol are defined for protocol_version zero only. Nonzero values of protocol_version may only be processed by Receivers designed to accommodate the later versions as they become standardized.

4.4.2 *Format Identifier*

format_identifier — A 32-bit unsigned integer value which unambiguously identifies the entity defining this network_private_table_section() syntax. Values for format_identifiers shall be obtained from SCTE.

4.4.3 *Private Message Body*

private_message_body() — A data structure defined by the private entity identified by format_identifier.

4.4.4 *CRC*

CRC_32 — The 32-bit CRC value defined in [1] for PSI sections. The MPEG-2 CRC shall be checked in the POD, and only messages that pass the CRC check shall be forwarded to the Host. The Host shall not check the CRC.

5 TABLE SECTION FORMATS

The following sections define the formats of table sections as they are delivered across an Extended Channel Interface.

5.1 *Network Information Table*

Sections of the Network Information Table shall be associated on the POD-Host interface with PID value 0x1FFC, the SI_base PID. This table delivers sections of non-textual tables applicable system-wide. The table types included are the Carrier Definition Subtable (CDS) and the Modulation Mode Subtable (MMS).

Table 5.1 shows the format of the Network Information Table section.

table_ID — The table_ID of the Network Information Table section shall be 0xC2.

first_index — An 8-bit unsigned integer number in the range one to 255 that indicates the index of the first record to be defined in this table section. If more than one record is provided, the additional records define successive table entries following first_index. The value zero is illegal and shall not be specified.

number_of_records — An 8-bit unsigned integer number that specifies the number of records being defined in this table section. The maximum is limited by the maximum allowed length of the table section.

transmission_medium — This 4-bit field shall be set to zero (0x0).

Table 5.1 Network Information Table section format

	Bits	Bytes	Format
network_info_table_section(){			
table_ID	8	1	uimsbf value 0xC2
zero	2	2	Bslbf
reserved	2		Bslbf
section_length	12		Uimsbf
zero	3	1	Bslbf
protocol_version	5		Sec. 4.4.1
first_index	8	1	uimsbf range 1-255
number_of_records	8	1	Uimsbf
transmission_medium	4	1	uimsbf
table_subtype	4		uimsbf see Table 5.2
for (i=0; i<number_of_records; i++) {			
if (table_subtype==CDS) {			
CDS_record()		((5))	
}			
if (table_subtype==MMS) {			
MMS_record()		((6))	
}			
descriptors_count	8	(1)	uimsbf range 0-255
for (i=0; i<descriptors_count; i++) {			
descriptor()	*	((*))	optional
}			
}			
for (i=0; i<N; i++) {			
descriptor()	*	(*)	optional
}			
CRC_32	32	4	rpchof
}			

table_subtype — A 4-bit value that defines the type of table delivered in the table section. One instance of a Network Information Table section can define entries within at most one type of table. The table_subtype parameter is defined in Table 5.2.

Table 5.2 Network Information Table Subtype

table_subtype	meaning
0	invalid
1	CDS — Carrier Definition Subtable
2	MMS — Modulation Mode Subtable
3-15	Reserved

The receiver shall discard a Network Information Table section with table_subtype indicating an unknown or unsupported table_subtype.

5.1.1 Carrier Definition Subtable (CDS)

Table 5.3 defines the structure of the CDS_record(). Each CDS defines a set of carrier frequencies. A full frequency plan table shall be constructed from one or more CDS_record() structures, each defining a starting frequency, a number of carriers, and a frequency spacing for carriers in this group.

The specified carrier represents the nominal center of the spectral band for all modulation methods, including analog. Carrier frequencies in the table thus represent the data carrier frequency for digital transmissions modulated using QAM or PSK.⁵

Each CDS_record represents a definition of N carriers. The first_index parameter reflects the index in a flat space between 1 and 255, representing the first carrier in the CDS_record. Starting from the first CDS_record defining carriers C₁, C₂, C₃, ..., C_N, where N = number_of_carriers, the carrier index for C₁ is equal to first_index + I – 1. If the table section includes more than one CDS_record(), the carrier index of the second CDS_record would be first_index plus the number of carriers defined in the first CDS_record(), namely, first_index + number_of_carriers. References to the Carrier Definition Subtable, such as the CDS_reference in the virtual_channel() of Table 5.17, are to the carrier index (a carrier defined within a CDS_record()), between 1 and N, where N is normally much smaller than 255. These references are *not* to the index of a CDS_record() itself, which is sequenced from first_index and is not reset to 1 until it exceeds 255.

Note that the carriers, as defined by one or more CDS_record()s, may or may not end up sorted in the order of increasing carrier frequency. Certain frequency plans may be specified by overlapping two or more CDS_record()s, each of which defines equally-spaced carriers.

Note also that carriers may be defined that are currently not in use. To facilitate the compressed delivery format, defined carriers may not reflect reality. An example: carriers at 1, 2, 4, 5, 7, 8 MHz could be defined as eight carriers at 1MHz spacing (3 MHz and 6 MHz do not really exist, or are not currently in use).

Table 5.3 CDS record format

	Bits	Bytes	Format
CDS_record(){			
number_of_carriers	8	1	uimsbf
spacing_unit	1	2	bslbf see Table 5.4
zero	1		bslbf
frequency_spacing	14		uimsbf range 1-16,383 units of 10 or 125kHz
frequency_unit	1	2	bslbf see Table 5.5
first_carrier_frequency	15		uimsbf range 0-32,767 units of 10 or 125kHz
}			

number_of_carriers — An unsigned integer in the range 1 to 255 that represents the number of carriers whose frequency is being defined by this CDS_record().

spacing_unit — A 1-bit field identifying the units for the frequency_spacing field. Table 5.4 defines the coding for spacing_unit.

⁵ Note that transmission systems using VSB modulation transmit spectra are not symmetrical about the carrier or pilot tone. Acquisition of a VSB-modulated signal involves computation of the pilot tone (or in analog VSB, the picture carrier) location relative to the center of the band. For example, for the ATSC Digital Television Standard (Ref. [3]), where the channel bandwidth is 6 MHz, the pilot tone is located 310 kHz above the lower edge of the channel, or 2.690 MHz below the specified center of the band. Similarly, for analog NTSC, the picture carrier is 1.25 MHz above the lower edge of the channel, or 1.75 MHz below the specified center of the band.

Table 5.4 Spacing Unit

spacing_unit	meaning
0	10 kHz spacing
1	125 kHz spacing

frequency_spacing — A 14-bit unsigned integer number in the range one to 16,383 that defines the frequency spacing in units of either 10 kHz or 125 kHz, depending upon the value of the spacing_unit parameter. If spacing_unit is zero, indicating 10 kHz, then a value of one indicates 10 kHz spacing; two indicates 20 kHz, and so on. If the number_of_carriers field is one, the frequency_spacing field is ignored. The maximum frequency spacing that can be represented is $(2^{14}-1) * 125 \text{ kHz} = 2047.875 \text{ MHz}$. The minimum frequency spacing is 10 kHz.

frequency_unit — A 1-bit field identifying the units for the first_carrier_frequency field. Table 5.5 defines the coding for frequency_unit.

Table 5.5 Frequency Unit

Frequency_unit	meaning
0	10 kHz units
1	125 kHz units

first_carrier_frequency — A 15-bit unsigned integer number in the range 0 to 32,767 that defines the starting carrier frequency for the carriers defined in this group, in units of either 10 kHz or 125 kHz, depending on the value of frequency_unit. If only one carrier is defined for the group, the first_carrier_frequency represents its frequency. When the frequency_unit indicates 125 kHz, the first_carrier_frequency can be interpreted as a fractional frequency (1/8 MHz) in the least-significant 3 bits, and an integer number of megahertz in the upper 12 bits. The range of frequencies that can be represented is $0 \text{ to } (2^{15}-1) * 125 \text{ kHz} = 4095.875 \text{ MHz}$.

5.1.2 Modulation Mode Subtable (MMS)

Table 5.6 defines the structure of the MMS_record().

Table 5.6 MMS record format

	Bits	Bytes	Format
MMS_record(){			
transmission_system	4	1	uimbsf see Table 5.7
inner_coding_mode	4		uimbsf see Table 5.8
split_bitstream_mode	1	1	bslbf {no, yes}
zero	2		bslbf
modulation_format	5		uimbsf see Table 5.9
zero	4	4	bslbf
symbol_rate	28		uimbsf units: symbols per sec.
}			

transmission_system — A 4-bit field that identifies the transmission standard employed for the waveform defined by this MMS record. Table 5.7 defines the coding for transmission_system.

Table 5.7 Transmission System

transmission_system	meaning
0	unknown — The transmission system is not known.
1	Reserved (ETSI)
2	ITU-T annex B — The transmission system conforms to the ITU North American standard specified in Annex B of ITU Rec. J.83 [10 ~~22~~].
3	Defined for use in other systems
4	ATSC — The transmission system conforms to the ATSC Digital Television Standard [3].
5-15	Reserved (satellite)

inner_coding_mode — A 4-bit field that indicates the coding mode for the inner code associated with the waveform described in this MMS record. The following values are currently defined: 5/11, 1/2, 3/5, 2/3, 3/4, 4/5, 5/6, and 7/8. Coding of the inner_coding_mode field is shown in Table 5.8.

modulation_format — A 5-bit field that defines the basic modulation format for the carrier. Table 5.9 defines the parameter.

Table 5.8 Inner Coding Mode

inner_coding_mode	meaning
0	rate 5/11 coding
1	rate 1/2 coding
2	Reserved
3	rate 3/5 coding
4	Reserved
5	rate 2/3 coding
6	Reserved
7	rate 3/4 coding
8	rate 4/5 coding
9	rate 5/6 coding
10	Reserved
11	rate 7/8 coding
12-14	Reserved
15	none — indicates that the waveform does not use concatenated coding

Table 5.9 Modulation Format

modulation_format	meaning
0	unknown — The modulation format is unknown.
1	QPSK — The modulation format is QPSK (Quadrature Phase Shift Keying).
2	BPSK — The modulation format is BPSK (Binary Phase Shift Keying).
3	OQPSK — The modulation format is offset QPSK.
4	VSF 8 — The modulation format is 8-level VSB (Vestigial Sideband).
5	VSF 16 — The modulation format is 16-level VSB.
6	QAM 16 — Modulation format 16-level Quadrature Amplitude Modulation (QAM).
7	QAM 32 — 32-level QAM
8	QAM 64 — 64-level QAM
9	QAM 80 — 80-level QAM
10	QAM 96 — 96-level QAM
11	QAM 112 — 112-level QAM
12	QAM 128 — 128-level QAM
13	QAM 160 — 160-level QAM
14	QAM 192 — 192-level QAM
15	QAM 224 — 224-level QAM
16	QAM 256 — 256-level QAM
17	QAM 320 — 320-level QAM
18	QAM 384 — 384-level QAM
19	QAM 448 — 448-level QAM
20	QAM 512 — 512-level QAM
21	QAM 640 — 640-level QAM
22	QAM 768 — 768-level QAM
23	QAM 896 — 896-level QAM
24	QAM 1024 — 1024-level QAM
25-31	Reserved

symbol_rate — A 28-bit unsigned integer field that indicates the symbol rate in symbols per second associated with the waveform described in this MMS record.

5.1.3 Descriptors Count

descriptors_count — An 8-bit unsigned integer value in the range 0 to 255 representing the number of descriptor blocks to follow.

descriptor() — The table section may include at its end one or more structures of the form tag, length, data. The number of descriptors present is determined indirectly by processing the **section_length** field. Descriptors are defined in Section 6.

5.2 Network Text Table

Sections of the Network Text Table shall be associated on an Extended Channel Interface with PID value 0x1FFC, the **SI_base** PID. This table delivers sections of textual tables applicable system-wide. Each instance of Network Text Table is associated with a language, as such the textual information may be provided multi-lingually. The Network Text Table delivers the Source Name Subtable (SNS).

Table 5.10 shows the format of the Network Text Table.

The Network Text Table carries Multilingual Text Strings, formatted as defined in Section 7.2. Text strings included in the Network Text Table shall not include format effectors (defined in Section 7.1.2). If format effectors are present in a text block, the Host is expected to disregard them.

Table 5.10 Network Text Table section format

	Bits	Bytes	Format
network_text_table_section(){			
table_ID	8	1	uimbsf value 0xC3
zero	2	2	bslbf
reserved	2		bslbf
section_length	12		uimbsf
zero	3	1	
protocol_version	5		see Sec. 4.4.1
ISO_639_language_code	24	3	per ISO 639.2/B
transmission_medium	4	1	uimbsf
table_subtype	4		uimbsf see Table 5.11
if (table_subtype==SNS) {			
source_name_subtable()	*	(*)	
}			
for (i=0; i<N; i++) {			
descriptor()	*	(*)	optional
}			
CRC_32	32	4	rpchof
}			

table_ID — The table_ID of the Network Text Table section shall be 0xC3.

ISO_639_language_code — A 3-byte language code per ISO 639.2/B defining the language associated with the text carried in this Network Text Table. The ISO_639_language_code field contains a three-character code as specified by ISO 639.2/B. Each character is coded into 8 bits according to ISO 8859-1 (ISO Latin-1) and inserted, in order, into the 24-bit field. The value 0xFFFFFFFF shall be used in case the text is available in one language only. The value 0xFFFFFFFF shall represent a “wild card” match when filtering by language.

transmission_medium — This 4-bit field shall be set to zero (0x0).

table_subtype — A 4-bit value that defines the type of table delivered in the table section. One instance of a Network Text Table section can define entries within at most one type of table. The table_subtype parameter is defined in Table 5.11.

Table 5.11 Network Text Table Subtype

table_subtype	meaning
0	invalid
1-5	Reserved
6	SNS — Source Name Subtable
7-15	Reserved

A Host shall discard a Network Text Table section with `table_subtype` indicating an unknown or unsupported value.

The SNS can provide a textual name associated with each service defined in the Short-form Virtual Channel Table, by reference to its `source_ID`. The format of the `source_name__subtable()` is given in Table 5.12.

number_of_SNS_records — An unsigned 8-bit integer number in the range 1 to 255 that specifies the number of records being defined in this table section.

application_type — A Boolean flag, when set, indicates that the name string being defined is for an application of the given `application_ID`. When the flag is clear, the name string being defined is for a source of the given `source_ID`. Support for application-type virtual channels is optional. Hosts not supporting application-type virtual channels may disregard name strings associated with these VC. Support for application-type virtual channels is beyond the scope of this standard.

application_ID — A 16-bit unsigned integer value identifying the application associated with the name string that follows. This field may be disregarded by Hosts not supporting application-type virtual channels.

source_ID — A 16-bit unsigned integer value identifying the programming source associated with the source name to follow.

name_length — An unsigned 8-bit integer number in the range 1 to 255 that defines the number of bytes in the `source_name()` that follows.

source_name() — A Multilingual Text String defining the name of the source or application, formatted according to the rules defined in Section 7.1.

Table 5.12 Source Name Subtable format

	Bits	Bytes	Format
source_name_subtable(){			
number_of_SNS_records	8	1	uimsbf range 1-255
for (i=0; i<number_of_SNS_records; i++) {			
application_type	1	(1)	bslbf {false, true}
zero	7		bslbf
if (application_type) {			
application_ID	16	((2))	uimsbf
} else {			
source_ID	16	((2))	uimsbf
}			
name_length	8	(1)	size of source_name() (L)
source_name()	L*8	(L)	multilingual text
SNS_descriptors_count	8	(1)	uimsbf range 0-255
for (i=0; i<SNS_descriptors_count; i++) {			
descriptor()	*	((*))	
}			
}			
}			

SNS_descriptors_count — An unsigned 8-bit integer number, in the range 0 to 255, that defines the number of descriptors to follow.

descriptor() — The table section may include, at its end, one or more structures of the form tag, length, data. The number of descriptors present is determined indirectly by processing the `section_length` field. Descriptors are defined in Section 6.

5.3 Short-form Virtual Channel Table Section

The Short-form Virtual Channel Table sections deliver portions of the Virtual Channel Map (VCM), the Defined Channels Map (DCM) and the Inverse Channel Map (ICM). Sections of the Short-form Virtual Channel Table shall be associated on an Extended Channel Interface with PID value 0x1FFC, the `SI_base` PID.

Table 5.13 shows the syntax of the Short-form Virtual Channel Table section.

Table 5.13 Short-form Virtual Channel Table section format

	Bits	Bytes	Format
shortform_virtual_channel_table_section(){			
table_ID	8	1	uimsbf value 0xC4
zero	2	2	bslbf
reserved	2		bslbf
section_length	12		uimsbf
zero	3	1	bslbf
protocol_version	5		see Sec. 4.4.1
transmission_medium	4	1	uimsbf
table_subtype	4		uimsbf see Table 5.14
VCT_ID	16	2	uimsbf
if (table_subtype==DCM) {			
DCM_structure()	*	(*)	
}			
if (table_subtype== VCM) {			
VCM_structure()	*	(*)	
}			
if (table_subtype== ICM) {			
ICM_structure()	*	(*)	
}			
for (i=0; i<N; i++) {			
descriptor()	*	(*)	optional
}			
CRC_32	32	4	rpchof
}			

table_ID — The `table_ID` of the Short-form Virtual Channel Table shall be 0xC4.

transmission_medium — This 4-bit field shall be set to zero (0x0).

table_subtype — A 4-bit field that indicates the map type being delivered in this S-VCT section. Three map types are currently defined, the Virtual Channel Map (VCM), the Defined Channels Map (DCM), and the Inverse Channel Map (ICM). Table 5.14 defines `table_subtype`.

Table 5.14 S-VCT Table Subtypes

table_subtype	meaning
0	VCM — Virtual Channel Map
1	DCM — Defined Channels Map
2	ICM — Inverse Channel Map
3-15	Reserved

An S-VCT section received with `table_subtype` indicating an unknown or unsupported map type shall be discarded.

VCT_ID — A 16-bit unsigned integer value, in the range 0x0000 to 0xFFFF, indicating the VCT to which the channel definitions in this table section apply. This 16-bit field may be used by the POD module for filtering purposes. Only one version of the S-VCT, corresponding to one value of `VCT_ID`, shall be delivered to the Host across the Extended Channel interface at a given time.

5.3.1 Defined Channels Map

Table 5.15 shows the format of the `DCM_structure()`.

Table 5.15 DCM structure format

	Bits	Bytes	Format
DCM_structure(){			
zero	4	2	bslbf
first_virtual_channel	12		uimsbf range 0-4095
zero	1	1	bslbf
DCM_data_length	7		uimsbf range 1-127
for (i=0; i<DCM_data_length; i++) {			
range_defined	1	(1)	bslbf {no, yes}
channels_count	7		uimsbf range 1-127
}			
}			

first_virtual_channel — An unsigned 12-bit integer reflecting the first virtual channel whose existence is being provided by this table section, for the map identified by the `VCT_ID` field. The range is 0 to 4095.

DCM_data_length — A 7-bit unsigned integer number, in the range 1 to 127, that defines the number of DCM data fields to follow in the table section.

The DCM data bytes taken as a whole define which virtual channels, starting at the channel number defined by `first_virtual_channel`, are defined and which are not. Each `DCM_data_field` defines two pieces of data: a flag indicating whether this block of channels is defined or not, and the number of channels in the block. The bytes are interpreted in an accumulative way, with a pointer into the Short-form Virtual Channel Table which is initialized to `first_virtual_channel`. As each byte is processed, the pointer is incremented by the number of channels indicated by the `channels_count` field.

For example, if channels 2-90, 200-210, 400-410, 600-610, 800-810, and 999 were defined, and `first_virtual_channel` was zero, the DCM data sequence (in decimal) would be the

following, where underlined numbers have the `range_defined` bit set: 2, 89, 109, 11, 127, 62, 11, 127, 62, 11, 127, 62, 11, 127, 61, 1.

range_defined — A Boolean flag that indicates, when true, that the number of channels given by `channels_count` is defined in the VCT, starting at the current pointer value. When the flag is clear, the number of channels equal to `channels_count` are currently not defined starting at the current pointer value.

channels_count — An unsigned 7-bit integer number, in the range one to 127, that indicates the number of defined (or undefined) channels in a group.

5.3.2 Virtual Channel Map

Table 5.16 shows the format of the `VCM_structure()`.

Table 5.16 VCM structure format

	Bits	Bytes	Format
<code>VCM_structure(){</code>			
zero	2	1	bslbf
descriptors_included	1		bslbf {no, yes}
zero	5		bslbf
splice	1	1	bslbf {no, yes}
zero	7		bslbf
activation_time	32	4	uimsbf
number_of_VC_records	8	1	
for (i=0; i<number_of_VC_records; i++) {			
virtual_channel()	*	(*)	
}			
<code>}</code>			

descriptors_included — A Boolean flag that indicates, when set, that one or more record-level descriptors are present in the table section. Record-level descriptors are those defined in Table 5.17 following the “if (`descriptors_included`)” statement. When the flag is clear, the record-level descriptor block is absent. The `descriptors_included` flag is not applicable to the section level descriptors shown at the bottom of Table 5.13.

The activation time indicates the time at which the data delivered in the table section will be valid.

splice — A Boolean flag that indicates, when set, that the Host should arm video processing hardware to execute the application of the data delivered in the `VCM_structure()` at the next MPEG-2 video splice point if the virtual channel changes described in the table section apply to a currently acquired channel, and the `activation_time` is reached. If the activation is immediate or specified as a time that has since passed, the data should be applied immediately. When the `splice` flag is clear, the virtual channel change is made directly, without arming video hardware for a splice.

activation_time — A 32-bit unsigned integer field providing the absolute second the virtual channel data carried in the table section will be valid, defined as the number of seconds since 0000 Hours UTC, January 6th, 1980. If the `GPS_UTC_offset` delivered in the System Time Table is zero, `activation_time` includes the correction for leap seconds. Otherwise, `activation_time` can be converted to

UTC by subtracting the `GPS.UTC_offset`. If the `activation_time` is in the past, the data in the table section shall be considered valid immediately. An `activation_time` value of zero shall be used to indicate immediate activation.

A Host may enter a virtual channel record whose activation times are in the future into a queue. Such a queue may be called a *pending virtual channel* queue. Hosts are not required to implement a pending virtual channel queue, and may choose to discard any data that is not currently applicable.

number_of_VC_records — An 8-bit unsigned integer number, in the range 1 to 255, that identifies the number of `virtual_channel()` records to follow in the table section. The number of records included is further limited by the allowed maximum table section length.

virtual_channel() — Table 5.17 defines the `virtual_channel()` record structure.

Table 5.17 Virtual channel record format

	Bits	Bytes	Format
virtual_channel(){			
zero	4	2	bslbf
virtual_channel_number	12		uimbsbf range 0-4095
application_virtual_channel	1	1	bslbf {no, yes}
zero	1		bslbf
path_select	1		bslbf see Table 5.18
transport_type	1		bslbf see Table 5.19
channel_type	4		uimbsbf see Table 5.20
if (application_virtual_channel) {			
application_ID	16	(2)	
} else {			
source_ID	16	(2)	
}			
if (transport_type==MPEG_2) {			
CDS_reference	8	((1))	uimbsbf range 1-255
program_number	16	((2))	
MMS_reference	8	((1))	uimbsbf range 1-255
} else { /* non-MPEG-2 */			
CDS_reference	8	((1))	uimbsbf range 0-255
scrambled	1	((1))	bslbf {no, yes}
zero	3		bslbf
video_standard	4		uimbsbf see Table 5.21
zero	16	((2))	bslbf
}			
if (descriptors_included) {			
descriptors_count	8	(1)	uimbsbf
for (i=0; i<descriptors_count; i++) {			
descriptor()	*	((*))	
}			
}			
}			

virtual_channel_number — An unsigned 12-bit integer, in the range zero to 4095, reflecting the virtual channel whose definition is being provided by this virtual channel record, for the map identified by the `VCT_ID` field.

application_virtual_channel — A binary flag that, when set, indicates this virtual channel defines an access point represented by the `application_ID`. When the flag is clear, the channel is not an application access point, and this virtual channel defines an access point represented by the `source_ID`. Support for application-type virtual channels is optional. Hosts not supporting application-type virtual channels may disregard all data associated with them. Support for application-type virtual channels is beyond the scope of this standard.

path_select — A 1-bit field that associates the virtual channel with a transmission path. For the cable transmission medium, `path_select` identifies which physical cable carries the Transport Stream associated with this virtual channel. Table 5.18 defines `path_select`.

Table 5.18 Path Select

<code>path_select</code>	meaning
0	path 1
1	path 2

transport_type — A 1-bit field identifying the type of transport carried on this carrier as either being an MPEG-2 transport (value zero), or not (value one). Table 5.19 defines the coding.

Table 5.19 Transport Type

<code>transport_type</code>	meaning
0	MPEG-2 transport
1	non-MPEG-2 transport

channel_type — A 4-bit field defining the channel type. Table 5.20 defines `channel_type`.

Table 5.20 Channel Type

<code>channel_type</code>	meaning
0	normal — Indicates that the record is a regular virtual channel record. For non-MPEG-2 channels, the <code>waveform_type</code> shall be defined as “normal.”
1	hidden — Indicates that the record identifies a virtual channel that may not be accessed by the user by direct entry of the channel number (hidden). Hidden channels are skipped when the user is channel surfing, and appear as if undefined if accessed by direct channel entry. Programs constructed for use by specific applications (such as NVD theaters) utilize hidden virtual channels. If a <code>channel_properties_descriptor()</code> is present and the <code>hide_guide</code> bit is 0, the channel may be considered to be <i>inactive</i> . Inactive channels may appear in EPG displays.
2-15	reserved — Hosts are expected to treat virtual channel records of unknown <code>channel_type</code> the same as non-existent (undefined) channels.

application_ID — A 16-bit unsigned integer number, in the range 0x0001 to 0xFFFF, that identifies the application associated with the virtual channel, on a system-wide basis. One particular program guide application, for example, may look for a program carrying data in its native transmission format by searching through the Short-form Virtual Channel Table for a match on its assigned `application_ID`. In some cases, one application may be able to process streams

associated with more than one application ID. The application ID may be used to distinguish content as well as format, for the benefit of processing within the application. The value zero for `application_ID` shall not be assigned; if specified in a Virtual Channel record, the value zero indicates “unknown” or “inapplicable” for the `application_ID/source_ID` field.

Support for application-type virtual channels is optional. Hosts not supporting application-type virtual channels may disregard all data associated with them. Support for application-type virtual channels is beyond the scope of this standard.

source_ID — A 16-bit unsigned integer number, in the range 0x0000 to 0xFFFF, that identifies the programming source associated with the virtual channel, on a system-wide basis. In this context, a *source* is one specific source of video, text, data, or audio programming. For the purposes of referencing virtual channels to the program guide database, each such program source is associated with a unique value of `source_ID`. The `source_ID` itself may appear in an EPG database, where it tags entries to specific services. The value zero for `source_ID`, if used, shall indicate the channel is not associated with a source ID.

program_number — A 16-bit unsigned integer number that associates the virtual channel number being defined with services defined in the Program Association and TS Program Map Table sections. Access to elementary streams defined in each virtual channel record involves first acquiring the Transport Stream on the carrier associated with the virtual channel, then referencing the Program Association section in PID 0 to find the PID associated with the TS Program Map Table section for this `program_number`. PIDs for each elementary stream are then found by acquisition of the TS Program Map Table section.

A `program_number` with value 0x0000 (invalid as a regular program number) is reserved to indicate that the Host is expected to discard the corresponding virtual channel record from the queue of pending virtual channel changes. Records are identified in the pending queue by their `activation_time`, `VCT_ID`, and `virtual_channel_number`. If no pending virtual channel change is found in the Host’s queue, no action should be taken for this virtual channel (i.e. the record is expected to be discarded).

For inactive channels (those not currently present in the Transport Stream), `program_number` shall be set to zero. This number shall **not** be interpreted as pointing to a Program Map Table entry.

descriptors_count — An 8-bit unsigned integer value, in the range 0 to 255, that defines the number of descriptors to follow.

CDS_reference — An unsigned 8-bit integer number, in the range 0 to 255, that identifies the frequency associated with this virtual channel. Values 1 to 255 of `CDS_reference` are used as indices into the Carrier Definition Subtable to find a frequency to tune to acquire the virtual channel. The value zero is reserved to indicate that the referenced service is carried on *all* digital multiplexes in this VCM. The `CDS_reference` field shall be disregarded for inactive channels.

MMS_reference — An 8-bit unsigned integer value, in the range 0 to 255, that references an entry in the Modulation Mode Subtable (MMS). The value zero is illegal and shall not be specified. For digital waveforms, the `MMS_reference` associates the carrier with a digital modulation mode. For Host implementations that support only one set of modulation parameters, in systems in

which one modulation method is used for all carriers, storage and processing of the `MMS_reference` is unnecessary. The `MMS_reference` field shall be disregarded for inactive channels.

video_standard — A 4-bit field that indicates the video standard associated with this non-Standard virtual channel. Table 5.21 defines `video_standard`.

Table 5.21 Video Standard

video_standard	meaning
0	NTSC — The video standard is NTSC
1	PAL 625 — The video standard is 625-line PAL
2	PAL 525 — The video standard is 525-line PAL
3	SECAM — The video standard is SECAM
4	MAC — The video standard is MAC
5-15	Reserved

descriptor() — The table section may include, at its end, one or more structures of the form tag, length, data. The number of descriptors present is determined indirectly by processing the `section_length` field. Descriptors are defined in Section 6.

5.3.3 Inverse Channel Map

The Inverse Channel Map, once reconstructed in the Host from a sequence of Virtual Channel records that belong to the ICM, consists of a list of `source_ID/virtual_channel_number` pairs, ordered by `source_ID`. The Host may use this table to quickly find the virtual channel carrying the program given by a particular value of `source_ID` (by binary search), if such a virtual channel exists. One Inverse Channel Map can be defined per Virtual Channel Map. The ICM may be constructed from the VCM, or linear searches may be done to resolve `source_ID` references. Transmission of the ICM is therefore optional.

Virtual channels that provide access points for applications (i.e., with the `application_virtual_channel` flag set to “yes”) are not included in the ICM.

Table 5.22 describes the format of the `ICM_structure()`.

Table 5.22 ICM structure format

	Bits	Bytes	Format
ICM_structure(){			
zero	4	2	bslbf
first_map_index	12		uimsbf range 0-4095
zero	1	1	bslbf
record_count	7		uimsbf range 1-127
for (i=0; i<record_count; i++) {			
source_ID	16	(2)	uimsbf
zero	4	(2)	bslbf
virtual_channel_number	12		uimsbf range 0-4095
}			
}			

first_map_index — A 12-bit unsigned integer, in the range 0 to 4095, that represents the index into the Inverse Channel Map where data carried in this ICM_structure() should be stored.

record_count — A 7-bit unsigned integer value, in the range 1 to 127, that represents the total number of source_ID/ virtual_channel pairs defined in this table section.

source_ID — A 16-bit unsigned integer number, in the range 0x0000 to 0xFFFF, that identifies the source associated with the virtual channel, on a system-wide basis. In this context, a “source” is one specific source of video, text, data, or audio programming. For the purposes of referencing virtual channels to the program guide database, each such source is associated with a unique value of source_ID. .

virtual_channel_number — A 12-bit unsigned integer value, in the range 0 to 4095, that represents the virtual channel, in the Short-form Virtual Channel Table section (ref. Table 5.13) given by VCT_ID, associated with the given source_ID through the virtual_channel() record (ref. Table 5.17). A virtual_channel_number of zero indicates that the program given by source_ID is currently not carried in this Short-form Virtual Channel Table. Such placeholders are useful in the case where the existence of a certain program within a VCM may come and go.

5.4 System Time Table Section

The System Time Table is used to synchronize Hosts with accurate calendar time. The System Time Table shall be associated on an Extended Channel Interface with PID value 0x1FFC, the SI_base PID. Rate of transmission is typically once per minute, at second 00 of each minute.

The processing of the System Time Table in the Host is time-critical. Delays between reception and processing of the table section increase the inaccuracy of timed events. Processing delays should be kept below 200 milliseconds.

Table 5.23 shows the format of the System Time Table section.

Table 5.23 System Time Table section format

	Bits	Bytes	Format
system_time_table_section(){			
table_ID	8	1	uimsbf value 0xC5
zero	2	2	bslbf
reserved	2		bslbf
section_length	12		uimsbf
zero	3	1	
protocol_version	5		see Sec. 4.4.1
zero	8	1	bslbf
system_time	32	4	uimsbf
GPS_UTC_offset	8	1	uimsbf seconds
for (i=0; i<N; i++) {			
descriptor()	*	(*)	optional
}			
CRC_32	32	4	rpchof
}			

table_ID — The `table_ID` of the System Time Table shall be 0xC5.

system_time — A 32-bit unsigned integer quantity representing the current system time, as the number of GPS seconds since 0000 Hours UTC, January 6th, 1980. The `system_time` value may or may not include the correction factor for leap seconds, depending upon the value of `GPS_UTC_offset`, as described below.

GPS_UTC_offset — An 8-bit value that serves dual roles. When set to zero, the field indicates that the `system_time` field carries UTC time directly. When `GPS_UTC_offset` is not equal to zero, it is interpreted as an 8-bit unsigned integer that defines the current offset in whole seconds between GPS and UTC time standards. To convert GPS time to UTC, the `GPS_UTC_offset` is subtracted from GPS time. Whenever the International Bureau of Weights and Measures decides that the current offset is too far in error, an additional leap second may be added (or subtracted), and the `GPS_UTC_offset` will reflect the change.

descriptor() — The table section may include at its end one or more structures of the form tag, length, data. The number of descriptors present is determined indirectly by processing the `section_length` field. Descriptors are defined in Section 6.

5.5 Master Guide Table (MGT)

The Master Guide Table is used to indicate the location, size, and version of tables it references. The MGT shall be associated on an Extended Channel Interface with PID value 0x1FFC, the `SI_base` PID. The MGT syntax is shown in Table 5.24. Syntax and semantics are identical to [5], except that additional table types are added to refer to all tables defined in this protocol.

table_ID — The `table_ID` of the Master Guide Table section shall be 0xC7.

section_syntax_indicator — This 1-bit field shall be set to ‘1’. It denotes that the section follows the generic section syntax beyond the section length field.

private_indicator — This 1-bit field shall be set to ‘1’.

section_length — 12-bit field specifying the number of remaining bytes in this section immediately following the `section_length` field up to the end of the section. The value of the `section_length` shall be no larger than 4,093.

map_ID — This 16-bit field may be used by the POD module for filtering purposes. The Host is expected to ignore `map_ID`. Only one version of the MGT, corresponding to one value of `map_ID` shall be delivered to the Host across an Extended Channel Interface at a given time. Consequently, the Host may disregard `map_ID` and may process the MGT `version_number` field as an indication that the MGT version has changed.

Note: The `map_ID` may be considered to be an identifier for this instance of the Master Guide Table. In some applications, the POD module may receive multiple Master Guide Table sections corresponding to distinct channel maps. In this case, the POD module is responsible for accepting one MGT and discarding the others using information provided by means outside the scope of this standard.

Table 5.24 Master Guide Table section format

	Bits	Bytes	Format
master_guide_table_section () {			
table_ID	8	1	0xC7
section_syntax_indicator	1	2	'1'
private_indicator	1		'1'
reserved	2		'11'
section_length	12		uimbsf
map_ID	16	2	uimbsf
reserved	2	1	'11'
version_number	5		uimbsf
current_next_indicator	1		'1'
section_number	8	1	0x00
last_section_number	8	1	0x00
protocol_version	8	1	uimbsf
tables_defined	16	2	uimbsf
for (i=0;i<tables_defined;i++) {			
table_type	16	2	uimbsf
reserved	3	2	'111'
table_type_PID	13		uimbsf
reserved	3	1	'111'
table_type_version_number	5		uimbsf
number_bytes	32	4	uimbsf
reserved	4	2	'1111'
table_type_descriptors_length	12		uimbsf
for (k=0;k<N;k++)			
descriptor()	var		
}			
reserved	4	2	'1111'
descriptors_length	12		uimbsf
for (I = 0;I< N;I++)			
descriptor()	var		
CRC_32	32	4	rpchof
}			

version_number — This 5-bit field is the version number of MGT. The version number shall be incremented by 1 modulo 32 when any field in the `table_types` defined in the loop below or the MGT itself changes.

current_next_indicator — This 1-bit indicator is always set to '1' for the MGT section; the MGT sent is always currently applicable.

section_number — The value of this 8-bit field shall always be 0x00 (this table is only one section long).

last_section_number — The value of this 8-bit field shall always be 0x00.

protocol_version — An 8-bit unsigned integer field whose function shall be to allow, in the future, this table type to carry parameters that may be structured differently than those defined in the current protocol. At present, the only valid value for `protocol_version` is zero. Non-zero values of `protocol_version` may only be processed by Hosts designed to accommodate the later versions as they become standardized.