Doc. A/65B
18 March 2003

ATSC Standard:

Program and System Information Protocol for Terrestrial Broadcast and Cable (Revision B)

Advanced Television Systems Committee
1750 K Street, N.W.
Suite 1200
Washington, D.C. 20006
http://www.atsc.org

The Advanced Television Systems Committee, Inc., is an international, non-profit organization developing voluntary standards for digital television. The ATSC member organizations represent the broadcast, broadcast equipment, motion picture, consumer electronics, computer, cable, satellite, and semiconductor industries.

Specifically, ATSC is working to coordinate television standards among different communications media focusing on digital television, interactive systems, and broadband multimedia communications. ATSC is also developing digital television implementation strategies and presenting educational seminars on the ATSC standards.

ATSC was formed in 1982 by the member organizations of the Joint Committee on InterSociety Coordination (JCIC): the Electronic Industries Association (EIA), the Institute of Electrical and Electronic Engineers (IEEE), the National Association of Broadcasters (NAB), the National Cable Television Association (NCTA), and the Society of Motion Picture and Television Engineers (SMPTE). Currently, there are approximately 170 members representing the broadcast, broadcast equipment, motion picture, consumer electronics, computer, cable, satellite, and semiconductor industries.

ATSC Digital TV Standards include digital high definition television (HDTV), standard definition television (SDTV), data broadcasting, multichannel surround-sound audio, and satellite direct-to-home broadcasting.

Editor's Note:

This document was updated on 28 October 2003 to correct editorial errors relating to reference numbers embedded within the document.

Table of Contents

Index of Tables and Figures

ATSC Standard:

Program and System Information Protocol for Terrestrial Broadcast and Cable (Revision B)

1. SCOPE

This document[1] defines a Standard for System Information (SI) and Program Guide (PG) data compatible with digital multiplex bit streams constructed in accordance with ISO/IEC 13818-1 (MPEG-2 Systems). The document defines the standard protocol for transmission of the relevant data tables contained within packets carried in the Transport Stream multiplex. The protocol defined herein is referred to as Program and System Information Protocol (PSIP).

This standard was prepared by the Advanced Television Systems Committee (ATSC) Technology Group on Distribution (T3). The document was approved by the members of the ATSC on 23 December 1997. Revision B to PSIP is the result of incorporating Amendment No. 1A (12 August 2002), Amendment No. 2 (6 August 2002), and Amendment No. 3 (1 April 2002) after their approval by the full ATSC. In addition, further clarifications were incorporated to produce this document. ATSC Standard A/65B was approved by the full membership on (date).

For an informative description of the purpose, concepts, and tables defined in this protocol, first-time readers are encouraged to start with Annex D. In addition, a Recommended Practice on PSIP implementation is available as ATSC document A/69, "ATSC Recommended Practice: Program and System Information Protocol Implementation Guidelines for Broadcasters." A companion document for receiver manufacturers is available from the Consumer Electronics Association: "EIA/CEA-CEB12 PSIP Recommended Practice."

1.1 Application

This document describes tables that shall be applicable to terrestrial (over-the-air) and cable signals. Some PSIP tables apply to terrestrial broadcast, some apply to cable, and others apply to both.

1.1.1 Terrestrial Broadcast

The following PSIP data shall be included in all ATSC-compliant Transport Streams to be transmitted via terrestrial broadcast:

- The Terrestrial Virtual Channel Table (TVCT) defining, at a minimum, MPEG-2 programs embedded in the Transport Stream in which the TVCT is carried.

[1] NOTE: The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to the validity of this claim, or of any patent rights in connection therewith. The patent holder has, however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the publisher. This document will undergo periodic review and may be subject to change by ballot of the ATSC membership

- The Master Guide Table (MGT) defining the type, packet identifiers, and versions for all the other PSIP tables in this Transport Stream, except for the System Time Table (STT).
- The Rating Region Table (RRT) defining the TV parental guideline system referenced by any content advisory descriptor carried within the Transport Stream, except the RRT corresponding to rating_region 0x01 (US + possessions).[2]
- The System Time Table (STT), defining the current date and time of day.
- A service_location_descriptor for each digital virtual channel in the VCT.
- The first four Event Information Tables (EIT-0, EIT-1, EIT-2 and EIT-3) describing 12 hours of events (TV programs), each with a coverage of 3 hours, and including all of the virtual channels listed in the TVCT.

1.1.2 Cable

The following PSIP data shall be included in all ATSC-compliant Transport Streams to be transmitted via cable:

- The Cable Virtual Channel Table (CVCT) defining, at a minimum, the virtual channel structure for the collection of MPEG-2 programs embedded in the Transport Stream in which the CVCT is carried.
- The Master Guide Table (MGT) defining the type, packet identifiers, and versions for all of the other PSIP tables included in this Transport Stream except for the System Time Table (STT).
- The Rating Region Table (RRT) defining the TV parental guideline system referenced by any content advisory descriptor carried within the Transport Stream, except the RRT corresponding to rating_region 0x01 (US + possessions).[3]
- The System Time Table (STT), defining the current date and time of day.

1.2 Organization

The sections of this document are organized as follows:

- **Section 1** — Provides this general introduction.
- **Section 2** — Lists references and applicable documents.
- **Section 3** — Provides a definition of terms and a list of acronyms and abbreviations used in this document.
- **Section 4** — Describes the data structure of the PSIP tables.
- **Section 5** — Describes the overall table hierarchy.
- **Section 6** — Describes formats for all of the PSIP tables.
- **Section 7** — Describes PSIP STD model.
- **Annex A**— Describes the daylight savings time control.

[2] Note: Interpretation in a receiver of the rating_region 0x01 RRT requires prior knowledge of EIA/CEA-766-A [15]; therefore transmission is unnecessary. A future extension or replacement of the content advisory system for the US is possible by assignment of a new, different rating_region code and creation of new content for an RRT.

[3] See footnote 2.

- **Annex B** — Describes the assignment of major_channel_number values for terrestrial broadcast in the U.S.
- **Annex C** — Describes the standard Huffman tables for text compression.
- **Annex D** — Provides an overview of PSIP for terrestrial broadcast with application examples.
- **Annex E** — Describes the typical sizes of PSIP tables.
- **Annex F** — Provides an overview of Huffman-based text compression.
- **Annex G** — Provides an overview of the use of PSIP for cable.

2. REFERENCES

The following documents are applicable to this Standard:

1) ATSC Standard A/53B (2001), ATSC Digital Television Standard, Revision B; Annex C, Sec. 5.7.2.1. (*normative*).
2) ATSC Standard A/53B (2001), ATSC Digital Television Standard, Revision B; Annexes B and D. (*informative*).
3) ATSC Standard A/55 (1996), Program Guide for Digital Television (*informative*).
4) ATSC Standard A/56 (1996), System Information for Digital Television (*informative*).
5) ATSC Standard A/57 (1996), Program/Episode/Version Identification (*normative*). Readers note that A/57 is being revised.
6) ISO 639.2, Code for the Representation of Names of Languages — Part 2: alpha-3 code, 1998 (*normative*).
7) ISO/IEC 10646-1:2000, Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane (*normative*).
8) ISO/IEC 8859**,** Information Processing — 8-bit Single-Octet Coded Character Sets, Parts 1 through 10 (*normative*).
9) ITU-T Rec. H.222.0 | ISO/IEC 13818-1:1996, Information Technology — Generic coding of moving pictures and associated audio — Part 1: Systems (*normative*).
10) ITU-T Rec. H.262 | ISO/IEC 13818-2:1996, Information Technology — Generic coding of moving pictures and associated audio — Part 2: Video (*normative*).
11) Digital Video Transmission Standard for Cable Television, SCTE DVS-031, Rev. 2, 29 May 1997 (*informative*).
12) EIA-708-A *Specification for Advanced Television Closed Captioning (ATVCC)*, Electronic Industry Association (*normative*).
13) EIA/CEA-608-B, *Line 21 Data Service*, Sec. 9.5.2.4 (*normative*).
14) Record of Test Results for Digital HDTV Grand Alliance System, September 8, 1995, Advanced Television Test Center (*informative*).
15) EIA/CEA-766-A *Specification for U.S. and Canadian Rating Region Table (RRT) and Content Advisory Descriptor for Transport of Content Advisory Information Using ATSC A/65A Program and System Information Protocol (PSIP)*, Electronic Industries Alliance, April 2001 (*informative*).

16) U.S. Code of Federal Regulations (CFR) Title 47, 47CFR11, Emergency Alert System (EAS), U.S. Government Printing Office, Washington, DC 20040, http://www.fcc.gov/wtb/rules.html (*normative*).
17) Federal Information Processing Standard, FIPS Pub 6-4, Counties and Equivalent Entities of the U.S., Its Possessions, and Associated Areas — 90 Aug 31, U.S. Government Printing Office, Washington, DC 20040, http://www.itl.nist.gov/fipspubs (*informative*).
18) The Unicode Standard, Version 3.0, The Unicode Consortium, Addison-Wesley Pub., ISBN 0201616335 (*normative*).
19) Unicode Technical Report #6, A Standard Compression Scheme for Unicode, Revision 3.0, 1999-11-12, The Unicode Consortium (*normative*).
20) ATSC A/90 (2000), Data Broadcast Standard (*informative*).

3. DEFINITIONS

3.1 Compliance Notation

As used in this document, “shall” or “will” 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 at the option of the implementer.

3.2 Acronyms and Abbreviations

The following acronyms and abbreviations are used within this specification:

ATSC Advanced Television Systems Committee
bslbf bit serial, leftmost bit first
BMP basic multilingual plane
CAT conditional access table
CRC cyclic redundancy check
CVCT cable virtual channel table
DCC directed channel change
DCCRR DCC capable DTV reference receiver
DCCSCT DCC selection code table
DET data event table
DTV digital television
EPG electronic program guide
EIT event information table
EMM entitlement management message
ETM extended text message
ETT extended text table
GPS Global Positioning System
PSIP Program and System Information Protocol

MGT master guide table
MPAA Motion Picture Association of America
MPEG Moving Picture Experts Group
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
PTC physical transmission channel
SCTE Society of Cable Telecommunications Engineers
SI system information
STD system target decoder
STT system time table
rpchof remainder polynomial coefficients, highest order first
RRT rating region table
TS transport stream
TVCT terrestrial virtual channel table
unicode Unicode™
UTC Coordinated Universal Time[4]
uimsbf unsigned integer, most significant bit first
VCT virtual channel table. Used in reference to either TVCT or CVCT.

3.3 Definition of Terms

The following terms are used throughout this document:

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 (see Table 6.25). 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.

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

[4] 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.

event A collection of elementary streams with a common time base, an associated start time, and an associated end time. An event is equivalent to the common industry usage of "television program."

instance See *table instance*.

logical channel See *virtual channel*.

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.

physical transmission channel See *physical channel*.

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.

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 commonly used in the context of a "television program" such as a scheduled daily news broadcast. In this specification the term "event" is used to refer to a "television program" to avoid ambiguity.

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

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). All sections begin with the table_id and end with the CRC_32 field, and their starting points within a packet payload are indicated by the pointer_field mechanism defined in the ISO/IEC 13818-1 International Standard.

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).

table The collection of re-assembled sections bearing a common table ID and version number.

table instance Tables are identified by the table_id field. However, in cases such as the Event Information Table, several instances of a table are defined simultaneously. All instances are conveyed in Transport Stream packets of the same PID value and have the same table_id field value. Each instance has a different table_id_extension value. The term table_id_extension is the generic label for the field per Table 4.1; it is renamed as appropriate for specific tables.

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.

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.

4. DATA STRUCTURE

This section describes the data structure common to all PSIP tables.

4.1 Table Format

Tables defined in this Standard are derived from the "long" form of the MPEG-2 private_section defined in sections 2.4.4.10 and 2.4.4.11 of ISO/IEC 13818-1 [9]. The table format common to all PSIP tables is given in Table 4.1. An additional field, protocol_version, has been added as a consistent first byte of every PSIP table section.

Table 4.1 Generic Table Format Used in PSIP

Syntax	No. of Bits	Format
PSIP_section( ) {		
table_id	8	uimsbf
section_syntax_indicator	1	'1'
private_indicator	1	'1'
reserved	2	'11'
section_length	12	uimsbf
table_id_extension	16	uimsbf
reserved	2	'11'
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
protocol_version	8	uimsbf
PSIP_table_data()	*	
CRC_32	32	rpchof
}		

table_id – The value of this 8-bit field shall identify the PSIP Table to which this section belongs. Tables defined in this PSIP Standard, and any created in other ATSC Standards are considered "private" with respect to ISO/IEC 13818-1 and standard with respect to other ATSC Standards. Table 4.2 lists Table ID ranges and values.

Table 4.2 Table ID Ranges and Values (Informative)

Table ID Value (hex)	Tables	PID	Ref.
ISO/IEC 13818-1 Sections:			
0x00	PROGRAM ASSOCIATION table (PAT)	0x0000	Ref. [9]
0x01	CONDITIONAL ACCESS table (CAT)	0x0001	Ref. [9]
0x02	TS PROGRAM MAP TABLE (PMT)	per PAT	Ref. [9]
0x03-0x3F	[ISO reserved]		
User Private Sections:			
0x40-0xBF	[User Private]		
Other documents:			
0xC0-0xC6	[ATSC coordinated values which are defined in other standards.]		
PSIP tables:			
0xC7	MASTER GUIDE TABLE (MGT)	0x1FFB	Sec.6.2
0xC8	TERRESTRIAL VIRTUAL CHANNEL TABLE (TVCT)	0x1FFB	Sec.6.3.1
0xC9	CABLE VIRTUAL CHANNEL TABLE (CVCT)	0x1FFB	Sec.6.3.2
0xCA	RATING REGION TABLE (RRT)	0x1FFB	Sec.6.4
0xCB	EVENT INFORMATION TABLE (EIT)	per MGT	Sec.6.5
0xCC	EXTENDED TEXT TABLE (ETT)	per MGT	Sec.6.6
0xCD	SYSTEM TIME TABLE (STT)	0x1FFB	Sec.6.1
0xCE-0xD2	[ATSC coordinated values which are defined in other standards.]		
0xD3	DIRECTED CHANNEL CHANGE TABLE (DCCT)	0x1FFB	Sec.6.7
0xD4	DIRECTED CHANNEL CHANGE SELECTION CODE TABLE (DCCSCT)	0x1FFB	Sec.6.8
0xD5-0xDF	[ATSC coordinated values which are defined in other standards.]		
0xE0-0xE5	[Used in other systems]		
0xE6-0xFE	[Reserved for future ATSC use]		

section_syntax_indicator – This 1-bit field shall be set to '1' to always indicate the "long" form of the MPEG-2 private_section table.

private_indicator – This 1-bit field shall be set to '1'.

private_section_length – A 12-bit field. It specifies the number of remaining bytes in the private section immediately following this field up to and including the CRC_32 field. The value in this field shall not exceed 4093 (0xFFD).

table_id_extension – This is a 16-bit field and is table-dependent. It shall be considered to be logically part of the table_id field providing the scope for the remaining fields.

version_number – This 5-bit field is the version number of the PSIP_section. The version_number shall be incremented by 1 modulo 32 when a change in the information carried within the PSIP_section occurs. When the current_next_indicator is set to '0', then the version_number shall be that of the next applicable PSIP_section with the same table_id, table_id_extension, and section_number.

current_next_indicator – A 1-bit field, which when set to '1' indicates that the PSIP_section sent is currently applicable. When the current_next_indicator is set to '1', then the version_number shall be that of the currently applicable PSIP_section. When the bit is set to '0', it indicates that the PSIP_section sent is not yet applicable and shall be the next PSIP_section with the same section_number, table_id_extension, and table_id to become valid.

section_number – This 8-bit field gives the number of the PSIP_section. The section_number of the first section in a PSIP table shall be 0x00. The section_number shall be incremented by 1 with each additional section in PSIP table. The scope of the section_number shall be defined by the table_id and table_id_extension. That is, for each PSIP table and value of the table_id_extension field, there is the potential for the full range of section_number values.

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 PSIP table of which this section is a part. Its scope is the same as for the section_number field.

protocol_version – An 8-bit unsigned integer field whose function is 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 be used by a future version of this standard to indicate structurally different tables.

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 ISO/IEC 13818-1, Annex A after processing the entire private section.

Note that individual PSIP tables defined in this Standard may further constrain the above fields. But when not further constrained in each table, the above semantics shall apply.

4.2 Extensibility

The PSIP protocol describes a number of tables conveying system information and content guide data structures. The Standard is designed to be extensible via the following mechanisms:

Reserved fields. Fields in this Standard marked reserved shall be reserved for use either when revising this Standard, or when another standard is issued that builds upon this one. See Section 4.4.

Standard table types. As indicated in Table 4.2, table_id values in the range 0xCE-0xDF and 0xE6-0xFE shall be reserved for use either when revising this PSIP Standard, or when another standard is issued.

Protocol version field. Future substantive structural modifications shall be accommodated by defining different protocol version numbers. Backwards compatible additions or modifications may be added without incrementing the protocol version number.

4.3 User Private Ranges

Certain fields in this Standard are defined to include “user private” ranges:

- table_ID values in the range 0x40 through 0xBF
- MGT table_type values in the range 0x0400 through 0x0FFF

Table sections with table_ID values in the user private range (0x40 through 0xBF) shall not appear in transport packets identified with the base_PID PID value (0x1FFB). The MGT may refer to private table sections with any value of table_ID including values in the user private range. Refer to Section 6.2 for a discussion of the use of the MPEG-2 Registration Descriptor in the MGT with respect to delivery of private table sections.

4.4 Reserved Fields

reserved — Fields in this PSIP Standard marked "reserved" shall not be assigned by the user, but shall be available for future use. Decoders are expected to disregard reserved fields for which no definition exists that is known to that unit. Each bit in the fields marked "reserved" shall be set to one until such time as it is defined and supported.

5. TABLE HIERARCHY AND STRUCTURE REQUIREMENTS

The Program and System Information Protocol (PSIP) is a collection of hierarchically arranged tables for describing system information and program guide data. These tables are packetized and multiplexed according to the transport protocol detailed in ISO/IEC 13818-1.

The base PID (base_PID) is an explicitly defined value (0x1FFB) used to identify the packets for the following tables for terrestrial and cable systems:

- The System Time Table (STT)
- Master Guide Table (MGT)
- Rating Region Table (RRT)
- Virtual Channel Table (VCT)
- The optional Directed Channel Change Table (DCCT)
- The optional Directed Channel Change Selection Code Table (DCCSCT)

Several Event Information Tables (EIT) are also part of the PSIP data structures, with their PIDs explicitly defined in the MGT. Figure 5.1 illustrates the relations between these elements.

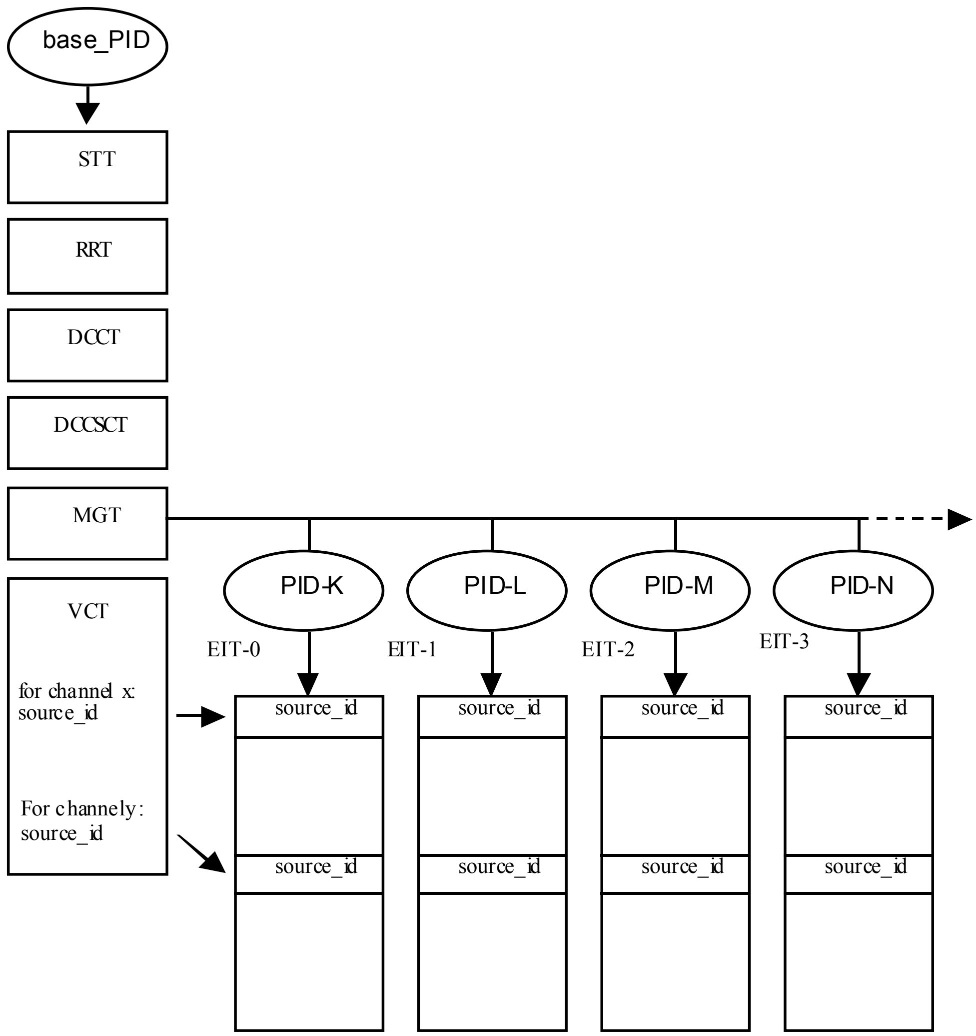


Figure 5.1 Table hierarchy for the Program and System Information Protocol (PSIP).

As the name indicates, the System Time Table (STT) carries time information needed for any application requiring synchronization. The Rating Region Table (RRT) defines rating tables valid for different regions or countries. The Master Guide Table (MGT) defines sizes, PIDs, and version numbers for all of the relevant tables. The Virtual Channel Table (VCT) actually exists in two versions: one for terrestrial and a second one for cable applications. Its purpose is to tabulate virtual channel attributes required for navigation and tuning. The terrestrial and cable versions are similar in structure, with the latter redefining the semantics of some fields pertinent to cable operations. The optional Directed Channel Change Table carries requests for a receiver to switch to specified virtual channels at specified times under specified circumstances. The optional Directed Channel Change Selection Code Table permits extension of the basic genre category and location code tables defined here.

Each of the Event Information Tables (EITs) lists TV programs (events) for the virtual channels described in the VCT. The EITs are sequentially and chronologically organized from EIT-0 to EIT-127. The first table (EIT-0) corresponds to the currently valid list of events. The second table (EIT-1) corresponds to the next time window, and so on.

During remultiplexing, EIT tables which originally existed in separate Transport Streams may be multiplexed into a common Transport Stream or *vice versa*. For this reason, it is very convenient to synchronize the start times and durations of the EITs. Consequently, the next three synchronization rules shall be followed when EIT tables are prepared.

- **Requirement 1:** Each EIT shall have a duration of 3 hours.
- **Requirement 2:** Start times for EITs are restricted to 0:00 (midnight), 3:00, 6:00, 9:00, 12:00 (noon), 15:00, 18:00 and 21:00. All of these times are UTC.
- **Requirement 3:** EIT-0 lists all of the available events for the current 3-hour time segment. EIT-1 lists all of the available events for the next 3-hour time segment, and likewise, non-overlapping sequential time windows are allocated for all of the other EITs.

For example, a broadcast group operating in the Eastern time zone of the U.S. at 15:30 EDT (19:30 UTC) is required to carry EIT-0 describing events from 14:00 to 17:00 EDT (18:00 to 21:00 in UTC time) plus EIT-1, EIT-2, and EIT-3 covering the next 9-hour interval between 17:00 to 2:00 EDT. At 17:00 EDT, the first table, EIT-0, will be obsolete while EIT-1 will still be valid. At this time, simply by shifting the listed PID values in the MGT, EIT-1 becomes EIT-0 and EIT-2 becomes EIT-1. Updating tables then becomes a process of shifting the list of PIDs in the MGT and their corresponding version numbers. However, updates and/or corrections to the information in the EITs may be performed at any time since the decoder monitors the MGT continuously, where the most current copy of the version number is maintained. Updates and/or corrections to the EIT (other than shifting) shall be signaled by increasing the version number by one.

Besides listing the PIDs for all of the EITs, the Master Guide Table (MGT) also lists a set of PIDs for Extended Text Tables (ETTs). The ETTs carry relatively long text messages for describing events and virtual channels. Each EIT has either zero or one associated ETT. Similarly, The VCT has either zero or one associated ETT. Figure 5.2 illustrates the concept.

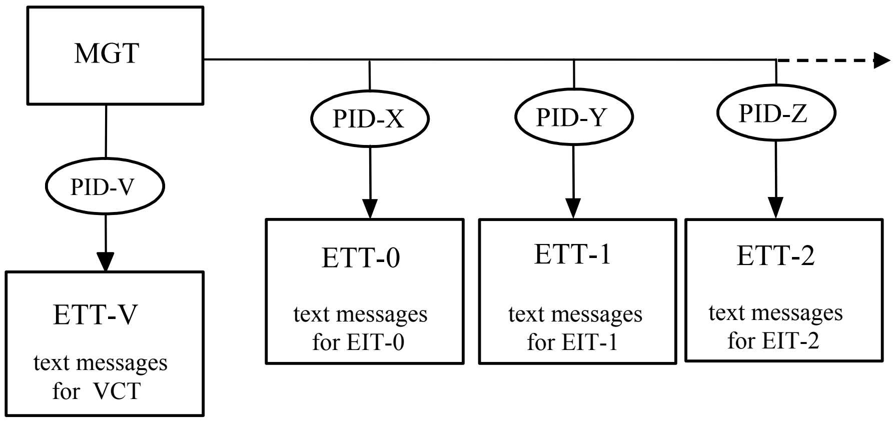


Figure 5.2 Extended Text Tables (ETTs) defined to carry text messages for describing virtual channels and events.

5.1 Requirements for Terrestrial Broadcast

The rules governing the transport of PSIP tables for terrestrial broadcast are:

- **Requirement 4:** Every digital Transport Stream in terrestrial broadcast shall include the STT, the TVCT, the MGT, and the first four Event Information Tables (EIT-0, EIT-1, EIT-2 and EIT-3)[5]. The CVCT, all of the other EITs and the whole collection of ETTs may be present but are not required. An RRT defining the rating system for a given region shall be included in the TS if any content_advisory_descriptor in use refers to that region, unless that region has explicit standards that define the rating system and the meaning of the values in the content_advisory_descriptor.
- **Requirement 5:** The PSIP tables shall describe all of the digital channels multiplexed in the Transport Stream. For convenience, the tables may optionally include information about analog channels as well as other digital channels available in different Transport Streams.

5.2 Requirements for Cable

The rules governing the transport of PSIP tables for cable are:

- **Requirement 6:** The required tables for a cable system are: the STT,either the CVCT or the TVCT, and the MGT. For any region that makes use of the capability to change the RRT, that RRT shall be included in the TS if any content_advisory_descriptor in use refers to that region. An RRT defining the rating system for a given region shall be included in the TS if any content_advisory_descriptor in use refers to that region, unless that region has explicit standards that define the rating system and the meaning of the values in the content_advisory_descriptor.
- **Requirement 7:** The PSIP tables shall describe all of the digital channels multiplexed in the Transport Stream. For convenience, the tables may optionally

[5] Exception: test signals may or may not be included in EIT/ETT data.

include information about analog channels as well as other digital channels available in different Transport Streams.

6. SPECIFICATIONS

This chapter describes the bit stream syntax and semantics for the System Time Table (STT), Master Guide table (MGT), Virtual Channel Table (VCT), Rating Region Table (RRT), Event Information Table (EIT), Extended Text Table (ETT), the optional Directed Channel Change Table (DCCT), the optional Directed Channel Change Selection Code Table (DCCSCT), core descriptors, and the multiple string structure.

6.1 System Time Table (STT)

The System Time Table provides the current date and time of day information.

The following constraints apply to the Transport Stream packet carrying the STT:

- PID for STT shall have the value 0x1FFB (base_PID)
- transport_scrambling_control bits shall have the value ‘00’
- adaptation_field_control bits shall have the value ‘01’

The bit stream syntax for the System Time Table is shown in Table 6.1.

Table 6.1 Bit Stream Syntax for the System Time Table

Syntax	No. of Bits	Format
system_time_table_section () {		
table_id	8	0xCD
section_syntax_indicator	1	‘1’
private_indicator	1	‘1’
reserved	2	‘11’
section_length	12	uimsbf
table_id_extension	16	0x0000
reserved	2	‘11’
version_number	5	‘00000’
current_next_indicator	1	‘1’
section_number	8	0x00
last_section_number	8	0x00
protocol_version	8	uimsbf
system_time	32	uimsbf
GPS_UTC_offset	8	uimsbf
daylight_savings	16	uimsbf
for (I = 0;I< N;I++) {		
descriptor()		
}		
CRC_32	32	rpchof
}		

table_id — This is an 8-bit field, which shall be set to 0xCD, identifying this table as the System Time Table.

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 1021.

table_id_extension — This 16-bit field shall be set to 0x0000.

version_number — This 5-bit field shall have a value of zero.

current_next_indicator — This 1-bit indicator is always set to '1' for an STT section; the STT 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 is 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 be used by a future version of this standard to indicate structurally different tables.

system_time — A 32-bit unsigned integer quantity representing the current system time as the number of GPS seconds since 00:00:00 UTC, January 6^{th}, 1980. The count of GPS seconds and leap second count shall be accurate and correct to within plus or minus one second, for a direct main broadcast signal RF receiving device, as timed at the arrival in the decoder of the Transport Stream packet carrying the last byte of the CRC. The STT seconds count should be set to the next second and sent approximately 2T milliseconds before the seconds count is due to increment, where T represents the average number of milliseconds between TS packets identified with the SI base_PID (0x1FFB). If one or more translators and/or repeaters are in the RF delivery path that introduce processing delays that impact the overall STT timing accuracy, the STT timing should be adjusted in the translated/repeated signal.

GPS_UTC_offset — 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.

daylight_savings — Daylight Savings Time Control bytes. Refer to Annex A for the use of these two bytes.

descriptor() — Zero or more descriptors, as appropriate, may be included.

CRC_32 — This is a 32-bit field that contains the CRC value that ensures a zero output from the registers in the decoder defined in Annex A of ISO/IEC 13818-1 "MPEG-2 Systems" after processing the entire System Time Table section.

6.2 Master Guide Table (MGT)

The MGT lists version numbers, length in bytes, and PIDs for all of the PSIP tables with the exception of the STT which works independently from the other tables.

The Master Guide Table is carried in a single section with table ID 0xC7, and obeys the syntax and semantics given in Section 4. The following constraints apply to the Transport Stream packet (or packets) carrying the MGT:

- PID for MGT shall have the value 0x1FFB (base_PID)
- transport_scrambling_control bits shall have the value '00'
- adaptation_field_control bits shall have the value '01'
- payload_unit_start_indicator of the Transport Stream packet carrying the table_id field of the MGT section shall be 1 (first Transport Stream packet of the section)
- pointer_field of the Transport Stream packet carrying the table_id field of the MGT section shall have the value 0x00 (section starts immediately after the pointer_field)

The MGT provides the version_number field of tables it references. For purposes of duplicate detection, the version_number shall be processed in accordance with the MPEG-2 definition of the scope of that field. If the version_number of a table transported in transport packets with a given table_type_PID value changes, that table may be assumed to be changed.

For the EIT and ETT tables, the table_type_version_number given in the MGT must not be interpreted as the table version for the associated timeslot. Refer to Annex D Section 8 for a discussion of this important distinction.

The MGT may refer to private table sections: those with the table_type field set to a value in the user private range (0x0400 through 0x0FFF). The table_type is the only semantic element in the MGT that has a user private range. When a table_type in the private range is present, an MPEG-2 registration_descriptor() (MRD) in accordance with Section 2.6.8 of [9] shall be present to identify the entity defining the private table type. Usage rules for the MRD are specified in Section 6.2.1 through 6.2.3. User private tables shall not be placed into transport packets with the PID value equal to the SI base_PID (0x1FFB).

The bit stream syntax for the Master Guide Table is shown in Table 6.2.

Table 6.2 Bit Stream Syntax for the Master Guide Table

Syntax	No. of Bits	Format
master_guide_table_section () {		
table_id	8	0xC7
section_syntax_indicator	1	'1'
private_indicator	1	'1'
reserved	2	'11'
section_length	12	uimsbf
table_id_extension	16	0x0000
reserved	2	'11'
version_number	5	uimsbf
current_next_indicator	1	'1'
section_number	8	0x00
last_section_number	8	0x00
protocol_version	8	uimsbf
tables_defined	16	uimsbf
for (i=0;i<tables_defined;i++) {		
table_type	16	uimsbf
reserved	3	'111'
table_type_PID	13	uimsbf
reserved	3	'111'
table_type_version_number	5	uimsbf
number_bytes	32	uimsbf
reserved	4	'1111'
table_type_descriptors_length	12	uimsbf
for (k=0;k<N;k++) {		
descriptor()		
}		
}		
reserved	4	'1111'
descriptors_length	12	uimsbf
for (I = 0;I< N;I++) {		
descriptor()		
}		
CRC_32	32	rpchof
}		

table_id — This is an 8-bit field which shall be set to 0xC7, identifying this table as the Master Guide Table.

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 4093.

table_id_extension — This 16-bit field shall be set to 0x0000.

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 be used by a future version of this standard to indicate structurally different tables.

tables_defined — This 16-bit unsigned has a range of 6 – 370 (for terrestrial) and 2 – 370 for cable.

table_type — This 16-bit unsigned integer specifies the type of table, based on Table 6.3.

Table 6.3 Table Types

table_type	Meaning
0x0000	Terrestrial VCT with current_next_indicator=1
0x0001	Terrestrial VCT with current_next_indicator=0
0x0002	Cable VCT with current_next_indicator=1
0x0003	Cable VCT with current_next_indicator=0
0x0004	Channel ETT
0x0005	DCCSCT
0x0006-0x00FF	[Reserved for future ATSC use]
0x0100-0x017F	EIT-0 to EIT-127
0x0180-0x01FF	[Reserved for future ATSC use]
0x0200-0x027F	Event ETT-0 to event ETT-127
0x0280-0x0300	[Reserved for future ATSC use]
0x0301-0x03FF	RRT with rating_region 1-255
0x0400-0x0FFF	[User private]
0x1000-0x13FF	[Reserved for future ATSC use]
0x1400-0x14FF	DCCT with dcc_id 0x00 – 0xFF
0x1500-0xFFFF	[Reserved for future ATSC use]

table_type_PID — This 13-bit field specifies the PID for the table_type described in the loop.

table_type_version_number — This 5-bit field reflects the version number of the table_type described in the loop. The value of this field shall be the same as the version_number entered in the corresponding fields of tables and table instances. For example, the value of this field for EIT-3 will be the same as that of the version_number that appears in the actual EIT-3. The version number for the next VCT (current_next_indicator = 0) shall be one unit more (modulo 32) than the version number for the current VCT (current_next_indicator = 1).

number_bytes — This 32-bit unsigned integer field indicates the total number of bytes used for the table_type described in the loop.

table_type_descriptors_length — Total length of the descriptors for the table_type described in the loop (in bytes).

descriptor() — Zero or more descriptors, as appropriate, may be included.

descriptors_length — Total length of the MGT descriptor list that follows (in bytes).

CRC_32 — This is a 32-bit field that contains the CRC value that ensures a zero output from the registers in the decoder defined in Annex A of ISO/IEC 13818-1 "MPEG-2 Systems" after processing the entire Master Guide Table section.

6.2.1 Descriptors in the MGT Outer Loop

An MPEG-2 registration_descriptor() in accordance with Section 2.6.8 of [9] may be placed in the descriptors_length "for" loop in the MGT (the "for" loop following the descriptors_length field). When used in this location, the scope of the MRD is all the user private table types referenced in the MGT. At most one MRD shall appear in the descriptors_length "for" loop.

6.2.2 Descriptors in the MGT Inner Loop

MPEG-2 registration_descriptor()s may be placed in the table_type_descriptors_length "for" loop of the MGT (the "for" loop following the table_type_descriptors_length field). When used in this location, the scope of the MRD is the individual table type being described in that iteration of the table_type_descriptors_length "for" loop. At most one MRD shall appear in any table_type_descriptors_length "for" loop.

6.2.3 Descriptor Precedence

When MRDs are found at both levels (descriptors_length "for" loop and table_type_descriptors_length "for" loop), then the MRD at the table_type_descriptors_length "for" loop shall further refine the meaning of the identification provided at the descriptors_length "for" loop level (i.e., assume the characteristics set by the higher level MRD and add additional characteristics).

6.3 Virtual Channel Table (VCT)

The Virtual Channel Table (VCT) contains a list of attributes for virtual channels carried in the Transport Stream. Any changes in the virtual channel structure shall be conveyed with a new version number. The basic information contained in the VCT table body includes Transport Stream ID, channel number (major and minor), short channel name, program number, access controlled flag, location field for extended text messages, and service type. Additional information may be carried by descriptors which may be placed in the descriptor loop after the basic information.

The Virtual Channel Table may be segmented into as many as 256 sections. One section may contain information for several virtual channels, but the information for one virtual channel shall not be segmented and put into two or more sections. Thus for each section, the first field after protocol_version shall be num_channels_in_section.

Each virtual channel is associated with a program_number. Every program element associated with that program_number shall be considered to be a part of that virtual channel.

6.3.1 Terrestrial Virtual Channel Table

The Terrestrial Virtual Channel Table is carried in private sections with table ID 0xC8, and obeys the syntax and semantics given in Section 4. The following constraints apply to the Transport Stream packets carrying the VCT sections:

- PID for Terrestrial VCT shall have the value 0x1FFB (base_PID)
- transport_scrambling_control bits shall have the value '00'
- adaptation_field_control bits shall have the value '01'

The bit stream syntax for the Terrestrial Virtual Channel Table is shown in Table 6.4.

Table 6.4 Bit Stream Syntax for the Terrestrial Virtual Channel Table

Syntax	No. of Bits	Format
terrestrial_virtual_channel_table_section () {		
table_id	8	0xC8
section_syntax_indicator	1	'1'
private_indicator	1	'1'
reserved	2	'11'
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	'11'
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
protocol_version	8	uimsbf
num_channels_in_section	8	uimsbf
for(i=0; i<num_channels_in_section; i++) {		
short_name	7*16	uimsbf
reserved	4	'1111'
major_channel_number	10	uimsbf
minor_channel_number	10	uimsbf
modulation_mode	8	uimsbf
carrier_frequency	32	uimsbf
channel_TSID	16	uimsbf
program_number	16	uimsbf
ETM_location	2	uimsbf
access_controlled	1	bslbf
hidden	1	bslbf
reserved	2	'11'
hide_guide	1	bslbf
reserved	3	'111'
service_type	6	uimsbf
source_id	16	uimsbf
reserved	6	'111111'
descriptors_length	10	uimsbf
for (i=0;i<N;i++) {		
descriptor()		
}		
}		
reserved	6	'111111'
additional_descriptors_length	10	uimsbf
for(j=0; j<N;j++) {		
additional_descriptor()		
}		
CRC_32	32	rpchof
}		

table_id — An 8-bit unsigned integer number that indicates the type of table section being defined here. For the terrestrial_virtual_channel_table_section(), the table_id shall be 0xC8.

section_syntax_indicator — The section_syntax_indicator is a one-bit field which shall be set to '1' for the terrestrial_virtual_channel_table_section().

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

section_length — This is a twelve 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 value in this field shall not exceed 1021.

transport_stream_id — The 16-bit MPEG-2 Transport Stream ID, as it appears in the Program Association Table (PAT) identified by a PID value of zero for this multiplex. The transport_stream_id distinguishes this Terrestrial Virtual Channel Table from others that may be broadcast in different PTCs.

version_number — This 5 bit field is the version number of the Virtual Channel Table. For the current VCT (current_next_indicator = 1), the version number shall be incremented by 1 whenever the definition of the current VCT changes. Upon reaching the value 31, it wraps around to 0. For the next VCT (current_next_indicator = 0), the version number shall be one unit more than that of the current VCT (also in modulo 32 arithmetic). In any case, the value of the version_number shall be identical to that of the corresponding entries in the MGT.

current_next_indicator — A one-bit indicator, which when set to '1' indicates that the Virtual Channel Table sent is currently applicable. When the bit is set to '0', it indicates that the table sent is not yet applicable and shall be the next table to become valid. This standard imposes no requirement that "next" tables (those with current_next_indicator set to '0') must be sent. An update to the currently applicable table shall be signaled by incrementing the version_number field.

> *Informative note:* It is inadvisable to deliver "next" tables far in advance of their use, because the complexity of the process necessary to make a change to either current *or* next is significantly increased. Transmission of a "next" table may be helpful when the table is so large it requires multiple sections for delivery. For tables small enough to fit into one section, use of "next" tables is deprecated (strongly discouraged).

section_number — This 8 bit field gives the number of this section. The section_number of the first section in the Terrestrial Virtual Channel Table shall be 0x00. It shall be incremented by one with each additional section in the Terrestrial Virtual Channel Table.

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 complete Terrestrial Virtual Channel Table.

protocol_version — An 8-bit unsigned integer field whose function is 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 be used by a future version of this standard to indicate structurally different tables.

num_channels_in_section — This 8 bit field specifies the number of virtual channels in this VCT section. The number is limited by the section length.

short_name — The name of the virtual channel, represented as a sequence of one to seven 16-bit code values interpreted in accordance with the UTF-16 representation of Unicode character data. If the length of the name requires fewer than seven 16-bit code values, this field shall be padded out to seven 16-bit code values using the Unicode NUL character (0x0000). Unicode character data shall conform to The Unicode Standard, Version 3.0 [18]. The UTF-16 representation of Unicode character data is in accordance with that defined by [18], which is identical to that defined by ISO/IEC 10646-1:2000 Annex C [7].

major_channel_number — A 10-bit number that represents the “major” channel number associated with the virtual channel being defined in this iteration of the “for” loop. Each virtual channel shall be associated with a major and a minor channel number. The major channel number, along with the minor channel number, act as the user’s reference number for the virtual channel. The major_channel_number shall be between 1 and 99. For major_channel_number assignments in the U.S., refer to Annex B.

minor_channel_number — A 10-bit number in the range 0 to 999 that represents the “minor” or “sub-“ channel number. This field, together with major_channel_number, performs as a two-part channel number, where minor_channel_number represents the second or right-hand part of the number. When the service_type is analog television, minor_channel_number shall be set to 0. Services whose service_type is either ATSC_digital_television or ATSC_audio_only shall use minor numbers between 1 and 99. For other types of services, such as data broadcasting, valid minor virtual channel numbers are between 1 and 999.

modulation_mode — An 8-bit unsigned integer number that indicates the modulation mode for the transmitted carrier associated with this virtual channel. Values of modulation_mode are defined by this standard in Table 6.5. For digital signals, the standard values for modulation mode (values below 0x80) indicate transport framing structure, channel coding, interleaving, channel modulation, forward error correction, symbol rate, and other transmission-related parameters, by means of a reference to an appropriate standard. The modulation_mode field shall be disregarded for inactive channels.

Table 6.5 Modulation Modes

modulation_mode	Meaning
0x00	[Reserved]
0x01	**Analog** — The virtual channel is modulated using standard analog methods for analog television.
0x02	**SCTE_mode_1** — The virtual channel has a symbol rate of 5.057 Msps, transmitted in accordance with *Digital Transmission Standard for Cable Television*, Ref. [11] (Mode 1). Typically, mode 1 will be used for 64-QAM.
0x03	**SCTE_mode_2** — The virtual channel has a symbol rate of 5.361 Msps, transmitted in accordance with *Digital Transmission Standard for Cable Television*, Ref. [11] (Mode 2). Typically, mode 2 will be used for 256-QAM.
0x04	**ATSC (8 VSB)** — The virtual channel uses the 8-VSB modulation method conforming to the *ATSC Digital Television Standard A/53B*. Ref. [2], Annex D.
0x05	**ATSC (16 VSB)** — The virtual channel uses the 16-VSB modulation method conforming to the *ATSC Digital Television Standard A/53B,* Ref. [2], Annex D.
0x06-0x7F	[Reserved for future use by ATSC]
0x80-0xFF	[User Private]

carrier_frequency — The recommended value for these 32 bits is zero. Use of this field to identify carrier frequency is allowed, but is deprecated. After January 1, 2010, these 32 bits shall be set to zero.

> *Informative note:* The receiver is expected to record the TSID value as indicated in the PAT of each digital Transport Stream and make note of the frequency used to tune it. In the case of an analog waveform, the analog TSID value carried in the VBI is recorded. The value of channel_TSID given in the VCT is matched against TSID values to form the association between PSIP data and transmitted content.

channel_TSID — A 16-bit unsigned integer field in the range 0x0000 to 0xFFFF that represents the MPEG-2 Transport Stream ID associated with the Transport Stream carrying the MPEG-2 program referenced by this virtual channel[6]. For inactive channels, channel_TSID shall represent the ID of the Transport Stream that will carry the service when it becomes active. The receiver is expected to use the channel_TSID to verify that any received Transport Stream is actually the desired multiplex. For analog channels (service_type 0x01), channel_TSID shall indicate the value of the analog TSID included in the VBI of the NTSC signal. Refer to Annex D Section 9 for a discussion on use of the analog TSID.

program_number — A 16-bit unsigned integer number that associates the virtual channel being defined here with the MPEG-2 PROGRAM ASSOCIATION and TS PROGRAM MAP tables. For virtual channels representing analog services, a value of 0xFFFF shall be specified for program_number. 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.

ETM_location — This 2-bit field specifies the existence and the location of an Extended Text Message (ETM), based on Table 6.6.

Table 6.6 ETM Location

ETM_location	Meaning
0x00	No ETM
0x01	ETM located in the PTC carrying this PSIP
0x02	ETM located in the PTC specified by the channel_TSID
0x03	[Reserved for future ATSC use]

access_controlled — A 1-bit Boolean flag that indicates, when set, that the events associated with this virtual channel may be access controlled. When the flag is set to 0, event access is not restricted.

hidden — A 1-bit Boolean flag that indicates, when set, that the virtual channel is not accessed by the user by direct entry of the virtual channel number. Hidden virtual channels are skipped when the user is channel surfing, and appear as if undefined, if accessed by direct channel

[6] *Informative note*: A registration authority for each region assigns TSID values, for both analog and digital signals. Contact ATSC for the name of the registration authority applicable to a specific region of interest.

entry. Typical applications for hidden channels are test signals and NVOD services. Whether a hidden channel and its events may appear in EPG displays depends on the state of the hide_guide bit.

hide_guide — A Boolean flag that indicates, when set to 0 for a hidden channel, that the virtual channel and its events may appear in EPG displays. This bit shall be ignored for channels which do not have the hidden bit set, so that non-hidden channels and their events may always be included in EPG displays regardless of the state of the hide_guide bit. Typical applications for hidden channels with the hide_guide bit set to 1 are test signals and services accessible through application-level pointers.

service_type — A 6-bit enumerated type field that shall identify the type of service carried in this virtual channel, based on Table 6.7.

Table 6.7 Service Types

service_type	Meaning
0x00	[Reserved]
0x01	analog_television — The virtual channel carries analog television programming
0x02	ATSC_digital_television — The virtual channel carries television programming (audio, video and optional associated data) conforming to ATSC standards
0x03	ATSC_audio — The virtual channel carries audio programming (audio service and optional associated data) conforming to ATSC standards.
0x04	ATSC_data_only_service — The virtual channel carries a data service conforming to ATSC standards, but no video of stream_type 0x02 or audio of stream_type 0x81.
0x05-0x3F	[Reserved for future ATSC use]

source_id — A 16-bit unsigned integer number that identifies the programming source associated with the virtual channel. In this context, a source is one specific source of video, text, data, or audio programming. Source ID value zero is reserved. Source ID values in the range 0x0001 to 0x0FFF shall be unique within the Transport Stream that carries the VCT, while values 0x1000 to 0xFFFF shall be unique at the regional level. Values for source_ids 0x1000 and above shall be issued and administered by a Registration Authority designated by the ATSC.

descriptors_length — Total length (in bytes) of the descriptors for this virtual channel that follows.

descriptor() — Zero or more descriptors, as appropriate, may be included.

additional_descriptors_length — Total length (in bytes) of the VCT descriptor list that follows.

CRC_32 — This is a 32-bit field that contains the CRC value that ensures a zero output from the registers in the decoder defined in Annex A of ISO/IEC 13818-1 "MPEG-2 Systems" after processing the entire Terrestrial Virtual Channel Table section.

An *inactive channel* is defined as a channel that has program guide data available, but the channel is not currently on the air. Inactive channels are represented as hidden channels with the hide_guide bit set to 0. The Transport Stream shall not carry a Program Map Table representing an inactive channel.

For inactive channels, the short_name, major_channel_number, and minor_channel_number fields reflect the name and channel number of the inactive channel, and may be used in construction of the program guide. The source_id for inactive channels is used, as it is for active channels, to link the virtual channel to the program guide data. The ETM_location indicates, as it does for active channels, the location of text related to the virtual channel. The service_type field and attribute flag access_controlled reflect the characteristics of the channel that will be valid when it is active.

6.3.2 Cable Virtual Channel Table

The Cable Virtual Channel Table is carried in private sections with table ID 0xC9, and obeys the syntax and semantics given in Section 4. The following constraints apply to the Transport Stream packets carrying the VCT sections:

- PID for Cable VCT shall have the value 0x1FFB (base_PID)
- transport_scrambling_control bits shall have the value '00'
- adaptation_field_control bits shall have the value '01'

The Cable Virtual Channel Table may be present in a terrestrial broadcast multiplex when a broadcaster has coordinated consistent channel labeling/numbering with all local cable operators carrying that multiplex, and different channel labeling and/or numbering between cable and terrestrial broadcast is desired. When both CVCT and TVCT are present in the multiplex, receiving devices are expected to use the TVCT to navigate services received via terrestrial broadcast and the CVCT to navigate services received via cable.

The bit stream syntax for the Cable Virtual Channel Table is shown in Table 6.8. The semantics for the CVCT are the same as the TVCT except for those fields explicitly defined below.

table_id — An 8-bit unsigned integer number that indicates the type of table section being defined here. For the cable_VCT_section, the table_id shall be 0xC9.

major_channel_number, **minor_channel_number** — These two 10-bit fields represent either a two-part or a one-part virtual channel number associated with the virtual channel being defined in this iteration of the "for" loop[7]. The one- or two-part number acts as the user's reference number for the virtual channel. Some channels in the CVCT may be represented with a one-part number while others are represented with two-part numbers.

To specify a two-part channel number, both the major_channel_number and the minor_channel_number fields shall be below 1000. To specify a one-part channel number, the six most significant bits of major_channel_number shall be '11 1111'. Values and combinations of major_channel_number and minor_channel_number falling outside these ranges are reserved. The one-part channel number is a 14-bit quantity that shall be computed by the following formula, represented in C syntax:

```
one_part_number = (major_channel_number & 0x00F) << 10 + minor_channel_number
```

[7] Note that these semantics permit these two forms of virtual channel numbers, in separate iterations of the "for" loop, to identify the same set of program elements.

Table 6.8 Bit Stream Syntax for the Cable Virtual Channel Table

Syntax	No. of Bits	Format
cable_virtual_channel_table_section () {		
table_id	8	0xC9
section_syntax_indicator	1	‘1’
private_indicator	1	‘1’
reserved	2	‘11’
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	‘11’
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
protocol_version	8	uimsbf
num_channels_in_section	8	uimsbf
for(i=0; i<num_channels_in_section;i++) {		
short_name	7*16	uimsbf
reserved	4	‘1111’
major_channel_number	10	uimsbf
minor_channel_number	10	uimsbf
modulation mode	8	uimsbf
carrier_frequency	32	uimsbf
channel_TSID	16	uimsbf
program_number	16	uimsbf
ETM_location	2	uimsbf
access_controlled	1	bslbf
hidden	1	bslbf
path_select	1	bslbf
out_of_band	1	bslbf
hide_guide	1	bslbf
reserved	3	‘111’
service_type	6	uimsbf
source_id	16	uimsbf
reserved	6	‘111111’
descriptors_length	10	uimsbf
for (i=0;i<N;i++) {		
descriptor()		
}		
}		
reserved	6	‘111111’
additional_descriptors_length	10	uimsbf
for(j=0; j<N;j++) {		
additional_descriptor()		
}		
CRC_32	32	rpchof
}		

path_select — A 1-bit field that associates the virtual channel with a transmission path. For the cable transmission medium, path_select identifies which of two physical input cables carries the Transport Stream associated with this virtual channel. Table 6.9 defines path_select. When the channel is inactive, path_select shall reflect the characteristics of the channel that will be valid when it is again active.

Table 6.9 Path Select

path_select	Meaning
0	path 1
1	path 2

out_of_band — A Boolean flag that indicates, when set, that the virtual channel defined in this iteration of the "for" loop is carried on the cable on the out-of-band physical transmission channel. When clear, the virtual channel is carried within a tuned multiplex. When the channel is inactive, out_of_band shall reflect the characteristics of the channel that will be valid when it is again active.

source_id — A 16-bit unsigned integer number that identifies the programming source associated with the virtual channel. In this context, a source is one specific source of video, text, data, or audio programming. Source ID value zero is reserved to indicate that the programming source is not identified. Source ID values in the range 0x0001 to 0x0FFF shall be unique within the Transport Stream that carries the VCT, while values 0x1000 to 0xFFFF shall be unique at the regional level. Values for source_Ids 0x1000 and above shall be issued and administered by a Registration Authority designated by the ATSC.

6.4 Rating Region Table (RRT)

The Rating Region Table (RRT) carries rating information for multiple geographical regions. Each RRT instance, identified by rating_region (the 8 least significant bits of table_id_extension), conveys the rating system information for one specific region. The size of each RRT instance shall not be more than 1024 bytes (including section header and trailer), and it shall be carried by only one MPEG-2 private section.

An RRT defining the rating system for a given region shall be included in the TS if any content_advisory_descriptor in use refers to that region, except if the RRT corresponding to that rating_region is fully defined in standards for that region.

The following constraints apply to the Transport Stream packets carrying the RRT sections.

- PID shall have the value 0x1FFB (base_PID)
- transport_scrambling_control bits shall have the value '00'
- adaptation_field_control bits shall have the value '01'

The bit stream syntax for the Rating Region Table is shown in Table 6.10.

Table 6.10 Bit Stream Syntax for the Rating Region Table

Syntax	No. of Bits	Format
rating_region_table_section () {		
table_id	8	0xCA
section_syntax_indicator	1	‘1’
private_indicator	1	‘1’
reserved	2	‘11’
section_length	12	uimsbf
table_id_extension {		
reserved	8	0xFF
rating_region	8	uimsbf
}		
reserved	2	‘11’
version_number	5	uimsbf
current_next_indicator	1	‘1’
section_number	8	uimsbf
last_section_number	8	uimsbf
protocol_version	8	uimsbf
rating_region_name_length	8	uimsbf
rating_region_name_text()	var	
dimensions_defined	8	uimsbf
for(i=0; i<dimensions_defined;i++) {		
dimension_name_length	8	uimsbf
dimension_name_text()	var	
reserved	3	‘111’
graduated_scale	1	bslbf
values_defined	4	uimsbf
for (j=0;j<values_defined;j ++) {		
abbrev_rating_value_length	8	uimsbf
abbrev_rating_value_ text()	var	
rating_value_length	8	uimsbf
rating_value_ text()	var	
}		
}		
reserved	6	‘111111’
descriptors_length	10	uimsbf
for (i=0;i<N;i++) {		
descriptor()		
}		
CRC_32	32	rpchof
}		

table_id — This is an 8-bit field, which shall be set to 0xCA, identifying this table as the Rating Region Table (RRT).

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 1021.

rating_region — An 8-bit unsigned integer number that defines the rating region to be associated with the text in this rating_region_table_section(). The value of this field is the identifier of this rating region, and thus this field may be used by the other tables (e.g., MGT) for referring to a specific rating region table. Assignment of values for rating_region is the responsibility of the ATSC, which shall be the registration authority. Contact ATSC for current assignment of values for rating_region.

version_number — This 5-bit field is the version number of the Rating Region table identified by combination of the fields table_id and table_id_extension. The version number shall be incremented by 1 modulo 32 when any field in this instance of the Rating Region Table changes. The value of this field shall be the same as that of the corresponding entry in MGT.

current_next_indicator — This 1-bit indicator is always set to '1'.

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

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

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

rating_region_name_length — An 8-bit unsigned integer number that defines the total length (in bytes) of the rating_region_name_text() field to follow.

rating_region_name_text() — A data structure containing a multiple string structure which represents the rating region name, e.g., "U.S. (50 states + possessions)", associated with the value given by rating_region. Text strings are formatted according to the rules outlined in Section 6.10. The display string for the rating region name shall be limited to 32 characters or less.

dimensions_defined — This 8-bit field (1-255) specifies the number of dimensions defined in this rating_region_table_section().

dimension_name_length — An 8-bit unsigned integer number that defines the total length in bytes of the dimension_name_text() field to follow.

dimension_name_text() — A data structure containing a multiple string structure which represents the dimension name being described in the loop. One dimension in the U.S. rating region, for example, is used to describe the MPAA list. The dimension name for such a case may be defined as "MPAA". Text strings are formatted according to the rules outlined in Section 6.10. The dimension name display string shall be limited to 20 characters or less.

graduated_scale — This 1-bit flag indicates whether or not the rating values in this dimension represent a graduated scale; i.e., higher rating values represent increasing levels of rated content within the dimension. Value 1 means yes, while value 0 means no.

values_defined — This 4-bit field (1–15) specifies the number of values defined for this particular dimension.

abbrev_rating_value_length — An 8-bit unsigned integer number that defines the total length (in bytes) of the abbrev_rating_value_text() field to follow.

abbrev_rating_value_text() — A data structure containing a multiple string structure which represents the abbreviated name for one particular rating value. The abbreviated name for rating value 0 shall be set to a null string; i.e., "". Text strings are formatted according to the rules outlined in Section 6.10. The abbreviated value display string shall be limited to 8 characters or less.

rating_value_length — An 8-bit unsigned integer number that defines the total length (in bytes) of the rating_value_text() field to follow.

rating_value_text() — A data structure containing a multiple string structure which represents the full name for one particular rating value. The full name for rating value 0 shall be set to a null string; i.e., “”. Text strings are formatted according to the rules outlined in Section 6.10. The rating value display string shall be limited to 150 characters or less.

descriptors_length — Length (in bytes) of all of the descriptors that follow this field.

descriptor() — Zero or more descriptors, as appropriate, may be included.

CRC_32 — This is a 32-bit field that contains the CRC value that ensures a zero output from the registers in the decoder defined in Annex A of ISO/IEC 13818-1 “MPEG-2 Systems” after processing the entire Rating Region Table section.

6.5 Event Information Table (EIT)

The Event Information Table (EIT) contains information (titles, start times, etc.) for events on defined virtual channels. An event is, in most cases, a typical TV program. The EIT shall be used to provide information for virtual channels of service_type 0x01, 0x02, or 0x03 (Table 6.7 defines service_type). For data-only services (those identified with service_type 0x04 in the VCT), the EIT shall not be used. Up to 128 EITs may be transmitted and each of them is referred to as EIT-k, with k = 0, 1, … 127.

Each EIT-k can have multiple instances, each of which contains information for one virtual channel, and each of which is identified by the combination of table_id and source_id. Each EIT-k instance may be segmented into as many as 256 sections. One section may contain information for several events, but the information for one event shall not be segmented and put into two or more sections. Thus the first field after protocol_version for each section shall be num_events_in_section.

PSIP supports up to 128 EITs, each of which provides the event information for a certain time span. For terrestrial broadcast, at least the first four EITs shall be included in the Transport Stream. Any event programmed for a time interval that extends over one or more EITs shall be described in each of these EITs, with the same event_id. For instance, an event that starts at 17:30 UTC and lasts until 19:30 UTC will appear in two EITs with the same event_id, the EIT covering 15:00-18:00 (UTC) as well as the EIT covering 18:00-21:00 (UTC). For a particular virtual channel, an event_id identifies uniquely each of the events programmed for the 3-hour interval of an EIT.

Each virtual channel defined in the VCT shall have a corresponding instance of EIT-k, unless the virtual channel belongs to a group sharing the same source_id. Virtual channels sharing a source_id appear in applications such as NVOD. In such a case, the entire group will have a unique instance of EIT-k identified precisely by the source_id. If a virtual channel has no event in the time span covered by EIT-k, its corresponding EIT instance shall have only one section, and the field num_events_in_section shall be set to zero.

Events shall be in the order of their starting times, i.e., the start time of the first event shall be ahead of that of the second event, and the start time of the last event in section one shall be equal or less than that of the first event in section two with the equality holding only when both events are the same.

For NVOD services, event entries in the EIT correspond to events scheduled in the virtual channel that carry the time_shifted_descriptor (the reference virtual channel). However, an NVOD event shall be listed in applicable EITs even when the NVOD event has finished in the reference virtual channel as long as the NVOD event remains on the air as a time shifted service in complementary virtual channels. Hence, an EIT may contain, in some cases, an expired event describing NVOD services.

The Event Information Table is carried in private sections with table ID 0xCB, and obeys the syntax and semantics given in Section 4. The following constraints apply to the Transport Stream packets carrying the EIT sections:

- PID for EIT-k shall have the same value as specified in the MGT, and shall be unique among the collection of table_type_PID values listed in the MGT.
- transport_scrambling_control bits shall have the value ‘00’.
- adaptation_field_control bits shall have the value ‘01’.

The bit stream syntax for the Event Information Table is shown in Table 6.11.

Table 6.11 Bit Stream Syntax for the Event Information Table

Syntax	No. of Bits	Format
event_information_table_section () {		
table_id	8	0xCB
section_syntax_indicator	1	‘1’
private_indicator	1	‘1’
reserved	2	‘11’
section_length	12	uimsbf
source_id	16	uimsbf
reserved	2	’11’
version_number	5	uimsbf
current_next_indicator	1	‘1’
section_number	8	uimsbf
last_section_number	8	uimsbf
protocol_version	8	uimsbf
num_events_in_section	8	uimsbf
for (j = 0; j< num_events_in_section;j++) {		
reserved	2	‘11’
event_id	14	uimsbf
start_time	32	uimsbf
reserved	2	‘11’
ETM_location	2	uimsbf
length_in_seconds	20	uimsbf
title_length	8	uimsbf
title_text()	var	
reserved	4	‘1111’
descriptors_length	12	
for (i=0;i<N;i++) {		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

table_id — This is an 8-bit field which shall be set to 0xCB, identifying this section as belonging to the Event Information Table.

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, including the CRC_32 field. The value of this field shall not exceed 4093.

source_id — This 16-bit field specifies the source_id of the virtual channel carrying the events described in this section.

version_number — This 5-bit field is the version number of EIT-i. The version number shall be incremented by 1 modulo 32 when any field in the EIT-i changes. Note that the version_number for EIT-i has no relation with that for EIT-j when j is not equal to i. The value of this field shall be identical to that of the corresponding entry in the MGT.

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

section_number — This 8-bit field gives the number of this section.

last_section_number — This 8-bit field specifies the number of the last section.

protocol_version — An 8-bit unsigned integer field whose function is 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 be used by a future version of this standard to indicate structurally different tables.

num_events_in_section — Indicates the number of events in this EIT section. Value 0 indicates no events defined in this section.

event_id — This field specifies the identification number of the event described. This number will serve as a part of the event ETM_id (identifier for event extended text message).

start_time — A 32-bit unsigned integer quantity representing the start time of this event as the number of GPS seconds since 00:00:00 UTC, January 6, 1980. In any virtual channel, the start_time value of an event shall not be less than the end time of the preceding event where the end time of an event is defined to be equal to that event's start_time value plus that event's length_in_seconds value.

ETM_location — This 2-bit field specifies the existence and the location of an Extended Text Message (ETM), based on Table 6.12.

Table 6.12 ETM Location

ETM_location	Meaning
0x0	No ETM
0x1	ETM located in the PTC carrying this PSIP
0x2	ETM located in the PTC carrying this event
0x3	[Reserved for future ATSC use]

length_in_seconds — Duration of this event in seconds.

title_length — This field specifies the length (in bytes) of the title_text(). Value 0 means that no title exists for this event.

title_text() — The event title in the format of a multiple string structure (see Section 6.10).

descriptors_length — Total length (in bytes) of the event descriptor list that follows.

descriptor() — Zero or more descriptors may be included in the EIT in an iteration of the event "for" loop. The types of descriptors defined for use in the EIT include the content_advisory_descriptor(), the caption_service_descriptor() and the AC-3 audio_stream_descriptor().

CRC_32 — This is a 32-bit field that contains the CRC value that ensures a zero output from the registers in the decoder defined in Annex A of ISO-13818-1 "MPEG-2 Systems" after processing the entire Event Information Table section.

6.6 Extended Text Table (ETT)

The Extended Text Table (ETT) contains Extended Text Message (ETM) streams, which are optional and are used to provide detailed descriptions of virtual channels (channel ETM) and events (event ETM). An ETM is a multiple string data structure (see Section 6.10), and thus, it may represent a description in several different languages (each string corresponding to one language). If necessary, the description may be truncated to fit allocated display space.

The Extended Text Message is carried in private sections with table ID 0xCC and obeys the syntax and semantics given in Section 4. Each description is distinguished by its unique 32-bit ETM_id immediately after the field protocol_version. This allows the receiver to search for a single description quickly without having to parse the payload of a large table.

The ETT section for a virtual channel or an event is carried in the home physical transmission channel (the physical transmission channel carrying that virtual channel or event) with PID specified by the field table_type_PID in corresponding entries in the MGT. This specific PID is exclusively reserved for the ETT stream.

The following constraints apply to the Transport Stream packets carrying the ETT sections.

- PID for ETT shall have the same value as the field table_type_PID in corresponding entries in the MGT, and shall be unique among the collection of table_type_PID values listed in the MGT.
- transport_scrambling_control bits shall have the value ‘00’
- adaptation_field_control bits shall have the value ‘01’

The bit stream syntax for the Extended Text Table is shown in Table 6.13.

Table 6.13 Bit Stream Syntax for the Extended Text Table

Syntax	No. of Bits	Format
extended_text_table_section () {		
table_id	8	0xCC
section_syntax_indicator	1	‘1’
private_indicator	1	‘1’
reserved	2	‘11’
section_length	12	uimsbf
ETT_table_id_extension	16	0x0000
reserved	2	‘11’
version_number	5	uimsbf
current_next_indicator	1	‘1’
section_number	8	0x00
last_section_number	8	0x00
protocol_version	8	uimsbf
ETM_id	32	uimsbf
extended_text_message ()	var	
CRC_32	32	rpchof
}		

table_id — Identifies this section as belonging to an Extended Text Table. (0xCC).

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 the 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 4093.

ETT_table_id_extension — A 16-bit unsigned integer value that serves to establish the uniqueness of each ETT instance when the tables appear in transport stream packets with common PID values. The ETT's table_ID_extension shall be set to a value such that separate ETT instances appearing in transport stream packets with common PID values have a unique ETT_table_ID_extension value.[8]

version_number — For the channel ETT, this 5-bit field indicates the version number of the channel ETT. The version number shall be incremented by 1 modulo 32 when any ETM in the channel ETT changes. For event ETT, this 5-bit field indicates the version number of event ETT-i, where i, as in the EIT case, is the index of time span. The version number shall be incremented by 1 modulo 32 when any ETM in the event ETT-i changes. Note that the version_number for event ETT-i has no relation with that for event ETT-j when j is not equal to i. The value of this field shall be identical to that of the corresponding entry in the MGT.

current_next_indicator — This 1-bit indicator is always set to '1' for ETT sections; the ETT 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 is 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 be used by a future version of this standard to indicate structurally different tables.

ETM_id — Unique 32-bit identifier of this extended text message. This identifier is assigned by the rule shown in Table 6.14.

Table 6.14 ETM ID

	MSB		LSB
Bit	31 … 16	15 … 2	'10'
channel ETM_id	source_id	0 … 0	'00'
event ETM_id	source_id	event_id	'10'

extended_text_message() — The extended text message in the format of a multiple string structure (see Section 6.10).

CRC_32 — This is a 32-bit field that contains the CRC value that ensures a zero output from the registers in the decoder defined in Annex A of ISO-13818-1 "MPEG-2 Systems" after processing the entire Transport Stream ETT section.

[8] Broadcasters in Korea are advised that some deployed receivers may not operate correctly with a non-zero value for this field.

6.7 Directed Channel Change Table (DCCT)

The optional Directed Channel Change Table provides definitions of virtual channel change requests. The table permits the broadcaster to indicate when the viewing experience can be enhanced by a change of virtual channels within or between physical channels. The requested channel change may be unconditional or may be based upon geographic, demographic, or categorical broadcast programming content selection criteria which may be specified and provided by the viewer to his/her "DCC capable DTV reference receiver[9]" (hereinafter DCCRR) through a menu setup type of procedure or through direct input. In the event that the viewer does not provide some of the Directed Channel Change Table setup selection criteria to the DCCRR, that portion of a DCC request shall be handled by the DCCRR in accordance with the specific rules defined in this specification. If Directed Channel Change is not supported by a DTV receiver there is no visible impact on the main broadcast program perceived by the viewer.

Several different DCCT instances may be present in the Transport Stream at any given time, each providing channel change information pertaining to one or more virtual channels.

Contained within the DCCT is a "for loop" structure that permits the ability to specify zero or more "tests" to be performed to determine whether or not a channel change may be effected. The tests, summarized in Table 6.17, may include requests such as determination if a viewer's DTV is located within a particular postal code region, whether the viewer is a member of a particular demographic group, or whether a program's content rating value results in a viewing block. Other tests are possible, as described within this standard. Additionally, different types of tests may be combined within one or more instances of a DCCT to allow logical "ORing" of channel change eligibility criteria.

A "DCC request" corresponds to one iteration of the dcc_test_count "for" loop. If evaluation of terms for a number of DCC requests given in one DCCT section indicates a true result for more than one request, the DCCRR is expected to take action on the first true request encountered. The term "DCC event" as used in this standard, shall refer to a channel change resulting from a DCC request.

The following constraints apply to the Transport Stream packet(s) carrying the DCCT:

- PID for DCCT shall have the value 0x1FFB (base_PID)
- transport_scrambling_control bits shall have the value '00'
- adaptation_field_control bits shall have the value '01'

The Directed Channel Change Table is carried in MPEG-2 private sections with table ID 0xD3, and obeys the syntax and semantics given in Section 4. The bit stream syntax for the Directed Channel Change Table section is shown in Table 6.15.

[9] Note: Receiver implementation is optional. For receiving devices that implement DCC functionality, the stated requirements for the DCCRR apply.

Table 6.15 Bit Stream Syntax for the Directed Channel Change Table

Syntax	No. of Bits	Format
directed_channel_change_table_section () {		
table_id	8	0xD3
section_syntax_indicator	1	'1'
private_indicator	1	'1'
reserved	2	'11'
section_length	12	uimsbf
dcc_subtype	8	0x00
dcc_id	8	uimsbf
reserved	2	'11'
version_number	5	uimsbf
current_next_indicator	1	'1'
section_number	8	0x00
last_section_number	8	0x00
protocol_version	8	uimsbf
dcc_test_count	8	uimsbf
for (i = 0; i < dcc_test_count; i++) {		
dcc_context	1	uimsbf
reserved	3	'111'
dcc_from_major_channel_number	10	uimsbf
dcc_from_minor_channel_number	10	uimsbf
reserved	4	'1111'
dcc_to_major_channel_number	10	uimsbf
dcc_to_minor_channel_number	10	uimsbf
dcc_start_time	32	uimsbf
dcc_end_time	32	uimsbf
dcc_term_count	8	uimsbf
for (j = 0; j < dcc_term_count; j++) {		
dcc_selection_type	8	uimsbf
dcc_selection_id	64	uimsbf
reserved	6	'111111'
dcc_term_descriptors_length	10	uimsbf
for (k = 0; k < N; k++) {		
dcc_term_descriptor()		
}		
}		
reserved	6	'111111'
dcc_test_descriptors_length	10	uimsbf
for (j = 0; j < N; j++) {		
dcc_test_descriptor()		
}		
}		
reserved	6	'111111'
dcc_additional_descriptors_length	10	uimsbf
for (i = 0; i < N; i++) {		
dcc_additional_descriptor()		
}		
CRC_32	32	rpchof
}		

table_id — This is an 8-bit field, which shall be set to 0xD3, identifying this table section as a Directed Channel Change Table section.

section_syntax_indicator — This 1-bit field shall be set to '1'. It denotes that the section follows the MPEG-2 long-form section syntax beyond the section length field.

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

section_length — A 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 this field shall not exceed 4093.

dcc_subtype — An 8-bit unsigned integer field that indicates the type of Directed Channel Change Table to follow. In the current specification only one type of DCC is defined, so this field shall be set to 0x00. Implementers are cautioned that dcc_subtype may be non-zero in a future version of this standard.

dcc_id — An 8-bit unsigned integer field that distinguishes different instances of transmitted DCC Table sections. The dcc_id shall be set so that no two currently active transmitted DCC Table sections are identified with the same value.

version_number — This 5-bit field is the version number of the DCC Table identified by the combination of fields dcc_subtype and dcc_id. The version number shall be incremented by 1 modulo 32 when any field in this instance of the DCC Table changes. In any case, the value of the version_number shall be identical to that of the corresponding entry in the MGT.

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

section_number — The value of this 8-bit field shall always be 0x00 (this table may be at most 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 is 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 0x00. Non-zero values of protocol_version may be used by a future version of this standard to indicate structurally different tables.

dcc_test_count — An 8-bit unsigned integer that specifies the number of channel change tests that will be defined by this DCC Table section. This outer loop associates a DCC request with each indicated virtual channel. A value of 0x00 for dcc_test_count indicates that the table section does not include channel change tests. A zero value may be used for a DCC Table section that carries descriptors in the dcc_additional_descriptors loop.

dcc_context — This 1-bit indicator indicates how a Directed Channel Change made in response to this test is to be handled by the DCCRR in the context of navigation and channel number display. Table 6.16 specifies requirements for operation in the two different specified modes. The dcc_context parameter identifies this DCC directive as being one of two types, either a Temporary Retune or a Channel Redirect.

"DCC From Channel Number" is defined to be the combination of dcc_from_major_channel_number and dcc_from_minor_channel_number. "DCC To Channel Number" is defined to be the combination of dcc_to_major_channel_number and dcc_to_minor_channel_number.

Table 6.16 DCC Context

dcc_context	Name and Function	Channel Number Displayed	New DCCs Accepted
0	**Temporary Retune**— acquire the virtual channel indicated in DCC To Channel Number and stay there until user changes channel, end time is reached, or DCC is canceled by a Return to Original Channel.	Original channel number: DCC From Channel Number	No (except to signal return to original channel)
1	**Channel Redirect**— tune to the virtual channel indicated in DCC To Channel Number	Actual channel number: DCC To Channel Number	Yes

Temporary Retune DCC

For a Temporary Retune DCC event, the displayed channel number shall stay constant (at DCC From Channel Number) to prevent confusion among viewers when it is desired to temporarily move them to an alternate program channel in a seamless manner. The DCCRR shall stay tuned to the DCC To Channel Number until one of the following events occurs:

1. The user manually changes channels; or
2. The dcc_end_time is reached; or
3. A DCCT is received containing a test in which:
   a. The DCC From Channel Number matches the original DCC From Channel Number; and
   b. The DCC To Channel Number matches the original DCC To Channel Number; and
   c. All the terms evaluate True; and
   d. One of the terms is Return to Original Channel (dcc_selection_type = 0x0F).

If the dcc_end_time is reached or a DCCT is received containing a test that matches the conditions in #3, the DCCRR shall immediately retune to the original channel number (DCC From Channel Number).

While tuned to the DCC To Channel Number in response to a Temporary Retune DCC event, the DCCRR shall not respond to DCC directives other than the type described in #3 above. If the user manually changes channels while so tuned, the state of the DCCRR shall be reset and the newly tuned channel shall be considered to be the "from" virtual channel.

The hide_guide bit and the hidden bit found within the VCT should both be set to 1 for the DCC To Channel Numbers in Temporary Retune DCC events to prevent those channels from appearing in EPG displays.

Channel Redirect DCC

Processing a Channel Redirect DCC event in the DCCRR simply involves a channel change—exactly as if it had been initiated manually by the viewer. Therefore the displayed channel number reflects the actual tuned virtual channel, the dcc_end_time in the DCC request shall not be processed, and the DCCRR shall be ready to process further DCC requests.

dcc_from_major_channel_number — A 10-bit number in the range of 1 to 999 that represents the "major" channel number, as defined in Section 6.3.1 Terrestrial Virtual Channel Table or 6.3.2 Cable Virtual Channel Table.

dcc_from_minor_channel_number — A 10-bit number in the range of 1 to 999 that represents the "minor" virtual channel number, as defined in Section 6.3.1 Terrestrial Virtual Channel Table or 6.3.2 Cable Virtual Channel Table.

The specified DCC To Channel Number shall correspond to a virtual channel currently defined in the VCT. If both a TVCT and a CVCT are present, the DCC To Channel Number shall be defined in each. The DCC To Channel Number fully identifies the virtual channel to which the DCCRR is requested to tune when the DCC request is in effect. The DCC To Channel Number shall be a major channel currently defined in the VCT and may have the "hidden" attribute.

dcc_to_major_channel_number — A 10-bit number in the range of 1 to 999 that represents the "major" channel number, (as defined in Section 6.3.1 Terrestrial Virtual Channel Table or 6.3.2 Cable Virtual Channel Table).

dcc_to_minor_channel_number — A 10-bit number in the range of 1 to 999 that represents the "minor" virtual channel number, (as defined in Section 6.3.1 Terrestrial Virtual Channel Table or 6.3.2 Cable Virtual Channel Table) of a virtual channel.

The DCC From Channel Number defined by the combination of these major and minor channel numbers shall identify a virtual channel that is currently defined in the VCT. This virtual channel may have the ''hidden" attribute set.

dcc_start_time — This field shall specify the nominal start time of a DCC request, expressed as the number of GPS seconds since 00:00:00 UTC, January 6, 1980. The optimum switch point for the start of a DCC event is indicated by splicing_point_flag and splice_countdown fields. The optimum switch point for the video ES shall be indicated by its splicing_point_flag and splice_countdown fields. The optimum switch point for each audio ES should be indicated by its splicing_point_flag and splice_countdown fields. The optimum switch point for all other Elementary Streams that are elements of the program may be indicated by their splicing_point_flag and splice_countdown fields.

The splicing_point_flag and splice_countdown fields shall be carried in the adaptation fields of the TS packets carrying the Elementary Streams, as defined in Sections 2.4.3.4 and 2.4.3.5 of MPEG-2 Systems [9]. The first TS packet transmitted containing splice point information for each Elementary Stream for a given DCC request shall indicate a splice_countdown value of not less than 5. The channel change should nominally occur when the DCCRR's internal time of day clock (as synchronized with the System Time Table received on the same Transport Stream) reaches dcc_start_time, but the DCCRR should use the splicing_point_flag and splice_countdown fields to effect the timing of the switchover. The switch point timing as signaled by the MPEG-2 splicing_point_flag if present shall be within one second (plus or minus) of the time indicated in dcc_start_time. If a splice point is not detected in an ES within one second following the dcc_start_time and a splice countdown is not in progress, the DCCRR shall perform the Elementary Stream switch without further delay. If a splice countdown is in progress in the ES, the DCCRR should perform the elementary stream switch at the splice point.

dcc_end_time — This field shall specify the nominal endpoint of the time interval during which the DCC request shall be in effect, expressed as the number of GPS seconds since 00:00:00 UTC, January 6, 1980. The optimum switch point for the end of a Temporary Retune DCC event is indicated by splicing_point_flag and splice_countdown fields. The optimum switch point for the video ES shall be indicated by its splicing_point_flag and splice_countdown fields. The optimum switch point for each audio ES should be indicated by its splicing_point_flag and splice_countdown fields. The optimum switch point for all other Elementary Streams that are elements of the program may be indicated by their splicing_point_flag and splice_countdown fields.

The splicing_point_flag and splice_countdown fields shall be carried in the adaptation fields of the TS packets carrying the Elementary Streams, as defined in Sections 2.4.3.4 and 2.4.3.5 of MPEG-2 Systems [9]. The first TS packet transmitted containing splice point information for each Elementary Stream for a given DCC request shall indicate a splice_countdown value of not less than 5. The channel change should nominally occur when the DCCRR's internal time of day clock (as synchronized with the System Time Table received on the same Transport Stream) reaches dcc_end_time, but the DCCRR should use the splicing_point_flag and splice_countdown fields to effect the timing of the switchover. The switch point timing as signaled by the MPEG-2 splicing_point_flag if present shall be within one second (plus or minus) of the time indicated in dcc_end_time. If a splice point is not detected in an ES within one second following the dcc_end_time and a splice countdown is not in progress, the DCCRR shall perform the elementary stream switch without further delay. If a splice countdown is in progress in the ES, the DCCRR should perform the Elementary Stream switch at the splice point.

dcc_term_count — This 8-bit unsigned integer specifies the number of dcc_selection_types and dcc_selection_ids to be associated with the DCC request. If the dcc_term_count is greater than one, the result of each dcc_selection_type in the "for" loop shall be evaluated as an intermediate term and then all intermediate terms for each dcc_selection_type logically ANDed together to determine the final result. If the final result is True (all terms evaluate True) a DCC channel change shall be indicated, otherwise no channel change shall be indicated. If it is desired to perform a combinatorial OR of individual dcc_selection_types then these types can be included in separate iterations of the dcc_test_count "for" loop. Alternatively, separate DCC Table transmissions may be sent.

A DCC request is eligible to be acted on whenever the DCCRR is tuned to the DCC From Channel Number, the current time is between the dcc_start_time and the dcc_end_time, and the result of evaluating and ANDing together all the terms in the dcc_term_count “for” loop is True. If one or more DCC requests in the loop are eligible to be acted on, the DCCRR shall act on the first eligible DCC request encountered in the loop. The action taken shall be tuning to the DCC To Channel Number.

dcc_selection_type — This 8-bit unsigned integer specifies the type of the value contained in the dcc_selection_id. Values for dcc_selection_type shall be as defined in Table 6.17.

Table 6.17 DCC Selection Type Assignments

dcc_selection_type	Name and Meaning	Value of DCC Selection ID	Test	Logic
* 0x00	**Unconditional** channel change	n.a.	Term always evaluates True.	True
* 0x01	**Numeric Postal Code Inclusion—** Inclusion test on numeric postal codes, with wild-card match on "?" characters	8 ASCII characters representing a specific or range of numeric character postal codes in the range 00000001 to 00099999. ASCII "?" matches any digit 0-9.	Term evaluates True if the DCCRR postal code matches, in the last five character positions, for those selection ID characters not equal to "?" and False otherwise. If postal code not specified in DCCRR, term evaluates False.	
0x02	**Alphanumeric Postal Code Inclusion—** Inclusion test on 8-character alphanumeric postal code, with wild-card match on "?" characters	8 ASCII characters representing an alphanumeric character postal code comprising 8 characters. ASCII "?" matches 0-9 or A-Z.	Term evaluates True if the DCCRR postal code matches, in all the character positions, for those selection ID characters not equal to "?" and False otherwise. If postal code not specified in DCCRR, term evaluates False.	
0x03-0x04	Reserved	Reserved		
0x05	**Demographic Category: one or more**— Test for membership in at least one indicated demographic category	A bit vector where each bit represents a demographic category	Term evaluates True if **any** of the selection ID bits correspond to a DCCRR membership demographic category and False otherwise.	(U & D & S) != 0
0x06	**Demographic Category: all**—Test for membership in all indicated demographic categories	A bit vector where each bit represents a demographic category	Term evaluates True if **all** of the selection ID bits correspond to DCCRR membership demographic categories and False otherwise.	(U & D & S) == D
0x07	**Genre Category: one or more**—Test for an interest in at least one genre category	Up to eight genre category codes	Term evaluates True if **any** of the selection ID category codes correspond to DCCRR interest categories and False otherwise.	(U & D & S) != 0 (see text)
0x08	**Genre Category: all**—Test for interest in all indicated genre categories	Up to eight genre category codes	Term evaluates True if **all** of the selection ID category codes correspond to DCCRR interest categories and False otherwise.	(U & D & S) == D (see text)

dcc_selection_type	Name and Meaning	Value of DCC Selection ID	Test	Logic
0x09	**Cannot Be Authorized—**A secondary redirect switch triggered upon detection of a failure to be authorized to remain on the requested "from" major/minor channel.	n.a.	Term evaluates True if the DCCRR cannot be authorized to decode services on the "from" channel and False otherwise.	
0x0A–0x0B	Reserved	n.a.		
0x0C	**Geographic Location Inclusion**	Value is a location_code conforming to the state_code, county_subdivision, and county_code.	Term evaluates True only if the DCCRR's geographic location matches the selection ID and False otherwise. If the geographic location data is not specified in the DCC, term evaluates False.	
0x0D	**Rating Blocked**—Test for rating blocked.	n.a.	Term evaluates True if the current program is blocked due to content, after a timeout (to allow user to override) and False otherwise.	
0x0E	Reserved.			
* 0x0F	**Return To Original Channel**	n.a.	Return unconditionally to previous Virtual Channel if engaged in a DCC request.	
0x10	Reserved.	Reserved.		
* 0x11	**Numeric Postal Code Exclusion—** Exclusion test on numeric postal codes, with wild-card match on "?" characters	8 ASCII characters representing a specific or range of numeric character postal codes in the range 00000001 to 00099999. ASCII "?" matches any digit 0-9.	Term evaluates True if the DCCRR postal code does **not** match, in the last five character positions, for those selection ID characters not equal to "?", and False otherwise. If postal code not specified, term evaluates False.	
0x12	**Alphanumeric Postal Code Exclusion—** Exclusion test on 8-character alphanumeric postal code, with wild-card match on "?" characters	8 ASCII characters representing an alphanumeric character postal code comprising 8 characters. ASCII "?" matches 0-9 or A-Z.	Term evaluates True if the DCCRR postal code does **not** match, for those selection ID characters not equal to "?" and False otherwise. If postal code not specified, term evaluates False.	
0x13-0x14	Reserved.	Reserved.		

dcc_selection_type	Name and Meaning	Value of DCC Selection ID	Test	Logic
0x15	**Demographic Category: one or more non-member**—Test for non-membership in at least one indicated demographic category	A bit vector where each bit represents a demographic category	Term evaluates True if **any** of the selection ID bits correspond to DCCRR **non**-membership in that demographic category and False otherwise.	(~U & D & S) != 0
0x16	**Demographic Category: all non-member**—Test for non-membership in all the indicated demographic categories	A bit vector where each bit represents a demographic category	Term evaluates True if **all** of the selection ID bits correspond to DCCRR **non**-membership demographic categories and False otherwise.	(~U & D & S) == D
0x17	**Genre Category: one or more non-member**—Test for non-interest in at least one indicated genre category	Up to eight genre category codes	Term evaluates True if **any** of the selection ID category codes correspond to genres of **no interest** in the DCCRR and False otherwise.	(~U & D & S) != 0 (see text)
0x18	**Genre Category: all non-member**—Test for non-interest in all the indicated genre categories	Up to eight genre category codes	Term evaluates True if **all** of the selection ID category codes correspond to genres of **no interest** in the DCCRR and False otherwise.	(~U & D & S) == D (see text)
0x19-0x1B	Reserved	Reserved		
0x1C	**Geographic Location Exclusion**	Value is a location_code conforming to the state_code, county_subdivision, and county_code.	Term evaluates True if the DCCRR's geographic location does **not** match the selection ID and False otherwise. If the geographic location data is not specified in the DCC, term evaluates False.	
0x1D-0x1F	Reserved	Reserved		
* 0x20-0x23	**Viewer-Direct-Select**— 0x20 corresponds to Button A; 0x21 corresponds to Button B; 0x22 corresponds to Button C; and 0x23 corresponds to Button D.	A 64-bit number associated with a given button choice; used in the VDS "persistence" function. See text.	Tune to the channel associated with the indicated function button if that button is selected. Term always evaluates True when viewer presses a Direct Select button.	
0x24-0xFF	Reserved.	Reserved.		

Table Legend and Notes

D = DCC Selection ID data: data sent within the dcc_selection_id field.

U = User-entered DCCRR data: data stored in non-volatile DCCRR memory that is entered by the user to establish the user's selected choices.

S = Specification mask indicating validity of U (i.e. user has entered a value for U): data stored in non-volatile DCCRR memory that indicates that a user has specified a choice for the associated data item.

NOTE: Items marked with an asterisk (*) above are required within a DTV device providing minimal support for Directed Channel Change within the United States.

Operators used within Table 6.17:

&	bitwise AND
==	equal to
~	1's complement (bitwise inversion)
!=	not equal to

dcc_selection_id — This 64-bit unsigned integer contains the data identified by the dcc_selection_type field, and is described below. Note: 8 bit characters specified for use within this section shall mean characters defined in ISO/IEC 8859-1 (ISO Latin-1) [8].

6.7.1 Case: Unconditional Channel Change (dcc_selection_type = 0x00)

If the dcc_selection_type is specified to be of type "unconditional" (0x00), the dcc_selection_id shall be 0x00 and the DCCRR shall unconditionally switch to the DCC To Channel Number if the current time is within the interval bounded by dcc_start_time and dcc_end_time.

6.7.2 Case: Numeric Postal Code (dcc_selection_type = 0x01, 0x11)

If the dcc_selection_type is specified to be of type 0x01 or 0x11, the dcc_selection_id shall consist of a right-justified five numeric 8-bit character postal code field in the range of 00001 to 99999 padded on the left with "0" (0x30) characters. The DCCRR shall compare that value to a stored representation of a numeric postal code entered by the user from setup menus within the DCCRR to determine if there is a match. If a question mark ("?" or 0x3F) character appears in any of the five least significant numeric character positions, that position shall be considered to be a wild card which will permit a selection on any numeric digit within that position. For example 00055?98 would permit matches on 00055098, 00055198, 00055298, 00055398, …, and 00055998. Similarly, 00055??8 would permit matches on 00055008, 00055018, 00055028, …, 00055108, 00055118, …, and 00055908, 00055918, …, and 00055998. Note that multiple numeric postal code specifications may be made within a single DCCT by means of the dcc_term_count loop.

If the dcc_selection_type is defined to be of type 0x01, the term shall evaluate True if the numeric postal code (with evaluated wild cards if any) given in the dcc_selection_id matches the stored numeric postal code entered by the user and False otherwise.

If the dcc_selection_type is defined to be of type 0x11, the term shall evaluate True if the numeric postal code (with evaluated wild cards if any) given in the dcc_selection_id does not match the stored numeric postal code entered by the user and False otherwise.

6.7.3 Case: Alphanumeric Postal Code (dcc_selection_type = 0x02, 0x12)

If the dcc_selection_type is specified to be of type 0x02 or 0x12, the dcc_selection_id shall consist of a right justified eight alphanumeric and special 8-bit character postal code field of unspecified format padded on the left with space characters (0x20). The field may also contain separator characters, as necessary, to format the postal code according to country conventions. The separator characters may consist of any of the following special characters: comma (0x2C), dash (0x2D), period (0x2E), slash (0x2F) or space (0x30). The separator characters shall be considered to be "do not care" placeholders for purposes of logical comparison to a postal code stored within the DCCRR. The DCCRR shall compare that value to a stored representation of a postal code entered by the user from setup menus within the DCCRR to determine if there is a match.

The alphanumeric and special characters permitted shall be any printing character within the character set from 0x20 through 0x7E inclusive. If a question mark ("?" or 0x3F) character appears in any of the eight character positions, that position shall be considered to be a wildcard which will permit a selection on any character within that position. For example "5B3-5Q?" would permit matches on 5B3-5Q0 through 5B3-5Q9 assuming the postal format convention was a numeric character in the rightmost character position. Similarly, “5B3-5?3” would permit matches on 5B3-5A3 through 5B3-5Z3 assuming the postal format convention for the second from the rightmost character is alphabetic. Note that multiple postal codes may be specified within a single DCCT by means of the dcc_term_count loop.

If the dcc_selection_type is defined to be of type 0x02, the term shall evaluate True if the alphanumeric postal code (with evaluated wild cards if any) given in the dcc_selection_id matches the stored alphanumeric postal code entered by the user and False otherwise.

If the dcc_selection_type is defined to be of type 0x12, the term shall evaluate True if the alphanumeric postal code (with evaluated wild cards if any) given in the dcc_selection_id does not match the stored alphanumeric postal code entered by the user and False otherwise.

6.7.4 Case: Demographic Category (dcc_selection_type = 0x05, 0x06, 0x15, 0x16)

If dcc_selection_type is equal to 0x05, 0x06, 0x15, or 0x16, the dcc_selection_id shall be specified to be a demographic selection bit field composed of one or more of the values described in Table 6.18 logically ORed together. The DCCRR shall perform a comparison, based upon the dcc_selection_type, of the value to a stored value within the DCCRR which had been entered by the user within setup menus to determine if there is a match.

Table 6.18 Demographic Selection Type Assignments

Value	Meaning
0x0000000000000001	Males
0x0000000000000002	Females
0x0000000000000004	Ages 2-5
0x0000000000000008	Ages 6-11
0x0000000000000010	Ages 12-17
0x0000000000000020	Ages 18-34
0x0000000000000040	Ages 35-49
0x0000000000000080	Ages 50-54
0x0000000000000100	Ages 55-64
0x0000000000000200	Ages 65+
0x0000000000000400	Working
0x0000000000000800 - 0x8000000000000000	Reserved

If the selection is specified to be of type "One-or-More Members" (dcc_selection_type 0x05), the received value within dcc_selection_id shall be logically bitwise ANDed with the DCCRR’s stored value and then logically bitwise ANDed with a specification mask that indicates whether the viewer had entered values for each demographic selection category. If the result is non-zero, the term shall evaluate True. If the result is zero, the term shall evaluate False. This test permits selection based upon membership in at least one and possibly more demographic categories. In other words, the term evaluates True if any of the categories that had been selected and validated by the viewer correspond with membership in the demographic categories specified in the dcc_selection_id field of the DCCT.

If the selection is specified to be of type "One-or-More Non-members" (dcc_selection_type 0x15), the received value within dcc_selection_id shall be logically bitwise ANDed with the 1's complement of the DCCRR's stored value and then logically bitwise ANDed with a specification mask that indicates whether the viewer had entered values for each demographic selection category. If the result is non-zero, the term shall evaluate True. If the result is zero, the term shall evaluate False. This test permits selection based upon non-membership in at least one and possibly more demographic categories. In other words, the term evaluates True if any of the categories that had been selected and validated by the viewer correspond with non-membership in the DCC demographic categories specified in the dcc_selection_id field of the DCCT.

If the selection is specified to be of type "All Members" (dcc_selection_type 0x06), the received value within dcc_selection_id shall be logically compared to the DCCRR's stored value which has been logically bitwise ANDed with a specification mask that indicates whether the viewer had entered values for each demographic selection category. If the result of the comparison is that the values are equal, the term shall evaluate True, and shall evaluate False otherwise. This test permits selection based upon membership in all requested demographic categories. In other words, the term evaluates True if the categories that had been selected and validated by the viewer correspond with membership in all of the DCC demographic categories specified in the dcc_selection_id field of the DCCT.

If the selection is specified to be of type "All Non-members" (dcc_selection_type 0x16), the received value within dcc_selection_id shall be logically bitwise ANDed with the 1's complement of the DCCRR's stored value and then logically bitwise ANDed with a specification mask that indicates whether the viewer had entered values for each demographic selection category. That result shall be compared to the DCCRR's stored value. If the result of the comparison is that the values are equal, the term shall evaluate True, and shall evaluate False otherwise. This test permits selection based upon non-membership in all requested demographic categories. In other words, the term evaluates True if the categories that had been selected and validated by the viewer correspond with non-membership in all of the DCC demographic categories specified in the dcc_selection_id field of the DCCT.

6.7.5 Case: Genre Category (dcc_selection_type = 0x07, 0x08, 0x17, 0x18)

If dcc_selection_type is equal to 0x07, 0x08, 0x17, or 0x18, the dcc_selection_id shall be specified to be a genre category selection code field. The DCCRR shall compare the code values obtained from the dcc_selection_id field to stored values which had been entered by the user through selection setup menus to determine if there is a match. Each occurrence of the dcc_selection_id may contain up to eight categorical selection codes, each code having a length of eight bits.

Up to a maximum of eight 8-bit Genre Category selection codes may be specified. Genre Category selection code bytes shall be placed right-justified in the 64-bit dcc_selection_id field. Each of the codes present shall consist of a value in the range 0x01 through 0xFF. If fewer than eight codes are specified, the remaining bytes in the dcc_selection_id field shall each contain 0x00. Table 6.19 illustrates the categorical selection criteria code placement within the dcc_selection_id field for four examples.

Table 6.19 Examples of Selection Code Packing

Value	Meaning
0x0000000000000000	no codes specified
0x0000000000222120	3 codes in least significant 24 bits
0x0000000052304120	4 codes in least significant 32 bits
0x3031323334353620	8 codes in 64 bits

If the selection is specified to be of type "One-or-More Members" (dcc_selection_type 0x07), each of the individual eight byte values received within dcc_selection_id shall be compared with data in the DCCRR's stored value tables. Each compare results in a "1" if the category corresponds to one of interest and "0" if not. That 8-bit result is ANDed with the S mask corresponding to whether choices have been registered for each of those categories. If the result is non-zero, the term shall evaluate True, and shall evaluate False otherwise. This test permits selection based upon interest membership in at least one and possibly more of the genre categories according to the Categorical Genre Code Assignment table (Table 6.20) including any downloaded extensions to that table. In other words, the term evaluates True if the genre categories that had been selected and validated by the viewer correspond with interest in any of the of the genre categories specified in the dcc_selection_id field of the DCCT.

Table 6.20 Categorical Genre Code Assignments

Value	Meaning	Value	Meaning	Value	Meaning
0x00	Not Available	0x4F	Hobby	0x80	Art
0x01-1F	Reserved (Basic)	0x50	Hockey	0x81	Auto Racing
0x20	Education	0x51	Home	0x82	Aviation
0x21	Entertainment	0x52	Horror	0x83	Biography
0x22	Movie	0x53	Information	0x84	Boating
0x23	News	0x54	Instruction	0x85	Bowling
0x24	Religious	0x55	International	0x86	Boxing
0x25	Sports	0x56	Interview	0x87	Cartoon
0x26	Other	0x57	Language	0x88	Children
0x27	Action	0x58	Legal	0x89	Classic Film
0x28	Advertisement	0x59	Live	0x8A	Community
0x29	Animated	0x5A	Local	0x8B	Computers
0x2A	Anthology	0x5B	Math	0x8C	Country Music
0x2B	Automobile	0x5C	Medical	0x8D	Court
0x2C	Awards	0x5D	Meeting	0x8E	Extreme Sports
0x2D	Baseball	0x5E	Military	0x8F	Family
0x2E	Basketball	0x5F	Miniseries	0x90	Financial
0x2F	Bulletin	0x60	Music	0x91	Gymnastics
0x30	Business	0x61	Mystery	0x92	Headlines
0x31	Classical	0x62	National	0x93	Horse Racing
0x32	College	0x63	Nature	0x94	Hunting/Fishing/Outdoors
0x33	Combat	0x64	Police	0x95	Independent
0x34	Comedy	0x65	Politics	0x96	Jazz
0x35	Commentary	0x66	Premier	0x97	Magazine
0x36	Concert	0x67	Prerecorded	0x98	Motorcycle Racing
0x37	Consumer	0x68	Product	0x99	Music/Film/Books
0x38	Contemporary	0x69	Professional	0x9A	News-International
0x39	Crime	0x6A	Public	0x9B	News-Local
0x3A	Dance	0x6B	Racing	0x9C	News-National
0x3B	Documentary	0x6C	Reading	0x9D	News-Regional
0x3C	Drama	0x6D	Repair	0x9E	Olympics
0x3D	Elementary	0x6E	Repeat	0x9F	Original
0x3E	Erotica	0x6F	Review	0xA0	Performing Arts
0x3F	Exercise	0x70	Romance	0xA1	Pets/Animals
0x40	Fantasy	0x71	Science	0xA2	Pop
0x41	Farm	0x72	Series	0xA3	Rock & Roll
0x42	Fashion	0x73	Service	0xA4	Sci-Fi
0x43	Fiction	0x74	Shopping	0xA5	Self Improvement
0x44	Food	0x75	Soap Opera	0xA6	Sitcom
0x45	Football	0x76	Special	0xA7	Skating
0x46	Foreign	0x77	Suspense	0xA8	Skiing
0x47	Fund Raiser	0x78	Talk	0xA9	Soccer
0x48	Game/Quiz	0x79	Technical	0xAA	Track/Field
0x49	Garden	0x7A	Tennis	0xAB	True
0x4A	Golf	0x7B	Travel	0xAC	Volleyball
0x4B	Government	0x7C	Variety	0xAD	Wrestling
0x4C	Health	0x7D	Video	0xAE-FE	Reserved (Detailed)
0x4D	High School	0x7E	Weather	0xFF	Null (not a category)
0x4E	History	0x7F	Western		

If the selection is specified to be of type "One-or-More Non-members" (dcc_selection_type 0x17), each of the individual eight byte values received within dcc_selection_id shall be compared with data in the DCCRR's stored value tables. Each compare results in a "1" if the category does not correspond to one of interest and "0" if it does. That 8-bit result is ANDed with the S mask corresponding to whether choices have been registered for each of those categories. If the result is non-zero, the term shall evaluate True, and shall evaluate False otherwise. This test permits selection based upon non-membership (meaning no interest) in at least one and possibly more specified genre categories. In other words, the term evaluates True if the categories that had been selected and validated by the viewer correspond with non-interest in any of the genre categories specified in the dcc_selection_id field of the DCCT.

If the selection is specified to be of type "All Members" (dcc_selection_type 0x08), each of the individual eight byte values received within dcc_selection_id shall be compared with data in the DCCRR's stored value tables. Each compare results in a "1" if the category corresponds to one of interest and "0" if not. That 8-bit result is compared with the S mask corresponding to whether choices have been registered for each of those categories. If the result is equal, the term shall evaluate True, and shall evaluate False otherwise. This test permits selection based upon interest in all specified genre categories. In other words, the term evaluates True if the categories that had been selected and validated by the viewer resulted in interest in all of the genre categories specified in the dcc_selection_id field of the DCCT.

If the selection is specified to be of type "All Non-members" (dcc_selection_type 0x18), each of the individual eight byte values received within dcc_selection_id shall be compared with data in the DCCRR's stored value tables. Each compare results in a "1" if the category does not correspond to one of interest and "0" if it does. That 8-bit result is compared with the S mask corresponding to whether choices have been registered for each of those categories. If the result is equal, the term shall evaluate True, and shall evaluate False otherwise This test permits selection based upon non-membership (meaning no interest) in all specified genre categories. In other words, the term evaluates True if the categories that had been selected and validated by the viewer resulted in no interest in all of the genre categories specified in the dcc_selection_id field of the DCCT.

The list of category names and their respective codes (Table 6.20) are broken down into two groups. The first group consists of codes 0x20 through 0x26 and may be called the "Basic" group. The second group contains the codes 0x27 through 0xAD and is called the "Detail" group.

The DCC Selection Code Table (see Sec. 6.8) can extend the codes given in Table 6.20. New entries in the Basic group can be defined in the range 0x01 to 0x1F. New entries in the Detail group can be defined in the range 0xAE to 0xFE.

6.7.6 Case: Cannot Be Authorized (dcc_selection_type = 0x09)

If the dcc_selection_type is specified to be of type 0x09, and if the DCCRR has been tuned by the viewer to a major and minor channel number specified by the dcc_from_major_channel_number and the dcc_from_minor_channel_number for which the viewer is not authorized (due to conditional access), the DCCRR shall promptly, upon determination of the unauthorized status, tune to the dcc_to_major_channel_number and dcc_to_minor_channel_number. The action of this mechanism provides an ability to "redirect" viewers to an alternate channel in the event they are not authorized to view the requested channel.

6.7.7 Case: Geographic Location (dcc_selection_type = 0x0C, 0x1C)

location_code — This 24-bit unsigned integer field contains state_code, county_subdivision, and county_code sub fields (defined below) used in identification of a geographic location.

For dcc_selection_type 0x0C, Geographic Location Inclusion, the term shall evaluate True if the geographic location indicated in the matches the geographic location of the DCCRR. If the geographic location indicated in the dcc_selection_id does not match the geographic location of the DCCRR, or if the DCCRR's geographic location is not known, the term shall evaluate False.

For dcc_selection_type 0x1C, Geographic Location Exclusion, the term shall evaluate True if the geographic location indicated in the dcc_selection_id does not match the geographic location of the DCCRR. If the geographic location indicated in the dcc_selection_id matches the geographic location of the DCCRR, or if the DCCRR's geographic location is not known, the term shall evaluate False.

The location_code fields shall be as specified in Table 6.21.

Table 6.21 Conditional Type Value Format

Syntax	No. of Bits	Format
dcc_selection_id {		
reserved	40	0xFFFFFFFFFF
location_code {		
state_code	8	uimsbf range 0..99
county_subdivision	4	uimsbf range 0..9
reserved	2	'11'
county_code	10	uimsbf rang 0..999
}		
}		

state_code — This 8-bit unsigned integer in the range 0 to 99 specifies the State or Territory. state_code is coded according to State and Territory FIPS number codes (see 47 CFR 11.15(f).). A list of state and county codes (extracted from FIPS Pub 6-4) is provided in Annex H of this document. A value of 0 specifies all states and territories.

county_subdivision — This 4-bit unsigned integer in the range 0 to 9 specifies county subdivisions as shown in Table 6.22.

Table 6.22 County Subdivision Coding

county_subdivision	Meaning
0x0	All or an unspecified portion of a county
0x1	Northwest
0x2	North Central
0x3	Northeast
0x4	West Central
0x5	Central
0x6	East Central
0x7	Southwest
0x8	South Central
0x9	Southeast
0xA-0xF	[Reserved]

county_code — This 10-bit unsigned integer in the range 0 to 999 specifies a county within a state. county_code is coded according to State and Territory Federal Information Processing Standard (FIPS) number codes maintained by the National Institute of Standards and Technology (NIST) in FIPS Pub 6-4. A list of state and county codes (extracted from FIPS Pub 6-4) is provided in Annex H of this document. A value of 0 specifies the entire state or territory.

6.7.8 Case: Rating Blocked (dcc_selection_type = 0x0D)

If the dcc_selection_type is specified to be of type 0x0D the term shall evaluate True if the current Virtual Channel is "blocked" as defined in EIA/CEA-766-A due to its Content Advisory and shall evaluate False otherwise. At the discretion of the DCCRR implementation, a timeout may be employed prior to the channel change to allow the viewer to override the blocked condition, for example by entering a parental password. Note that dcc_departing_request_descriptors and dcc_arriving_request_descriptors may be employed to provide a message to viewers that a channel-blocked situation has arisen and that the DCCRR has switched to a new channel.

6.7.9 Case: Return to Original Channel (dcc_selection_type = 0x0F)

If the dcc_selection_type is specified to be of type 0x0F, and if the DCCRR is engaged in a DCC request of type Temporary Retune (dcc_context = 0), the DTV shall tune back to the channel from which it was previously directed (the previous DCC From Channel Number).

6.7.10 Case: Viewer-Direct-Select (dcc_selection_type = 0x20 – 0x23)

If the dcc_selection_type is specified to be of type 0x20 through 0x23, the DCCRR shall tune to the virtual channel specified in the DCC To Channel Number based upon the viewer's selection of one of four "Viewer-Direct-Select" (VDS) function buttons (or equivalent). To correspond with textual or verbal prompting in the video program, the buttons shall be labeled or otherwise identified to the user as "A," "B," "C," and "D." For example, if the viewer chooses Viewer-Direct-Select Button B and a DCC request has been defined for that button (dcc_selection_type value 0x21), the DCCRR shall immediately switch to the DCC To Channel Number associated with the request.

For dcc_selection_type values 0x20 through 0x23, the dcc_selection_id field is a 64-bit number that, when non-zero, enables a Viewer-Direct-Select "persistence" function. Value zero for the dcc_selection_id field for dcc_selection_type values 0x20 through 0x23 indicate that a persistence function for this button choice for this DCC request is not offered.

6.7.11 Viewer-Direct-Select Persistence

The DCCRR shall implement a persistence function for the VDS DCC function. The persistence function allows the DCCRR to automatically take the same branch that was chosen by a prior VDS button selection when a new VDS opportunity arises. The following rules shall be followed:

1) If in response to a VDS opportunity the user selects a button, the DCCRR shall record the value of that button's 64-bit dcc_selection_id (if non-zero).
2) If a new VDS opportunity arises in which a 64-bit dcc_selection_id value appears that corresponds to a recorded ID, the DCCRR shall use that branch as the default (the branch to be taken if no button is selected).

3) If a new VDS opportunity arises in which a 64-bit dcc_selection_id value appears that matches more than one recorded ID, the DCCRR shall take as the default branch the one associated with the most recently saved ID of those matching.
4) Recorded IDs shall be saved in a stack in the DCCRR. The requirements regarding the size and organization of this stack are not specified and are left to the discretion of the implementer.

The way in which the 64-bit VDS ID numbers are managed and assigned is outside the scope of this document. If broadcasters intend for them to work across different transmissions, coordination of their assignment will be required. If they want them *not* to work across to other broadcaster's transmissions, appropriate steps should be taken to assure the 64 bit number is suitably random across the full 64-bit number space.

dcc_selection_id — This 64-bit unsigned integer contains the data identified by the dcc_selection_type field that has been described above in each of the case descriptions and summarized in Table 6.17.

dcc_term_descriptors_length — A 10-bit unsigned integer number that indicates the number of bytes of optional DCC term descriptors (if any) to follow. DCC term descriptors are an expansion mechanism allowing, in a future revision of this protocol, additional data to be associated with a given term. At present, no descriptors are defined for such use. The DCCRR shall disregard any descriptors encountered.

dcc_term_descriptor() — A data structure in standard descriptor format (tag, length, data) that provides additional information about the term defined in this iteration of the dcc_term_count "for" loop.

dcc_test_descriptors_length — A 10-bit unsigned integer number that indicates the number of bytes of optional DCC test descriptors (if any) to follow. DCC test descriptors can provide an additional piece of data to be associated with a given test. At present, two descriptors are defined for such use, the dcc_departing_request_descriptor() defined in Section 6.9.11 and the dcc_arriving_request_descriptor() defined in Section 6.9.12. The DCCRR shall disregard any unsupported descriptors encountered.

dcc_test_descriptor() — A data structure in standard descriptor format (tag, length, data) that provides additional information about the test defined in this iteration of the dcc_test_count "for" loop.

dcc_additional_descriptors_length — A 10-bit unsigned integer number that indicates the number of bytes of optional DCC additional descriptors (if any) to follow. DCC additional descriptors are an expansion mechanism allowing, in a future revision of this protocol, additional data to be associated with a given DCC table section. At present, no descriptors are defined for such use.

dcc_additional_descriptor() — A data structure in standard descriptor format (tag, length, data) that provides additional information about the DCC requests described in this directed_channel_change_table_section().

CRC_32 — This is a 32-bit field that contains the CRC value that ensures a zero output from the registers in the decoder defined in Annex A of ISO/IEC 13818-1 [9] after processing the entire Directed Channel Change Table section.

6.8 DCC Selection Code Table (DCCSCT)

The optional Directed Channel Change Selection Code Table (DCCSCT) carries genre code values and genre criteria name values and/or state/county location codes for use in extending the original data sets of those codes defined in Table 6.20 and Annex H.

The DCC Selection Code Table is carried in private sections with table ID 0xD4, and obeys the syntax and semantics given in Section 4.

The following constraints apply to the Transport Stream packets carrying DCCSCT sections.

- PID shall have the value 0x1FFB (base_PID)
- transport_scrambling_control bits shall have the value '00'
- adaptation_field_control bits shall have the value '01'

The bit stream syntax for the Directed Channel Change Selection Code Table shall be as shown in Table 6.23.

Table 6.23 Bit Stream Syntax for the DCC Selection Code Table

Syntax	No. of Bits	Format
dcc_selection_code_table_section () {		
table_id	8	0xD4
section_syntax_indicator	1	‘1’
private_indicator	1	‘1’
reserved	2	‘11’
section_length	12	uimsbf
dccsct_type	16	uimsbf
reserved	2	‘11’
version_number	5	uimsbf
current_next_indicator	1	‘1’
section_number	8	0x00
last_section_number	8	0x00
protocol_version	8	uimsbf
updates_defined	8	uimsbf
for (i = 0; i < updates_defined; i++) {		
update_type	8	uimsbf
update_data_length	8	uimsbf
if (update_type==new_genre_category) {		
genre_category_code	8	uimsbf
genre_category_name_text()	var	
}		
if (update_type==new_state) {		
dcc_state_location_code	8	uimsbf
dcc_state_location_code_text()	var	
}		
if (update_type==new_county) {		
state_code	8	uimsbf
reserved	6	‘111111’
dcc_county_location_code	10	uimsbf
dcc_county_location_code_text()	var	
}		
reserved	6	‘111111’
dccsct_descriptors_length	10	uimsbf
for (j = 0; j < N; j++) {		
dccsct_descriptors()		
}		
}		
reserved	6	‘111111’
dccsct_additional_descriptors_length	10	uimsbf
for (i = 0; i < N; i++) {		
dccsct_additional_descriptors()		
}		
CRC_32	32	rpchof
}		

table_id — This is an 8-bit field, which shall be set to 0xD4, identifying this table as the DCC Selection Code Table (DCCSCT).

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.

dccsct_type — A 16-bit unsigned integer field whose value specifies the type of DCC selection code information contained within the table section, and its syntax and semantics. Currently only dccsct_type value 0x0000 is defined. The DCCRR is expected to discard DCCSCT table sections with nonzero values of dccsct_type until such time as they are standardized and supported.

version_number — This 5-bit field is the version number of the DCCSC Table identified by the combination of the table_id and dccsct_type fields. The version number shall be incremented by 1 modulo 32 when any field in this instance of the DCC Selection Code Table changes. The value of the version_number shall be identical to that of the corresponding entry in the MGT.

current_next_indicator — This 1-bit indicator is always set to '1'.

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

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

protocol_version — An 8-bit unsigned integer field whose function is 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 0x00. Non-zero values of protocol_version may be used by a future version of this standard to indicate structurally different tables.

update_type – This 8-bit field indicates the type of update to be supplied in this iteration of the "for" loop. Table 6.24 defines the coding.

Table 6.24 Update Type Coding

update_type	Meaning
0x00	Reserved
0x01	**new_genre_category**—Genre table update
0x02	**new_state**—Addition to state code data
0x03	**new_county**—Addition to county code data
0x04-0xFF	Reserved for future use

update_data_length – An unsigned integer field that shall indicate the number of bytes of data in the "if" statement to follow. After skipping ahead the number of bytes given by update_length, the next field will be the 6-bit reserved field ahead of dccsct_descriptors_length. Receiving devices are expected to use update_data_length to skip data for unknown values of update_type.

genre_category_code – An 8-bit unsigned integer code that references a reserved value in the Categorical Genre Code Assignment Table (Table 6.20). Values for genre_category_code shall be in the range 0x01 to 0x1F for expansion of the Basic genre categories, or in the range 0xAE through 0xFE for expansion of the Detailed genre categories. The integer values specifying new Categorical Genre Codes shall be those assigned by the ATSC.

selection_category_name_text() — A data structure containing a multiple string structure which specifies the genre category name; e.g., "Rugby." Text strings are formatted according to the rules outlined in Section 6.10. The displayed string for the genre category name shall be limited to 24 characters or less.

dcc_state_location_code — This 8-bit unsigned integer in the range 79 to 99 specifies the State or Territory to be added. The integer values specifying new States or Territories shall be those assigned by the ATSC.

dcc_state_location_code_text() — The name of the new State or Territory in the format of a multiple string structure (see Section 6.10).

state_code — This 8-bit unsigned integer number identifies the state to which the county information to follow applies. Allowed values for state_code include the codes associated with the states defined in Annex H[10] plus the codes for any new states added by state updates in this DCCSCT.

dcc_county_location_code — This 10-bit unsigned integer in the range 1 to 999 specifies a new county within the state identified in state_code. Allowed values for dcc_county_location_code shall be limited to values not already assigned in Annex H for the state identified in state_code[11]. The integer values specifying new counties shall be those assigned by the ATSC.

dcc_county_location_code_text() — The name of a new county in the format of a multiple string structure (see Section 6.10).

dccsct_descriptors_length — Total length of the descriptor list that follows (in bytes).

dccsct_descriptors() — Zero or more descriptors, as appropriate, may be included.

CRC_32 — This is a 32-bit field that contains the CRC value that ensures a zero output from the registers in the decoder defined in Annex A of ISO/IEC 13818-1 "MPEG-2 Systems" after processing the entire DCC Selection Code Table section.

6.9 Core Descriptors

Table 6.25 lists descriptor tags along with their names, and shows the use of each when the relevant table section listed is present in the transport stream. Table 6.25 indicates all the core descriptors, their tags, and their allowed locations in the PSIP tables. The service_location_descriptor() shall always be present in the terrestrial VCT (shown with an "S"). Table 6.25 does not specify the rules governing whether or not a particular descriptor must be present in any given situation. When used, some descriptors shall be in each indicated location (shown with an "M"). Some descriptors also may be present in a second location within either the terrestrial or the cable case (shown with an "O"). Asterisks mark the tables where the descriptors may appear without restrictions.

---

[10] Although unlikely to occur, this provision allows the DCCSCT to indicate a change to an existing state's spelling or name.

[11] If a county changes its name, the convention is to add a new county code rather than re-define the old one.

Table 6.25a List and Location of PSIP Descriptors

Descriptor Name	Descriptor Tag	Terrestrial Broadcast					
		PMT	MGT	VCT	EIT	DCCT	DCCSCT
stuffing descriptor	0x80	*	*	*	*	*	*
AC-3 audio descriptor	0x81	M			M		
caption service descriptor	0x86	O			M		
content advisory descriptor	0x87	O			M		
extended channel name descriptor	0xA0			M			
service location descriptor	0xA1			S			
time-shifted service descriptor	0xA2			M			
component name descriptor	0xA3	M					
DCC departing request descriptor	0xA8					M	
DCC arriving request descriptor	0xA9					M	
redistribution control descriptor	0xAA	M			M		

Table 6.25b List and Location of PSIP Descriptors

Descriptor Name	Descriptor Tag	Cable					
		PMT	MGT	VCT	EIT	DCCT	DCCSCT
stuffing descriptor	0x80	*	*	*	*	*	*
AC-3 audio descriptor	0x81	M			M		
caption service descriptor	0x86	M			M		
content advisory descriptor	0x87	M			M		
extended channel name descriptor	0xA0			M			
service location descriptor	0xA1			M			
time-shifted service descriptor	0xA2			M			
component name descriptor	0xA3	M					
DCC departing request descriptor	0xA8					M	
DCC arriving request descriptor	0xA9					M	
redistribution control descriptor	0xAA	M			M[12]		

The descriptors listed here are those currently defined. Others may be defined in the future. Receivers are expected to disregard unsupported descriptors. Additions to PSIP functionality are contemplated in the future and may result in additional descriptors being present in each of these tables.

6.9.1 AC-3 Audio Descriptor

The AC-3 audio descriptor, of Ref. [1] may be used in the EIT.

[12] When the EIT is present.

6.9.2 Program Identifier Descriptor

The program_identifier_descriptor, as defined in Ref. [5], shall not be used in any PSIP descriptor loops.

6.9.3 Caption Service Descriptor

The caption service descriptor provides closed captioning information, such as closed captioning type and language code for events with closed captioning service. This descriptor shall appear in the descriptor() loop for each EIT event that has closed captioning.

A caption_service_descriptor() may be present in the descriptor loop describing the video stream_type, immediately following the ES_info_length field in the TS_program_map_section() (as defined in ISO/IEC 13818-1 [9] sections 2.4.4.8 and 2.4.4.9).

The bit stream syntax for the closed captioning service descriptor is shown in Table 6.26.

Table 6.26 Bit Stream Syntax for the Caption Service Descriptor

Syntax	No. of Bits	Format
caption_service_descriptor () {		
descriptor_tag	8	0x86
descriptor_length	8	uimsbf
reserved	3	‘111’
number_of_services	5	uimsbf
for (i=0;i<number_of_services;i++) {		
language	8*3	uimsbf
cc_type	1	bslbf
reserved	1	‘1’
if (cc_type==line21) {		
reserved	5	‘11111’
line21_field	1	bslbf
}		
else		
caption_service_number	6	uimsbf
easy_reader	1	bslbf
wide_aspect_ratio	1	bslbf
reserved	14	‘11111111111111’
}		
}		

descriptor_tag — An 8-bit field that identifies the type of descriptor. For the caption_service_descriptor() the value is 0x86.

descriptor_length — An 8-bit count of the number of bytes following the descriptor_length itself.

number_of_services — An unsigned 5-bit integer in the range 1 to 16 that indicates the number of closed caption services present in the associated video service. Note that if the video service does not carry television closed captioning, the caption_service_descriptor() shall not be present either in the Program Map Table or in the Event Information Table.

Each iteration of the “for” loop defines one closed caption service present as a sub-stream within the 9600 bit per second closed captioning stream. Each iteration provides the sub-stream’s language, attributes, and (for advanced captions) the associated Service Number reference. Refer to Ref. [13] for a description of the use of the Service Number field within the syntax of the closed caption stream.

language — A 3-byte language code per ISO 639.2/B (Ref. [6]) defining the language associated with one closed caption service. 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.

cc_type — A flag that indicates, when set, that an advanced television closed caption service is present in accordance with Ref. [13]. When the flag is clear, a line-21 closed caption service is present. For line 21 closed captions, the line21_field field indicates whether the service is carried in the even or odd field.

line21_field — A flag that indicates, when set, that the line 21 closed caption service is associated with the field 2 of the NTSC waveform. When the flag is clear, the line-21 closed caption service is associated with field 1 of the NTSC waveform. The line21_field flag is defined only if the cc_type flag indicates line-21 closed caption service.

caption_service_number — A 6-bit unsigned integer value in the range zero to 63 that identifies the Service Number within the closed captioning stream that is associated with the language and attributes defined in this iteration of the “for” loop. See Ref. [13] for a description of the use of the Service Number. The caption_service_number field is defined only if the cc_type flag indicates closed captioning in accordance with Ref. [13].

easy_reader — A Boolean flag which indicates, when set, that the closed caption service contains text tailored to the needs of beginning readers. Refer to Ref. [13] for a description of “easy reader” television closed captioning services. When the flag is clear, the closed caption service is not so tailored.

wide_aspect_ratio — A Boolean flag which indicates, when set, that the closed caption service is formatted for displays with 16:9 aspect ratio. When the flag is clear, the closed caption service is formatted for 4:3 display, but may be optionally displayed centered within a 16:9 display.

6.9.4 Content Advisory Descriptor

The Content Advisory Descriptor is used to indicate, for a given event, ratings for any or all of the rating dimensions defined in the RRT (Rating Region Table). Ratings may be given for any or all of the defined regions, up to a maximum of 8 regions per event. An Event without a Content Advisory Descriptor indicates that the rating value for any rating dimension defined in any rating region is zero. The absence of ratings for a specific dimension is completely equivalent to having a zero-valued rating for such a dimension. The absence of ratings for a specific region implies the absence of ratings for all of the dimensions in the region. The absence of a Content Advisory Descriptor for a specific event implies the absence of ratings for all of the regions for the event.

A content_advisory_descriptor() may be present in the descriptor loop immediately following the program_info_length field in the TS_program_map_section() (as defined in ISO/IEC 13818-1 [9] sections 2.4.4.8 and 2.4.4.9). This descriptor shall appear in the descriptor() loop for each EIT event that has closed captioning.

The bit stream syntax for the Content Advisory Descriptor is shown inTable 6.27.

Table 6.27 Stream Syntax for the Content Advisory Descriptor

Syntax	No. of Bits	Format
content_advisory_descriptor () {		
descriptor_tag	8	0x87
descriptor_length	8	uimsbf
reserved	2	'11'
rating_region_count	6	
for (i=0; i<rating_region_count; i++) {		
rating_region	8	uimsbf
rated_dimensions	8	uimsbf
for (j=0;j<rated_dimensions;j++) {		
rating_dimension_j	8	uimsbf
reserved	4	'1111'
rating_value	4	uimsbf
}		
rating_description_length	8	uimsbf
rating_description_text()	var	
}		
}		

descriptor_tag — This 8-bit unsigned integer shall have the value 0x87, identifying this descriptor as content_advisory_descriptor.

descriptor_length — This 8-bit unsigned integer specifies the length (in bytes) immediately following this field up to the end of this descriptor.

rating_region_count — A 6-bit unsigned integer value in the range 1 to 8 that indicates the number of rating region specifications to follow.

rating_region — An unsigned 8-bit integer that specifies the rating region for which the data in the bytes to follow is defined. The rating_region associates ratings data given here with data defined in a Ratings Region Table tagged with the corresponding rating region.

rated_dimensions — An 8-bit unsigned integer field that specifies the number of rating dimensions for which content advisories are specified for this event. The value of this field shall not be greater than the value specified by the field dimensions_defined in the corresponding RRT section.

rating_dimension_j — An 8-bit unsigned integer field specifies the dimension index into the RRT instance for the region specified by the field rating_region. These dimension indices shall be listed in numerical order; i.e., the value of rating_dimension_j+1 shall be greater than that of rating_dimension_j.

rating_value — A 4-bit field represents the rating value of the dimension specified by the field rating_dimension_j for the region given by rating_region.

rating_description_length — An 8-bit unsigned integer value in the range zero to 80 that represents the length of the rating_description_text() field to follow.

rating_description_text() — The rating description in the format of a multiple string structure (see Section 6.10). The rating_description display string shall be limited to 16 characters or less. The rating description text shall represent the program's rating in an abbreviated form suitable for on-screen display. The rating description text collects multidimensional text information into a single small text string. If "xxx" and "yyy" are abbreviated forms for rating values in two dimensions, then "xxx-yyy" and "xxx (yyy)" are examples of possible strings represented in rating_description_text().

6.9.5 Extended Channel Name Descriptor

The extended channel name descriptor provides the long channel name for the virtual channel containing this descriptor.

The bit stream syntax for the extended channel name descriptor is shown in Table 6.28.

Table 6.28 Bit Stream Syntax for the Extended Channel Name Descriptor

Syntax	No. of Bits	Format
extended_channel_name_descriptor () {		
descriptor_tag	8	0xA0
descriptor_length	8	uimsbf
long_channel_name_text()	var	
}		

descriptor_tag — This 8-bit unsigned integer shall have the value 0xA0, identifying this descriptor as extended_channel_name_descriptor().

descriptor_length — This 8-bit unsigned integer specifies the length (in bytes) immediately following this field up to the end of this descriptor.

long_channel_name_text() — The long channel name in the format of a multiple string structure (see Section 6.10).

6.9.6 Service Location Descriptor

This descriptor specifies the stream types, PID and language code for each elementary stream. An instance of this descriptor shall appear in the TVCT for each active channel. A service_location_descriptor() shall not be present for any inactive channel. When present, the service_location_descriptor() must be valid for the current event in the corresponding virtual channel.

Note that for cable, the information in the service_location_descriptor() is carried in the PMT with the syntax defined by Ref. [9].

The service_location_descriptor() shall indicate the same Elementary Stream data as the corresponding portion of the Program Map Table currently being transmitted. At minimum, the Service Location Descriptor shall include the video elementary stream (if one is present in the service), and all audio streams present in the service.

The bit stream syntax for the service_location_descriptor() is shown in Table 6.29.

Table 6.29 Bit Stream Syntax for the Service Location Descriptor

Syntax	No. of Bits	Format
service_location_descriptor () {		
descriptor_tag	8	0xA1
descriptor_length	8	uimsbf
reserved	3	'111'
PCR_PID	13	uimsbf
number_elements	8	uimsbf
for (i=0;i<number_elements;i++) {		
stream_type	8	uimsbf
reserved	3	'111'
elementary_PID	13	uimsbf
ISO_639_language_code	8*3	uimsbf
}		
}		

descriptor_tag — This 8-bit unsigned integer shall have the value 0xA1, identifying this descriptor as service_location_descriptor().

descriptor_length — This 8-bit unsigned integer specifies the length (in bytes) immediately following this field up to the end of this descriptor.

PCR_PID — This is a 13 bit field indicating the PID of the Transport Stream packets which shall contain the PCR fields valid for the program specified by program_number. If no PCR is associated with a program definition for private streams then this field shall take the value of 0x1FFF. The value of PCR_PID shall be the same as the PCR_PID field of the TS_program_map_section() currently being transmitted for that program number.

number_elements — This 8-bit unsigned integer indicates the number of PIDs used for this program.

stream_type — This 8-bit unsigned integer field specifies the type of the elementary stream. Values for each stream_type, elementary_PID, and associated ISO 639 language code shall be the same as those delivered in the corresponding ES_info loop of the TS_program_map_section() currently being transmitted. Informative Table 6.30 lists assignments for selected stream_type values.

Table 6.30 Selected Stream Type Assignments (Informative)

stream_type	Description
0x02	ITU-T Rec. H.262 | ISO/IEC 13818-2 Video or ISO/IEC 11172-2 constrained parameter video stream
0x06	PES packets containing A/90 streaming, synchronized data
0x0B	DSM-CC sections containing A/90 asynchronous data
0x0D	DSM-CC addressable sections per A/90
0x14	DSM-CC sections containing non-streaming, synchronized data per A/90
0x81	Audio per ATSC A/53B [2] Annex B
0x95	Sections conveying A/90 Data Service Table, Network Resources Table
0xC2	PES packets containing A/90 streaming, synchronous data

elementary_PID — Packet Identifier for the elementary stream. The value for elementary_PID, stream_type, and the associated ISO 639 language code shall be the same as those delivered in an ES_info loop of the TS_program_map_section()currently being transmitted.

ISO_639_language_code — This 3-byte (24 bits) field, in conformance with ISO 639.2/B, specifies the language used for the elementary stream. In case of no language specified for this elementary stream, e.g. video, each byte shall have the value 0x00. The value of the ISO_639_language_code field in the service_location_descriptor() shall be the same as the value in the corresponding ES_info loop of the TS_program_map_section()currently being transmitted. If no ISO_639_language_descriptor() is present in the ES_info loop for this Elementary Stream in the currently transmitted TS_program_map_section(), each byte of the ISO_639_language_code field shall be zero.

6.9.7 Time-Shifted Service Descriptor

This descriptor links one virtual channel with one or more virtual channels that carry the same programming on a time-shifted basis. The typical application is for Near Video On Demand (NVOD) services. The time_shifted_service_descriptor() shall be used only in the case that the time shift is constant across the time period covered by all EITs.

The base channel of a set of NVOD channels is the sole channel in that set that carries a time_shifted_service_descriptor() within its VCT entry. An NVOD base channel's EIT subtables shall be the same as that of a normal channel, except that EIT-0, in addition to its normal entries, shall list all events that have *expired* in the base channel's schedule until they have expired in all its child channels. NVOD child services derive their EIT information from that of the base channel by applying the time_shift specified in the base channels' time_shifted_service_descriptor(). ETMs for events expired in the base channel shall be carried on the PID listed for ETT-0 until they are no longer referenced by EIT-0.

Two example scenarios can be found in informative Annex D Section 7.

The bit stream syntax for the time_shifted_service_descriptor() is shown in Table 6.31.

Table 6.31 Bit Stream Syntax for the Time Shifted Service Descriptor

Syntax	No. of Bits	Format
time_shifted_service_descriptor () {		
descriptor_tag	8	0xA2
descriptor_length	8	uimsbf
reserved	3	'111'
number_of_services	5	uimsbf
for (i=0;i<number_of_services;i++) {		
reserved	6	'111111'
time_shift	10	uimsbf
reserved	4	'1111'
major_channel_number	10	uimsbf
minor_channel_number	10	uimsbf
}		
}		

descriptor_tag — This 8-bit unsigned integer shall have the value 0xA2, identifying this descriptor as time_shifted_service_descriptor().

descriptor_length — This 8-bit unsigned integer specifies the length (in bytes) immediately following this field up to the end of this descriptor.

number_of_services — A 5-bit number in the range 1 to 20 that indicates the number of time-shifted services being defined here.

time_shift — A 10-bit number in the range 1 to 720 that represents the number of minutes the time-shifted service indicated by major_channel_number and minor_channel_number is time-shifted from the virtual channel associated with this descriptor.

major_channel_number — A 10-bit number in the range 1 to 999 that represents the “major” channel number associated with a time-shifted service.

minor_channel_number — A 10-bit number in the range 0 to 999 that, when non-zero, represents the “minor” or “sub-“ channel number of the virtual channel that carries a time-shifted service.

6.9.8 Component Name Descriptor

Table 6.32 defines the component_name_descriptor(), which serves to define an optional textual name tag for any component of the service.

Whenever a service includes two or more audio components labeled with the same ISO_639_language_code (in the ISO 639 Language Descriptor) and bsmod (in the AC-3 Audio Descriptor), a unique component_name_descriptor() shall be placed into the descriptor loop immediately following ES_info_length in the TS_program_map_section() describing each such audio component.

Whenever a service includes an audio component whose bit stream mode (bsmod, as indicated in the AC-3 Audio Descriptor) is music and effects (ME) (bsmod = 1), and the component is a full service suitable for presentation (as indicated by the full_svc flag in the AC-3 Audio Descriptor being set) a unique component_name_descriptor() should be placed into the descriptor loop immediately following ES_info_length in the TS_program_map_section() of the TS_program_map_section() describing that component.

Table 6.32 Bit Stream Syntax for the Component Name Descriptor

Syntax	No. of Bits	Format
component_name_descriptor() {		
descriptor_tag	8	0xA3
descriptor_length	8	uimsbf
component_name_string()	var	
}		

descriptor_tag — This 8-bit unsigned integer shall have the value 0xA3, identifying this descriptor as component_name_descriptor.

descriptor_length — This 8-bit unsigned integer specifies the length (in bytes) immediately following this field up to the end of this descriptor.

component_name_string() — The name string in the format of a multiple string structure (see Section 6.10).

6.9.9 Stuffing Descriptor

For certain applications it is necessary to define a block of N bytes as a placeholder. The N bytes themselves are not to be processed or interpreted. The stuffing_descriptor() is specified for this purpose. The stuffing_descriptor() is simply a descriptor type for which the contents, as indicated by the descriptor_length field, are to be disregarded. The tag type for the stuffing descriptor is 0x80. The stuffing_descriptor() may appear where descriptors are allowed in any table defined in the PSIP.

6.9.10 Descriptors for Inactive Channels

The service_location_descriptor() shall not be present for inactive channels. Any other descriptors, if present, shall provide valid information about the inactive channel. The extended_channel_name_descriptor(), for example, can be used to provide the long-form channel name of the inactive channel.

6.9.11 DCC Departing Request Descriptor

This descriptor provides instructions for the actions to be performed by a DCCRR upon detection of a manual channel change requested by the viewer using the channel change controls on the DCCRR or a DCCRR remote control device, or any DCC term set that evaluates to cause a channel change, just prior to executing the channel change itself. This function shall be defeatable by the viewer within setup menu selections and shall default to “not enabled” if the viewer does not explicitly enable it. This descriptor may appear within the dcc_additional_descriptor loop if it is desired to associate a departing request with a manual channel change, and within the dcc_test_descriptor loop if it is desired to associate a departing request with a particular virtual channel's DCC event.

The bit stream syntax for the dcc_departing_request_descriptor() is shown in Table 6.33.

Table 6.33 Bit Stream Syntax for the DCC Departing Request Descriptor

Syntax	No. of Bits	Format
dcc_departing_request_descriptor () {		
descriptor_tag	8	0xA8
descriptor_length	8	uimsbf
dcc_departing_request_type	8	uimsbf
dcc_departing_request_text_length	8	uimsbf
dcc_departing_request_text()	var	
}		

descriptor_tag — This 8-bit unsigned integer shall have the value 0xA8, identifying this descriptor as dcc_departing_request_descriptor().

descriptor_length — This 8-bit unsigned integer specifies the length (in bytes) immediately following this field up to the end of this descriptor.

dcc_departing_request_type — This 8-bit unsigned integer specifies the type of the DCC departing request and shall have the values listed in Table 6.34.

Table 6.34 DCC Departing Request Type Assignments

dcc_departing_request_type	Meaning
0x00	Reserved.
0x01	Cancel any outstanding departing request type and immediately perform a channel change upon request by the viewer.
* 0x02	Display departing request text in a centered window for a minimum of 10 seconds prior to performing the channel change requested by the viewer or for a lesser amount of time if the viewer issues another channel change request or a "continue", "OK", "proceed", or equivalent command.
* 0x03	Display departing request text in a centered window indefinitely until viewer issues another channel change request or a "continue", "OK", "proceed", or equivalent command.
0x04-0xFF	Reserved

* Note: The above suggested behavior of the DCCRR's implementation of Departing Request types 0x02 and 0x03 are within the discretion of DCCRR manufacturers. The DCCRR's reaction to these commands may also be disabled by viewers through an interactive setup session.

dcc_departing_request_text_length — An 8-bit unsigned integer number that specifies the total length in bytes of the dcc_departing_request_text() field to follow.

dcc_departing_request_text() — The departing request window text in the format of a multiple string structure (see Section 6.10).

6.9.12 DCC Arriving Request Descriptor

This descriptor provides instructions for the actions to be performed by a DCCRR upon arrival at a newly changed channel. The arrival channel change request shall be executed within 30 seconds of arrival at, and detection within, the channel PSIP stream (this implies that, and is dependent upon, the descriptor being repeated or issued by the broadcaster and detected by the DCCRR in at least 30 second cycles). The dcc_arriving_request_descriptor shall only be located within the dcc_test_descriptor loop. This function shall be defeatable by the viewer within setup menu selections and shall default to "not enabled" if the viewer does not explicitly enable it.

The bit stream syntax for the dcc_arriving_request_descriptor() is shown in Table 6.35.

Table 6.35 Bit Stream Syntax for the DCC Arriving Request Descriptor

Syntax	No. of Bits	Format
dcc_arriving_request_descriptor () {		
descriptor_tag	8	0xA9
descriptor_length	8	uimsbf
dcc_arriving_request_type	8	uimsbf
dcc_arriving_request_text_length	8	uimsbf
dcc_arriving_request_text()	var	
}		

descriptor_tag — This 8-bit unsigned integer shall have the value 0xA9, identifying this descriptor as dcc_arriving_request_descriptor().

descriptor_length — This 8-bit unsigned integer specifies the length (in bytes) immediately following this field up to the end of this descriptor.

dcc_arriving_request_type — This 8-bit unsigned integer specifies the type of the DCC arriving request and shall have the values listed in Table 6.36.

Table 6.36 DCC Arriving Request Type Assignments

dcc_arriving_request_type	Meaning
0x00	Reserved
0x01*	Display arriving request text in a centered window for a minimum of 10 seconds after performing the channel change requested by the viewer, or for a less amount of time if the viewer issues a "continue", "OK", "proceed", or equivalent command.
0x02*	Display arriving request text in a centered window indefinitely after performing a channel change request requested by the viewer until viewer issues a "continue", "OK", "proceed", or equivalent command.
0x03-0xFF	Reserved

* Note: The above suggested behavior of the DCCRR's implementation of Arriving Request types 0x01 and 0x02 are within the discretion of DCCRR manufacturers. The DCCRR's reaction to these commands may also be disabled by viewers through an interactive setup session.

dcc_arriving_request_text_length — An 8-bit unsigned integer number that specifies the total length in bytes of the dcc_arriving_request_text() field to follow.

dcc_arriving_request_text() — The arriving request window text in the format of a multiple string structure (see Section 6.10).

6.9.13 Redistribution Control (RC) Descriptor

The purpose of the Redistribution Control descriptor is to convey a certain type of redistribution information held by the program rightsholder for audio, video, or data events. The descriptor's existence within the ATSC stream shall mean: "technological control of consumer redistribution is signaled."

The redistribution control information conveyed by the rc_descriptor() defined in Table 6.37 concerns the video/audio/data programming identified either by the event_id within the EIT or the program_number within the TS_program_map_section().

For terrestrial broadcast transport, the rc_descriptor(), when transmitted, shall be present in both the EIT and TS_program_map_section(). For cable transport, the rc_descriptor(), when transmitted, shall be present in the TS_program_map_section(), and, when the EIT is carried, in the EIT.

The rc_descriptor(), when in the EIT, shall apply to a specific event associated with the Virtual Channel and the related MPEG-2 Program. It shall be placed within the descriptor loop after descriptors_length for the event_id for which this information is being signaled. The rc_descriptor(), shall be placed within the descriptor loop after program_info_length in the TS_program_map_section() for the program_number for which this information is being signaled. When the descriptor is placed in the TS_program_map_section(), it shall also be placed in the current event of EIT-0 for the Virtual Channel associated with the MPEG-2 Program; and it should be placed in the EIT for this event as far in advance as possible (i.e., minimally EIT-1, EIT-2, and EIT-3).

For data-only services[13], the rc_descriptor() shall be placed in the DET (whose syntax and semantics are defined in ATSC A/90 Data Broadcasting Standard) under the same provisions described for the EIT.

It is out of the scope of this standard to assert how any receiving device reacts when the rc_descriptor is present.

The bit stream syntax for the redistribution control descriptor is shown in Table 6.37.

Table 6.37 Bit Stream Syntax for the Redistribution Control Descriptor

Syntax	No. of Bits	Format
rc_descriptor() {		
descriptor_tag	8	0xAA
descriptor_length	8	uimsbf
for (i = 0; i < descriptor_length; i++) {		
rc_information()	8	uimsbf
}		
}		

descriptor_tag — This 8-bit unsigned integer shall have the value 0xAA, identifying this descriptor as the rc_descriptor().

descriptor_length — This 8-bit unsigned integer specifies the length (in bytes) immediately following this field through the last byte of this descriptor. The descriptor_length may, in the future, have a value other than 0x00. If the descriptor_length is not 0x00, optional information having a length of descriptor_length shall be contained within the rc_information field.

rc_information() — Optional additional redistribution control information that may be defined in the future.

6.10 Multiple String Structure

This is a general data structure used specifically for text strings. Text strings appear as event titles, long channel names, the ETT messages, and RRT text items. The bit stream syntax for the Multiple String Structure is shown in Table 6.38.

[13] As defined in the ATSC A/90 Data Broadcast Standard.

Table 6.38 Bit Stream Syntax for the Multiple String Structure

Syntax	No. of Bits	Format
multiple_string_structure () {		
number_strings	8	uimsbf
for (i= 0;i< number_strings;i++) {		
ISO_639_language_code	24	uimsbf
number_segments	8	uimsbf
for (j=0;j<number_segments;j++) {		
compression_type	8	uimsbf
mode	8	uimsbf
number_bytes	8	uimsbf
for (k= 0;k<number_bytes;k++)		
compressed_string_byte [k]	8	bslbf
}		
}		
}		

number_strings — This 8-bit unsigned integer field identifies the number of strings in the following data.

ISO_639_language_code — This 3-byte (24 bits) field, in conformance with ISO 639.2/B, specifies the language used for the i^th^ string.

number_segments — This 8-bit unsigned integer field identifies the number of segments in the following data. A specific mode is assigned for each segment.

compression_type — This 8-bit field identifies the compression type for the j^th^ segment. Allowed values for this field are shown in Table 6.39. Compression types 0x01 and 0x02 shall be restricted for use with text mode 0x00.

Table 6.39 Compression Types

compression_type	Compression Method
0x00	No compression
0x01	Huffman coding using standard encode/decode tables defined in Table C.4 and C.5 in Annex C.
0x02	Huffman coding using standard encode/decode tables defined in Table C.6 and C.7 in Annex C.
0x03 to 0xAF	reserved
0xB0 to 0xFF	Used in other systems

mode — An 8-bit value representing a mode to be used to interpret the segment's contents as encoded character (textual) data. Allowed values for this field are shown in Table 6.40. In the case that a decoder does not support a mode, then the string entry that includes that mode within the multiple string structure shall be ignored. When using modes 0x01-0x06, 0x09-0x0E, 0x10, 0x20-0x27, 0x30-0x33, the value of compression_type shall be 0x00 (no compression).

When mode values 0x01-0x06, 0x09-0x0E, 0x10, 0x20-0x27, or 0x30-0x33 are used, the segment's bytes shall represent the least significant 8-bits of a sequence of 16-bit Unicode code values where the most significant 8-bits of these code values is implied by the mode value itself. For example, the sequence of Unicode code values [0x0E50, 0x0E51, 0x0E52] may be represented by a single segment with mode 0x0E and with segment string bytes [0x50, 0x51, 0x52]. These modes provide a simple form of run-length encoding for a sequence of 16-bit Unicode code values.

See [19] for a detailed specification of the encoding of mode 0x3E.

When using mode 0x3F, the UTF-16 representation of Unicode character data shall apply. UTF-16 is a sequence of 16-bit Unicode code values where each code value either (1) designates a particular standardized or user private character; or (2) forms either the low or high part of a surrogate pair that collectively designates a particular standardized or user private character. See The Unicode Standard [18] for precise specification of UTF-16. When using mode 0x3F, 16-bit Unicode code values comprising UTF-16 code elements shall be encoded as unsigned integers, most significant bit first (uimsbf).

It is recommended that the representation of strings not covered by mode 0x00 make use of either modes 0x3E (preferred) or 0x3F (alternate) according to whichever mode results in the fewest octets.

number_bytes — This 8-bit unsigned integer field identifies the number of bytes that follows.

compressed_string_byte[k] — The kth byte of the jth segment. This field shall occupy an integral number of octets and shall be padded as necessary to satisfy this requirement. The interpretation of these octets shall be in accordance with the segment's mode and compression_type.

Table 6.40 Modes

Mode	Meaning
0x00	Select Unicode Code Range 0x0000 – 0x00FF
0x01	Select Unicode Code Range 0x0100 – 0x01FF
0x02	Select Unicode Code Range 0x0200 – 0x02FF
0x03	Select Unicode Code Range 0x0300 – 0x03FF
0x04	Select Unicode Code Range 0x0400 – 0x04FF
0x05	Select Unicode Code Range 0x0500 – 0x05FF
0x06	Select Unicode Code Range 0x0600 – 0x06FF
0x07-0x08	Reserved
0x09	Select Unicode Code Range 0x0900 – 0x09FF
0x0A	Select Unicode Code Range 0x0A00 – 0x0AFF
0x0B	Select Unicode Code Range 0x0B00 – 0x0BFF
0x0C	Select Unicode Code Range 0x0C00 – 0x0CFF
0x0D	Select Unicode Code Range 0x0D00 – 0x0DFF
0x0E	Select Unicode Code Range 0x0E00 – 0x0EFF
0x0F	Select Unicode Code Range 0x0F00 – 0x0FFF
0x10	Select Unicode Code Range 0x1000 – 0x10FF
0x11-0x1F	Reserved
0x20	Select Unicode Code Range 0x2000 – 0x20FF
0x21	Select Unicode Code Range 0x2100 – 0x21FF
0x22	Select Unicode Code Range 0x2200 – 0x22FF
0x23	Select Unicode Code Range 0x2300 – 0x23FF
0x24	Select Unicode Code Range 0x2400 – 0x24FF
0x25	Select Unicode Code Range 0x2500 – 0x25FF
0x26	Select Unicode Code Range 0x2600 – 0x26FF
0x27	Select Unicode Code Range 0x2700 – 0x27FF
0x28-0x2F	Reserved
0x30	Select Unicode Code Range 0x3000 – 0x30FF
0x31	Select Unicode Code Range 0x3100 – 0x31FF
0x32	Select Unicode Code Range 0x3200 – 0x32FF
0x33	Select Unicode Code Range 0x3300 – 0x33FF
0x34-0x3D	Reserved
0x3E	Select Standard Compression Scheme for Unicode (SCSU) [19]
0x3F	Select Unicode, UTF-16 Form
0x40-0x41	Assigned to ATSC standard for Taiwan
0x42-0x47	Reserved for future ATSC use
0x48	Assigned to ATSC standard for South Korea
0x49-0xDF	Reserved for future ATSC use
0xE0-0xFE	Used in other systems
0xFF	Not applicable

7. PSIP STD MODEL

7.1 Buffer Model for Terrestrial Broadcast

Table 7.1 lists the maximum cycle time for all PSIP tables, except EITs and ETTs. Table 7.2 lists the maximum transmission rate for PSIP packet streams according to their PIDs. The recommended maximum cycle time for EIT-0 is 500 ms.

Table 7.1 Maximum Cycle Time for the STT, MGT, VCT and RRT

Table	STT	MGT	VCT	RRT
Cycle time (ms)	1000	150	400	60000

Table 7.2 Maximum Rate for Each PSIP Packet Stream

PID	base_PID	EIT_PID	ETT_PID
Rate (bps)	250,000	250,000	250,000

For terrestrial broadcast applications the following constraints apply:

- In terrestrial broadcast applications, the PSIP elementary streams identified by Transport Stream packets with PID 0x1FFB (base_PID), EIT PIDs and ETT PIDs shall adhere to an STD model with the following parameters:
- sb_leak_rate shall be 625 (indicating a leak rate of 250,000 bps)
- sb_size shall be 1024 (indicating a smoothing buffer size of 1024 bytes)

7.2 Buffer Model for Cable

Transmission rates for cable will be standardized by the SCTE.

7.3 Buffer Model Considerations to Support Directed Channel Change for Terrestrial Broadcast

The maximum cycle time for the Directed Channel Change Table (DCCT) is recommended not to exceed 150 ms. while a DCC request is in progress. The maximum cycle time for the DCCT is recommended not to exceed 400 ms. within 2 seconds of an impending DCC request. It is recommended that there be no maximum cycle time for the DCCT if there are no impending DCC requests.

The maximum cycle time for the Directed Channel Change Selection Code Table (DCCSCT) is recommended not to exceed 1 hour.

Annex A

Daylight Savings Time Control

(Normative)

In order to convert GPS into local time, the receiver needs to store a time offset (from GPS to local time) in local memory and an indicator as to whether daylight savings is observed. These two quantities can be obtained from the user interface (indicating time zone and daylight savings observance) or from the conditional access system, if present, and stored in non-volatile receiver memory.

Since there is a common time (GPS) transmitted in the PSIP, there needs to be a mechanism to indicate when the receiver should switch into (or out of) daylight savings time at the appropriate local time. Once all the receivers have transitioned at their local times, the entire system can be shifted into daylight savings time. This is accomplished by appropriate setting of the daylight_savings in the STT. The structure of daylight savings time control is shown in Table A1, and the basic use of daylight savings fields through the year is shown in Table A2.

Table A1 Structure of Daylight Savings Time Control

Syntax	No. of Bits	Format
daylight_savings () {		
DS_status	1	bslbf
reserved	2	'11'
DS_day_of_month	5	uimsbf
DS_hour	8	uimsbf
}		

DS_status — This bit indicate the status of daylight savings.

DS_status = '0': Not in daylight savings time.

DS_status = '1': In daylight savings time.

DS_day_of_month — This 5-bit unsigned integer field indicates the local day of the month on which the transition into or out of daylight savings time is to occur (1–31).

DS_hour — This 8-bit unsigned integer field indicates the local hour at which the transition into or out of daylight savings time is to occur (0–18). This usually occurs at 2 a.m. in the U.S.

Table A2 Basic Use of Daylight Savings Fields Through the Year

Conditions	DS_status	DS_day_of_month	DS_hour
At the beginning of the year (January) daylight savings is off. This is the status of the fields until:	0	0	0
• When the transition into daylight savings time is within less than one month, the DS_day_of_month field takes the value day_in, and the DS_hour field takes the value hour_in. The DS_status bit is 0 indicating it is not yet daylight savings time. (The transition is to occur on the day_in day of the month at hour=hour_in; for example, if the transition were on April 15 at 2 a.m., then day_in=15 and hour_in=2)	0	day_in	hour_in
• After all time zone daylight transitions (within the span of the network) have occurred, the DS_status bit takes the value 1, indicating that daylight savings time is on. The DS_day_of_month field and the DS_hour field take the value 0. (In the U.S., this transition has to occur no later than 7 p.m. Pacific Time on the day day_in). This is the status of the fields until:	1	0	0
• When the transition out of daylight savings time is within less than one month, the DS_day_of_month field takes the value day_out, and the DS_hour field takes the value hour_out. The DS_status bit is 1 indicating it is still daylight savings time. (The transition is to occur on the day_out day of the month at hour=hour_out; for example, if the transition were on October 27 at 2 a.m., then day_out=27 and hour_out=2)	1	day_out	hour_out
• After all time zones (within the span of the network) have shifted out of daylight savings time, the DS_status bit takes the value 0, indicating that daylight savings time is off. The DS_day_of_month field and the DS_hour field take the value 0. (In the U.S., this transition has to occur no later than 7 p.m. Pacific Time on the day day_out). This finishes the cycle.	0	0	0

Annex B

Additional Constraints on Virtual Channel Table For the U.S.

(Normative)

1. ASSIGNMENT OF MAJOR CHANNEL NUMBER VALUES FOR TERRESTRIAL BROADCAST IN THE U.S.

The assignment of major_channel_number values in the U.S. is based on the rules below.

1. For broadcasters with existing NTSC licenses, the major_channel_number for the existing NTSC channels, as well as the digital virtual channels, controlled by the broadcaster, shall be set to the current NTSC RF channel number. E.g., assume a broadcaster who has an NTSC broadcast license for RF channel 13 is assigned RF channel 39 for digital ATSC broadcast. That broadcaster is required to use major_channel_number 13 for identification of the analog NTSC channel on RF channel 13, as well as the digital virtual channels it is controlling on RF channel 39.
2. For a new broadcaster without an existing NTSC license, the major_channel_number for the digital virtual channels controlled by the broadcaster shall be set to the FCC assigned RF channel number for ATSC digital TV broadcast. E.g., assume a broadcaster who currently has no NTSC broadcast license applies and receives a license for digital ATSC broadcast on RF channel 49. That broadcaster is required to use major_channel_number 49 for identification of the digital virtual channels that it is controlling on RF channel 49.
3. If during or at the end of the transition period, the RF channel assigned to a broadcaster for digital ATSC broadcast is changed for any reason, the major_channel_number used by that broadcaster shall not change.
4. If, after the transition, a previously used NTSC RF channel in a market is assigned to a newly-licensed DTV broadcaster in that market, the newly-licensed DTV broadcaster shall use, as his major_channel_number, the number of the current actual DTV RF channel being used by the previous NTSC licensee of the assigned channel.
5. If a broadcaster owns or controls broadcast licenses for two or more different RF channels having overlapping service areas, he may use a common major_channel_number for all services on all channels. He may choose the major_channel_number as determined above for any one of the RF channels. The values in the minor_channel_number fields must be partitioned to insure that there is no duplication of the two-part channel number in the DTV service area, including the overlapping DTV service areas of other broadcasters using that same major_channel_number.
6. The two-part channel numbers for other broadcasts may be included in the DTV transport stream, provided that the channel_TSID and source_id are exactly associated with the two-part channel number combinations used by the referenced broadcaster

and there is no duplication with those used by any broadcaster whose DTV service[14] area overlaps with the emitting station's DTV service[15] area.

7. A broadcaster may include in the transmitted multiplex programming originating from a different licensed broadcaster and use the major/minor channel numbers of the original broadcast if the major/minor channel number combinations are coordinated in the local broadcast area to avoid conflicts. The coordination process is beyond the scope of this document.
8. The provisions listed above assign major_channel_number values 2 through 69 uniquely to broadcasters licensed to broadcast Digital ATSC signals and guarantee that the two-part channel number combinations used by a broadcaster will be different from those used by any other broadcaster with an overlapping DTV service[16] area.
9. Values for major_channel_number from 70 to 99 may be used to identify groups of digital services carried in an ATSC multiplex that the broadcaster wishes to be identified by a different major channel number. Values 70 through 99 must be unique in each potential receiving location or the receiver will not be able to correctly select such services. For example a local broadcaster transmitting community college lectures in its bit stream may want to use a major_channel_number different than its own major_channel_number for the virtual channel carrying the lectures. The assessment of the feasibility of using this capability, as well as the coordination process for assignment of these major_channel_number values is beyond the scope of this document.
10. For a translated signal, the major/minor channel numbers shall remain the same as the original broadcast station unless the major channel conflicts with a broadcaster operating in the service area of the translator. In that case, the translator shall change the major number to a non-conflicting number.

2. REQUIREMENT TO TRANSMIT ANALOG TRANSMISSION SIGNAL ID

Broadcasters which reference an NTSC signal by inserting a channel_TSID in a VCT shall cause insertion of an analog Transmission Signal ID within the VBI of each referenced NTSC signal per [13]. Refer to Annex D Section 9 for a discussion of the use of the analog Transmission Signal ID.

[14] CFR 47 73.622(e)
[15] CFR 47 73.622(e)
[16] CFR 47 73.622(e)

Annex C

Standard Huffman Tables for Text Compression[17]

(Normative)

1. SCOPE

This Annex describes the compression method adopted for the transmission of English-language text strings in PSIP. The method distinguishes two types of text strings: titles and program descriptions. For each of these types, Huffman tables are defined based on 1st-order conditional probabilities. Section 3 defines standard Huffman encode and decode tables optimized for English-language text such as that typically found in program titles. Section 4 defines Huffman encode and decode tables optimized for English-language text such as that typically found in program descriptions. Receivers supporting the English language are expected to support decoding of text using either of these two standard Huffman compression tables.

The encode tables provide necessary and sufficient information to build the Huffman trees that need to be implemented for decoding. The decode tables described in Tables C5 and C7 are a particular mapping of those trees into a numerical array suitable for storage. This array can be easily implemented and used with the decoding algorithm. However, the user is free to design its own decoding tables as long as they follow the Huffman trees and rules defined in this Annex.

2. CHARACTER SET DEFINITION

This compression method supports the full ISO/IEC 8859-1 (Latin-1) character set, although only characters in the ASCII range (character codes 1 to 127) can be compressed. The following characters have special definitions:

Table C1 Characters with Special Definitions

Character	Value (Decimal)	Meaning
String Terminate (ASCII Null)	0	The *Terminate* character is used to terminate strings. The Terminate character is appended to the string in either compressed or uncompressed form. The first encoded character in a compressed string is encoded/decoded from the Terminate sub-tree. In other words, when encoding or decoding the first character in a compressed string, assume that the previous character was a Terminate character.
Order-1 Escape (ASCII ESC)	27	Used to escape from first-order context to uncompressed context. The character which follows the Escape character is uncompressed.

2.1 First Order Escape

The order-1 Huffman trees are *partial*, that is, codes are not defined for every possible character sequence. For example, the standard decode tables do not contain codes for the character

[17] Tables C4 through C7 are © 1997 General Instrument Corporation. Unlimited use in conjunction with this ATSC standard is granted on a royalty-free basis by General Instrument Corporation. All other rights are reserved.

sequence *qp*. When uncompressed text contains a character sequence which is not defined in the decode table, the order-1 escape character is used to escape back to the uncompressed context. Uncompressed symbols are coded as 8-bit ASCII (Latin I). For example, the character sequence *qpa* would be coded with *compressed q*, *compressed ESC*, *uncompressed p*, *compressed a*.

First-order escape rules for compressed strings:

- Any character which follows a first-order escape character is an uncompressed (8-bit) character. (Any character which follows an uncompressed escape character is compressed).
- Characters (128 .. 255) cannot be compressed.
- Any character which follows a character from the set (128 .. 255) is uncompressed.

2.2 Decode Table Data Structures

Decode tables have two sections:

- **Tree Root Offset List:** Provides the table offsets, in *bytes* from the start of the decode table, for the roots of the 128 first-order decode trees. The list is contained in bytes (0 .. 255) of the decode table, and is defined by the first "for" loop in Table C1.
- **Order-1 Decode Trees:** Each and every character in the range (0 .. 127) has a corresponding first-order decode tree. For example, if the previous character was "s", then the decoder would use the "s" first-order decode tree (decode tree #115) to decode the next character (ASCII "s" equals 115 decimal). These 128 decode trees are delimited by the second "for" loop in Table C2.

Decode tables have the following format:

Table C2 Decode Table Format

Syntax	No. of Bits	Format
decode_table() {		
for (i==0; i<128; i++) {		
byte_offset_of_char_i_tree_root	16	uimsbf
}		
for (i==0; i<128; i++) {		
character_i_order_1_tree()	8*M	
}		
}		

Note that even though the ISO Latin-1 character set supports up to 256 characters, only the first 128 characters may be represented in compressed form.

2.2.1 Tree Root Byte Offsets

byte_offset_of_character_i_tree_root — A 16-bit unsigned integer specifying the location, in bytes from the beginning of the decode table, of the root for the i^th^ character's order-1 tree.

2.2.2 Order-1 Decode Trees

Order-1 decode trees are binary trees. The roots of the decode trees are located at the table offsets specified in the tree root offset list. The left and right children of a given node are specified as *word* offsets from the root of the tree (a *word* is equivalent to two bytes).

Decode trees have the format given in Table C3.

Table C3 Decode Tree Format

Syntax	No. of Bits	Format
character_i_order_1_tree() {		
for (j==0; j<N; j++) {		
left_child_word_offset_or_char_leaf	8	uimsbf
right_child_word_offset_or_char_leaf	8	uimsbf
}		
}		

left_child_word_offset_or_char_leaf — An 8-bit unsigned integer number with the following interpretation: If the highest bit is cleared (i.e., bit 7 is zero), the number specifies the offset, in words, of the left child from the root of the order-1 decode tree; if the highest bit is set (bit 7 is one), the lower 7 bits give the code (e.g., in ASCII) for a leaf character.

right_child_word_offset_or_char_leaf — An 8-bit unsigned integer number with the following interpretation: If the highest bit is cleared (i.e., bit 7 is zero), the number specifies the offset, in words, of the right child from the root of the order-1 decode tree; if the highest bit is set (bit 7 is one), the lower 7 bits give the code (e.g., in ASCII) for a leaf character.

It can be seen from Figure F3 (Annex F) that each node (corresponding to one iteration of the for-loop) has a byte for the left child or character, and a byte for the right child or character.

Characters are *leaves* of the order-1 decode trees, and are differentiated from intermediate nodes by the byte's most significant bit. When the most significant bit is set, the byte is a character leaf. When the most significant bit is not set, the byte contains the tabular word offset of the child node.

3. STANDARD COMPRESSION TYPE 1 ENCODE/DECODE TABLES

The following encode/decode tables are optimized for English-language program title text. These tables correspond to multiple_string_structure() with compression_type value 0x01, and a mode equal to 0xFF.

Table C4 English-language Program Title Encode Table

Prior Symbol: 0	Symbol: 27	Code: 11001011
Prior Symbol: 0	Symbol: '$'	Code: 1100101011
Prior Symbol: 0	Symbol: '2'	Code: 011010010
Prior Symbol: 0	Symbol: '4'	Code: 1100101010
Prior Symbol: 0	Symbol: '7'	Code: 011010011
Prior Symbol: 0	Symbol: 'A'	Code: 0111
Prior Symbol: 0	Symbol: 'B'	Code: 1001
Prior Symbol: 0	Symbol: 'C'	Code: 1011
Prior Symbol: 0	Symbol: 'D'	Code: 11011
Prior Symbol: 0	Symbol: 'E'	Code: 10001
Prior Symbol: 0	Symbol: 'F'	Code: 11000
Prior Symbol: 0	Symbol: 'G'	Code: 11100
Prior Symbol: 0	Symbol: 'H'	Code: 11111
Prior Symbol: 0	Symbol: 'I'	Code: 10000
Prior Symbol: 0	Symbol: 'J'	Code: 01100
Prior Symbol: 0	Symbol: 'K'	Code: 1100110
Prior Symbol: 0	Symbol: 'L'	Code: 11101
Prior Symbol: 0	Symbol: 'M'	Code: 1010
Prior Symbol: 0	Symbol: 'N'	Code: 0011
Prior Symbol: 0	Symbol: 'O'	Code: 011011
Prior Symbol: 0	Symbol: 'P'	Code: 11110
Prior Symbol: 0	Symbol: 'Q'	Code: 01101000
Prior Symbol: 0	Symbol: 'R'	Code: 11010
Prior Symbol: 0	Symbol: 'S'	Code: 000
Prior Symbol: 0	Symbol: 'T'	Code: 010
Prior Symbol: 0	Symbol: 'U'	Code: 0110101
Prior Symbol: 0	Symbol: 'V'	Code: 1100111
Prior Symbol: 0	Symbol: 'W'	Code: 0010
Prior Symbol: 0	Symbol: 'Y'	Code: 1100100
Prior Symbol: 0	Symbol: 'Z'	Code: 110010100
Prior Symbol: 1	Symbol: 27	Code: 1
Prior Symbol: 2	Symbol: 27	Code: 1
Prior Symbol: 3	Symbol: 27	Code: 1
Prior Symbol: 4	Symbol: 27	Code: 1
Prior Symbol: 5	Symbol: 27	Code: 1
Prior Symbol: 6	Symbol: 27	Code: 1
Prior Symbol: 7	Symbol: 27	Code: 1
Prior Symbol: 8	Symbol: 27	Code: 1
Prior Symbol: 9	Symbol: 27	Code: 1
Prior Symbol: 10	Symbol: 27	Code: 1
Prior Symbol: 11	Symbol: 27	Code: 1
Prior Symbol: 12	Symbol: 27	Code: 1
Prior Symbol: 13	Symbol: 27	Code: 1
Prior Symbol: 14	Symbol: 27	Code: 1
Prior Symbol: 15	Symbol: 27	Code: 1
Prior Symbol: 16	Symbol: 27	Code: 1
Prior Symbol: 17	Symbol: 27	Code: 1
Prior Symbol: 18	Symbol: 27	Code: 1
Prior Symbol: 19	Symbol: 27	Code: 1
Prior Symbol: 20	Symbol: 27	Code: 1
Prior Symbol: 21	Symbol: 27	Code: 1
Prior Symbol: 22	Symbol: 27	Code: 1
Prior Symbol: 23	Symbol: 27	Code: 1
Prior Symbol: 24	Symbol: 27	Code: 1
Prior Symbol: 25	Symbol: 27	Code: 1
Prior Symbol: 26	Symbol: 27	Code: 1
Prior Symbol: 27	Symbol: 27	Code: 1
Prior Symbol: 28	Symbol: 27	Code: 1
Prior Symbol: 29	Symbol: 27	Code: 1
Prior Symbol: 30	Symbol: 27	Code: 1
Prior Symbol: 31	Symbol: 27	Code: 1
Prior Symbol: ' '	Symbol: 27	Code: 10010100
Prior Symbol: ' '	Symbol: '&'	Code: 010001
Prior Symbol: ' '	Symbol: '''	Code: 010000100
Prior Symbol: ' '	Symbol: '-'	Code: 00000001
Prior Symbol: ' '	Symbol: '1'	Code: 010000101
Prior Symbol: ' '	Symbol: '2'	Code: 00000010
Prior Symbol: ' '	Symbol: '3'	Code: 01000001
Prior Symbol: ' '	Symbol: '9'	Code: 000000000
Prior Symbol: ' '	Symbol: 'A'	Code: 10111
Prior Symbol: ' '	Symbol: 'B'	Code: 0010
Prior Symbol: ' '	Symbol: 'C'	Code: 1100
Prior Symbol: ' '	Symbol: 'D'	Code: 11100
Prior Symbol: ' '	Symbol: 'E'	Code: 011010
Prior Symbol: ' '	Symbol: 'F'	Code: 10011
Prior Symbol: ' '	Symbol: 'G'	Code: 00001
Prior Symbol: ' '	Symbol: 'H'	Code: 10101
Prior Symbol: ' '	Symbol: 'I'	Code: 111111
Prior Symbol: ' '	Symbol: 'J'	Code: 111110
Prior Symbol: ' '	Symbol: 'K'	Code: 010011
Prior Symbol: ' '	Symbol: 'L'	Code: 11110
Prior Symbol: ' '	Symbol: 'M'	Code: 0101
Prior Symbol: ' '	Symbol: 'N'	Code: 10110
Prior Symbol: ' '	Symbol: 'O'	Code: 011011
Prior Symbol: ' '	Symbol: 'P'	Code: 11101
Prior Symbol: ' '	Symbol: 'Q'	Code: 100100011
Prior Symbol: ' '	Symbol: 'R'	Code: 10100
Prior Symbol: ' '	Symbol: 'S'	Code: 1101
Prior Symbol: ' '	Symbol: 'T'	Code: 1000
Prior Symbol: ' '	Symbol: 'U'	Code: 1001001
Prior Symbol: ' '	Symbol: 'V'	Code: 1001011
Prior Symbol: ' '	Symbol: 'W'	Code: 0011
Prior Symbol: ' '	Symbol: 'X'	Code: 0000000010
Prior Symbol: ' '	Symbol: 'Y'	Code: 000001
Prior Symbol: ' '	Symbol: 'Z'	Code: 00000011
Prior Symbol: ' '	Symbol: 'a'	Code: 01100
Prior Symbol: ' '	Symbol: 'b'	Code: 10010101
Prior Symbol: ' '	Symbol: 'c'	Code: 01000000
Prior Symbol: ' '	Symbol: 'd'	Code: 01000011
Prior Symbol: ' '	Symbol: 'e'	Code: 0000000011
Prior Symbol: ' '	Symbol: 'f'	Code: 10010000
Prior Symbol: ' '	Symbol: 'i'	Code: 010010
Prior Symbol: ' '	Symbol: 'l'	Code: 100100010
Prior Symbol: ' '	Symbol: 'o'	Code: 0001
Prior Symbol: ' '	Symbol: 't'	Code: 0111
Prior Symbol: '!'	Symbol: 0	Code: 1
Prior Symbol: '!'	Symbol: 27	Code: 01
Prior Symbol: '!'	Symbol: ' '	Code: 00
Prior Symbol: '"'	Symbol: 27	Code: 1
Prior Symbol: '#'	Symbol: 27	Code: 1
Prior Symbol: '$'	Symbol: 27	Code: 1
Prior Symbol: '$'	Symbol: '1'	Code: 0
Prior Symbol: '%'	Symbol: 27	Code: 1
Prior Symbol: '&'	Symbol: 27	Code: 0
Prior Symbol: '&'	Symbol: ' '	Code: 1
Prior Symbol: '''	Symbol: 27	Code: 011
Prior Symbol: '''	Symbol: ' '	Code: 010
Prior Symbol: '''	Symbol: '9'	Code: 0001
Prior Symbol: '''	Symbol: 'd'	Code: 0000
Prior Symbol: '''	Symbol: 's'	Code: 1
Prior Symbol: '''	Symbol: 't'	Code: 001
Prior Symbol: '('	Symbol: 27	Code: 1
Prior Symbol: ')'	Symbol: 27	Code: 1
Prior Symbol: '*'	Symbol: 27	Code: 00
Prior Symbol: '*'	Symbol: 'A'	Code: 01
Prior Symbol: '*'	Symbol: 'H'	Code: 10
Prior Symbol: '*'	Symbol: 'S'	Code: 11
Prior Symbol: '+'	Symbol: 27	Code: 1
Prior Symbol: ','	Symbol: 27	Code: 0
Prior Symbol: ','	Symbol: ' '	Code: 1
Prior Symbol: '-'	Symbol: 27	Code: 01
Prior Symbol: '-'	Symbol: ' '	Code: 111
Prior Symbol: '-'	Symbol: '-'	Code: 1101
Prior Symbol: '-'	Symbol: '1'	Code: 1000
Prior Symbol: '-'	Symbol: 'A'	Code: 001
Prior Symbol: '-'	Symbol: 'M'	Code: 000
Prior Symbol: '-'	Symbol: 'R'	Code: 1001
Prior Symbol: '-'	Symbol: 'S'	Code: 1010
Prior Symbol: '-'	Symbol: 'T'	Code: 1011
Prior Symbol: '-'	Symbol: 'U'	Code: 1100
Prior Symbol: '.'	Symbol: 0	Code: 111
Prior Symbol: '.'	Symbol: 27	Code: 101
Prior Symbol: '.'	Symbol: ' '	Code: 0
Prior Symbol: '.'	Symbol: '.'	Code: 110
Prior Symbol: '.'	Symbol: 'I'	Code: 10010
Prior Symbol: '.'	Symbol: 'S'	Code: 1000
Prior Symbol: '.'	Symbol: 'W'	Code: 10011
Prior Symbol: '/'	Symbol: 27	Code: 1
Prior Symbol: '0'	Symbol: 0	Code: 01
Prior Symbol: '0'	Symbol: 27	Code: 001
Prior Symbol: '0'	Symbol: ' '	Code: 10
Prior Symbol: '0'	Symbol: '-'	Code: 000
Prior Symbol: '0'	Symbol: '0'	Code: 11
Prior Symbol: '1'	Symbol: 0	Code: 010
Prior Symbol: '1'	Symbol: 27	Code: 011
Prior Symbol: '1'	Symbol: ' '	Code: 110
Prior Symbol: '1'	Symbol: '0'	Code: 111
Prior Symbol: '1'	Symbol: '1'	Code: 100
Prior Symbol: '1'	Symbol: '2'	Code: 101
Prior Symbol: '1'	Symbol: '9'	Code: 00
Prior Symbol: '2'	Symbol: 0	Code: 11
Prior Symbol: '2'	Symbol: 27	Code: 10
Prior Symbol: '2'	Symbol: '0'	Code: 01
Prior Symbol: '2'	Symbol: '1'	Code: 000
Prior Symbol: '2'	Symbol: ':'	Code: 001
Prior Symbol: '3'	Symbol: 0	Code: 0
Prior Symbol: '3'	Symbol: 27	Code: 11
Prior Symbol: '3'	Symbol: '0'	Code: 10
Prior Symbol: '4'	Symbol: 27	Code: 0
Prior Symbol: '4'	Symbol: '8'	Code: 1
Prior Symbol: '5'	Symbol: 27	Code: 1
Prior Symbol: '6'	Symbol: 27	Code: 1
Prior Symbol: '7'	Symbol: 27	Code: 0
Prior Symbol: '7'	Symbol: '0'	Code: 1
Prior Symbol: '8'	Symbol: 27	Code: 0
Prior Symbol: '8'	Symbol: ' '	Code: 1
Prior Symbol: '9'	Symbol: 27	Code: 11
Prior Symbol: '9'	Symbol: '0'	Code: 01
Prior Symbol: '9'	Symbol: '1'	Code: 100
Prior Symbol: '9'	Symbol: '3'	Code: 101
Prior Symbol: '9'	Symbol: '9'	Code: 00
Prior Symbol: ':'	Symbol: 27	Code: 0
Prior Symbol: ':'	Symbol: ' '	Code: 1
Prior Symbol: ';'	Symbol: 27	Code: 1
Prior Symbol: '<'	Symbol: 27	Code: 1
Prior Symbol: '='	Symbol: 27	Code: 1
Prior Symbol: '>'	Symbol: 27	Code: 1
Prior Symbol: '?'	Symbol: 0	Code: 1
Prior Symbol: '?'	Symbol: 27	Code: 0
Prior Symbol: '@'	Symbol: 27	Code: 1
Prior Symbol: 'A'	Symbol: 27	Code: 00010
Prior Symbol: 'A'	Symbol: ' '	Code: 010
Prior Symbol: 'A'	Symbol: '*'	Code: 1101000
Prior Symbol: 'A'	Symbol: '-'	Code: 1101001
Prior Symbol: 'A'	Symbol: '.'	Code: 1101010
Prior Symbol: 'A'	Symbol: 'B'	Code: 110110
Prior Symbol: 'A'	Symbol: 'b'	Code: 110010
Prior Symbol: 'A'	Symbol: 'c'	Code: 01100
Prior Symbol: 'A'	Symbol: 'd'	Code: 001
Prior Symbol: 'A'	Symbol: 'f'	Code: 01101
Prior Symbol: 'A'	Symbol: 'g'	Code: 011110
Prior Symbol: 'A'	Symbol: 'i'	Code: 110011
Prior Symbol: 'A'	Symbol: 'l'	Code: 100
Prior Symbol: 'A'	Symbol: 'm'	Code: 111

Prior Symbol: 'A' Symbol: 'n' Code: 101
Prior Symbol: 'A' Symbol: 'p' Code: 110111
Prior Symbol: 'A' Symbol: 'r' Code: 0000
Prior Symbol: 'A' Symbol: 's' Code: 00011
Prior Symbol: 'A' Symbol: 't' Code: 011111
Prior Symbol: 'A' Symbol: 'u' Code: 11000
Prior Symbol: 'A' Symbol: 'v' Code: 1101011
Prior Symbol: 'A' Symbol: 'w' Code: 01110
Prior Symbol: 'B' Symbol: 27 Code: 00010
Prior Symbol: 'B' Symbol: 'A' Code: 000110
Prior Symbol: 'B' Symbol: 'C' Code: 0000
Prior Symbol: 'B' Symbol: 'S' Code: 000111
Prior Symbol: 'B' Symbol: 'a' Code: 111
Prior Symbol: 'B' Symbol: 'e' Code: 01
Prior Symbol: 'B' Symbol: 'i' Code: 1010
Prior Symbol: 'B' Symbol: 'l' Code: 1011
Prior Symbol: 'B' Symbol: 'o' Code: 110
Prior Symbol: 'B' Symbol: 'r' Code: 001
Prior Symbol: 'B' Symbol: 'u' Code: 100
Prior Symbol: 'C' Symbol: 27 Code: 00101
Prior Symbol: 'C' Symbol: ' ' Code: 10110
Prior Symbol: 'C' Symbol: 'A' Code: 0011100
Prior Symbol: 'C' Symbol: 'B' Code: 001111
Prior Symbol: 'C' Symbol: 'O' Code: 101110
Prior Symbol: 'C' Symbol: 'a' Code: 100
Prior Symbol: 'C' Symbol: 'e' Code: 101111
Prior Symbol: 'C' Symbol: 'h' Code: 01
Prior Symbol: 'C' Symbol: 'i' Code: 00110
Prior Symbol: 'C' Symbol: 'l' Code: 000
Prior Symbol: 'C' Symbol: 'o' Code: 11
Prior Symbol: 'C' Symbol: 'r' Code: 1010
Prior Symbol: 'C' Symbol: 'u' Code: 00100
Prior Symbol: 'C' Symbol: 'y' Code: 0011101
Prior Symbol: 'D' Symbol: 27 Code: 01001
Prior Symbol: 'D' Symbol: 'a' Code: 10
Prior Symbol: 'D' Symbol: 'e' Code: 111
Prior Symbol: 'D' Symbol: 'i' Code: 110
Prior Symbol: 'D' Symbol: 'o' Code: 00
Prior Symbol: 'D' Symbol: 'r' Code: 011
Prior Symbol: 'D' Symbol: 'u' Code: 0101
Prior Symbol: 'D' Symbol: 'y' Code: 01000
Prior Symbol: 'E' Symbol: 27 Code: 011
Prior Symbol: 'E' Symbol: 'C' Code: 1010
Prior Symbol: 'E' Symbol: 'a' Code: 111
Prior Symbol: 'E' Symbol: 'd' Code: 000
Prior Symbol: 'E' Symbol: 'l' Code: 1100
Prior Symbol: 'E' Symbol: 'm' Code: 0100
Prior Symbol: 'E' Symbol: 'n' Code: 1101
Prior Symbol: 'E' Symbol: 'q' Code: 101110
Prior Symbol: 'E' Symbol: 's' Code: 10110
Prior Symbol: 'E' Symbol: 'u' Code: 101111
Prior Symbol: 'E' Symbol: 'v' Code: 100
Prior Symbol: 'E' Symbol: 'x' Code: 001
Prior Symbol: 'E' Symbol: 'y' Code: 0101
Prior Symbol: 'F' Symbol: 27 Code: 011111
Prior Symbol: 'F' Symbol: ' ' Code: 011110
Prior Symbol: 'F' Symbol: 'L' Code: 01110
Prior Symbol: 'F' Symbol: 'a' Code: 10
Prior Symbol: 'F' Symbol: 'e' Code: 0110
Prior Symbol: 'F' Symbol: 'i' Code: 110
Prior Symbol: 'F' Symbol: 'l' Code: 000
Prior Symbol: 'F' Symbol: 'o' Code: 010
Prior Symbol: 'F' Symbol: 'r' Code: 111
Prior Symbol: 'F' Symbol: 'u' Code: 001
Prior Symbol: 'G' Symbol: 27 Code: 10110
Prior Symbol: 'G' Symbol: '.' Code: 101010
Prior Symbol: 'G' Symbol: 'A' Code: 101111
Prior Symbol: 'G' Symbol: 'a' Code: 1110
Prior Symbol: 'G' Symbol: 'e' Code: 110
Prior Symbol: 'G' Symbol: 'h' Code: 10100
Prior Symbol: 'G' Symbol: 'i' Code: 100
Prior Symbol: 'G' Symbol: 'l' Code: 101011
Prior Symbol: 'G' Symbol: 'o' Code: 01
Prior Symbol: 'G' Symbol: 'r' Code: 00
Prior Symbol: 'G' Symbol: 'u' Code: 1111
Prior Symbol: 'G' Symbol: 'y' Code: 101110
Prior Symbol: 'H' Symbol: 0 Code: 111010
Prior Symbol: 'H' Symbol: 27 Code: 111011
Prior Symbol: 'H' Symbol: 'a' Code: 110
Prior Symbol: 'H' Symbol: 'e' Code: 10
Prior Symbol: 'H' Symbol: 'i' Code: 1111
Prior Symbol: 'H' Symbol: 'o' Code: 0
Prior Symbol: 'H' Symbol: 'u' Code: 11100
Prior Symbol: 'I' Symbol: 0 Code: 1000
Prior Symbol: 'I' Symbol: 27 Code: 1001
Prior Symbol: 'I' Symbol: ' ' Code: 11110
Prior Symbol: 'I' Symbol: '.' Code: 111110
Prior Symbol: 'I' Symbol: ':' Code: 101110
Prior Symbol: 'I' Symbol: 'I' Code: 1100
Prior Symbol: 'I' Symbol: 'T' Code: 101111
Prior Symbol: 'I' Symbol: 'c' Code: 10110
Prior Symbol: 'I' Symbol: 'm' Code: 1010
Prior Symbol: 'I' Symbol: 'n' Code: 0
Prior Symbol: 'I' Symbol: 'r' Code: 111111
Prior Symbol: 'I' Symbol: 's' Code: 1101
Prior Symbol: 'I' Symbol: 't' Code: 1110
Prior Symbol: 'J' Symbol: 27 Code: 000
Prior Symbol: 'J' Symbol: 'a' Code: 01
Prior Symbol: 'J' Symbol: 'e' Code: 11
Prior Symbol: 'J' Symbol: 'o' Code: 10
Prior Symbol: 'J' Symbol: 'u' Code: 001
Prior Symbol: 'K' Symbol: 27 Code: 000
Prior Symbol: 'K' Symbol: 'a' Code: 0100
Prior Symbol: 'K' Symbol: 'e' Code: 001
Prior Symbol: 'K' Symbol: 'i' Code: 1
Prior Symbol: 'K' Symbol: 'n' Code: 0111
Prior Symbol: 'K' Symbol: 'o' Code: 0101
Prior Symbol: 'K' Symbol: 'u' Code: 0110
Prior Symbol: 'L' Symbol: 27 Code: 01001
Prior Symbol: 'L' Symbol: ' ' Code: 01000
Prior Symbol: 'L' Symbol: 'a' Code: 10
Prior Symbol: 'L' Symbol: 'e' Code: 011
Prior Symbol: 'L' Symbol: 'i' Code: 11
Prior Symbol: 'L' Symbol: 'o' Code: 00
Prior Symbol: 'L' Symbol: 'u' Code: 0101
Prior Symbol: 'M' Symbol: 27 Code: 1011111
Prior Symbol: 'M' Symbol: '*' Code: 10111100
Prior Symbol: 'M' Symbol: 'T' Code: 10111101
Prior Symbol: 'M' Symbol: 'a' Code: 11
Prior Symbol: 'M' Symbol: 'c' Code: 101110
Prior Symbol: 'M' Symbol: 'e' Code: 1010
Prior Symbol: 'M' Symbol: 'i' Code: 100
Prior Symbol: 'M' Symbol: 'o' Code: 00
Prior Symbol: 'M' Symbol: 'r' Code: 10110
Prior Symbol: 'M' Symbol: 'u' Code: 010
Prior Symbol: 'M' Symbol: 'y' Code: 011
Prior Symbol: 'N' Symbol: 27 Code: 1000
Prior Symbol: 'N' Symbol: ' ' Code: 110001
Prior Symbol: 'N' Symbol: 'B' Code: 1001
Prior Symbol: 'N' Symbol: 'F' Code: 110010
Prior Symbol: 'N' Symbol: 'N' Code: 110000
Prior Symbol: 'N' Symbol: 'a' Code: 1101
Prior Symbol: 'N' Symbol: 'e' Code: 0
Prior Symbol: 'N' Symbol: 'i' Code: 111
Prior Symbol: 'N' Symbol: 'o' Code: 101
Prior Symbol: 'N' Symbol: 'u' Code: 110011
Prior Symbol: 'O' Symbol: 27 Code: 010
Prior Symbol: 'O' Symbol: ' ' Code: 001
Prior Symbol: 'O' Symbol: 'd' Code: 01110
Prior Symbol: 'O' Symbol: 'f' Code: 11010
Prior Symbol: 'O' Symbol: 'l' Code: 1100
Prior Symbol: 'O' Symbol: 'n' Code: 10
Prior Symbol: 'O' Symbol: 'p' Code: 0001
Prior Symbol: 'O' Symbol: 'r' Code: 0110
Prior Symbol: 'O' Symbol: 's' Code: 01111
Prior Symbol: 'O' Symbol: 'u' Code: 111
Prior Symbol: 'O' Symbol: 'v' Code: 11011
Prior Symbol: 'O' Symbol: 'w' Code: 0000
Prior Symbol: 'P' Symbol: 27 Code: 111111
Prior Symbol: 'P' Symbol: ' ' Code: 1111100
Prior Symbol: 'P' Symbol: '.' Code: 011001
Prior Symbol: 'P' Symbol: 'G' Code: 111101
Prior Symbol: 'P' Symbol: 'R' Code: 111100
Prior Symbol: 'P' Symbol: 'a' Code: 00
Prior Symbol: 'P' Symbol: 'e' Code: 010
Prior Symbol: 'P' Symbol: 'i' Code: 0111
Prior Symbol: 'P' Symbol: 'l' Code: 1110
Prior Symbol: 'P' Symbol: 'o' Code: 110
Prior Symbol: 'P' Symbol: 'r' Code: 10
Prior Symbol: 'P' Symbol: 's' Code: 1111101
Prior Symbol: 'P' Symbol: 'u' Code: 01101
Prior Symbol: 'P' Symbol: 'y' Code: 011000
Prior Symbol: 'Q' Symbol: 27 Code: 00
Prior Symbol: 'Q' Symbol: 'V' Code: 01
Prior Symbol: 'Q' Symbol: 'u' Code: 1
Prior Symbol: 'R' Symbol: 27 Code: 10001
Prior Symbol: 'R' Symbol: 'a' Code: 101
Prior Symbol: 'R' Symbol: 'e' Code: 11
Prior Symbol: 'R' Symbol: 'h' Code: 10000
Prior Symbol: 'R' Symbol: 'i' Code: 00
Prior Symbol: 'R' Symbol: 'o' Code: 01
Prior Symbol: 'R' Symbol: 'u' Code: 1001
Prior Symbol: 'S' Symbol: 27 Code: 101110
Prior Symbol: 'S' Symbol: ' ' Code: 1110100
Prior Symbol: 'S' Symbol: '*' Code: 1011000
Prior Symbol: 'S' Symbol: '.' Code: 1011011
Prior Symbol: 'S' Symbol: 'a' Code: 1111
Prior Symbol: 'S' Symbol: 'c' Code: 11100
Prior Symbol: 'S' Symbol: 'e' Code: 000
Prior Symbol: 'S' Symbol: 'h' Code: 100
Prior Symbol: 'S' Symbol: 'i' Code: 1100
Prior Symbol: 'S' Symbol: 'k' Code: 101111
Prior Symbol: 'S' Symbol: 'l' Code: 1011001
Prior Symbol: 'S' Symbol: 'm' Code: 1110110
Prior Symbol: 'S' Symbol: 'n' Code: 1110111
Prior Symbol: 'S' Symbol: 'o' Code: 1010
Prior Symbol: 'S' Symbol: 'p' Code: 001
Prior Symbol: 'S' Symbol: 'q' Code: 1011010
Prior Symbol: 'S' Symbol: 't' Code: 01
Prior Symbol: 'S' Symbol: 'u' Code: 1101
Prior Symbol: 'S' Symbol: 'w' Code: 1110101
Prior Symbol: 'T' Symbol: 27 Code: 1111010
Prior Symbol: 'T' Symbol: '-' Code: 11110110
Prior Symbol: 'T' Symbol: 'N' Code: 11110111
Prior Symbol: 'T' Symbol: 'V' Code: 111100
Prior Symbol: 'T' Symbol: 'a' Code: 1010
Prior Symbol: 'T' Symbol: 'e' Code: 1011
Prior Symbol: 'T' Symbol: 'h' Code: 0
Prior Symbol: 'T' Symbol: 'i' Code: 1110
Prior Symbol: 'T' Symbol: 'o' Code: 110
Prior Symbol: 'T' Symbol: 'r' Code: 100
Prior Symbol: 'T' Symbol: 'u' Code: 111110
Prior Symbol: 'T' Symbol: 'w' Code: 111111
Prior Symbol: 'U' Symbol: 27 Code: 101
Prior Symbol: 'U' Symbol: '.' Code: 1001
Prior Symbol: 'U' Symbol: 'l' Code: 1000
Prior Symbol: 'U' Symbol: 'n' Code: 0
Prior Symbol: 'U' Symbol: 'p' Code: 11
Prior Symbol: 'V' Symbol: 0 Code: 000
Prior Symbol: 'V' Symbol: 27 Code: 0011
Prior Symbol: 'V' Symbol: ' ' Code: 01010
Prior Symbol: 'V' Symbol: 'C' Code: 01011
Prior Symbol: 'V' Symbol: 'a' Code: 011
Prior Symbol: 'V' Symbol: 'e' Code: 0100
Prior Symbol: 'V' Symbol: 'i' Code: 1
Prior Symbol: 'V' Symbol: 'o' Code: 0010
Prior Symbol: 'W' Symbol: 27 Code: 00011
Prior Symbol: 'W' Symbol: 'F' Code: 000100
Prior Symbol: 'W' Symbol: 'W' Code: 000101
Prior Symbol: 'W' Symbol: 'a' Code: 111
Prior Symbol: 'W' Symbol: 'e' Code: 110
Prior Symbol: 'W' Symbol: 'h' Code: 001
Prior Symbol: 'W' Symbol: 'i' Code: 01
Prior Symbol: 'W' Symbol: 'o' Code: 10
Prior Symbol: 'W' Symbol: 'r' Code: 0000
Prior Symbol: 'X' Symbol: 27 Code: 1
Prior Symbol: 'Y' Symbol: 27 Code: 001
Prior Symbol: 'Y' Symbol: 'a' Code: 000
Prior Symbol: 'Y' Symbol: 'e' Code: 01
Prior Symbol: 'Y' Symbol: 'o' Code: 1
Prior Symbol: 'Z' Symbol: 27 Code: 00
Prior Symbol: 'Z' Symbol: 'a' Code: 01
Prior Symbol: 'Z' Symbol: 'o' Code: 1
Prior Symbol: '[' Symbol: 27 Code: 1
Prior Symbol: '\' Symbol: 27 Code: 1
Prior Symbol: ']' Symbol: 27 Code: 1
Prior Symbol: '^' Symbol: 27 Code: 1
Prior Symbol: '_' Symbol: 27 Code: 1
Prior Symbol: '`' Symbol: 27 Code: 1
Prior Symbol: 'a' Symbol: 0 Code: 00010

Prior Symbol: 'a' Symbol: 27 Code: 1111010110
Prior Symbol: 'a' Symbol: ' ' Code: 10110
Prior Symbol: 'a' Symbol: ''' Code: 11110100
Prior Symbol: 'a' Symbol: ':' Code: 1111010111
Prior Symbol: 'a' Symbol: 'b' Code: 010010
Prior Symbol: 'a' Symbol: 'c' Code: 11111
Prior Symbol: 'a' Symbol: 'd' Code: 10100
Prior Symbol: 'a' Symbol: 'e' Code: 101011000
Prior Symbol: 'a' Symbol: 'f' Code: 10101101
Prior Symbol: 'a' Symbol: 'g' Code: 01000
Prior Symbol: 'a' Symbol: 'h' Code: 0100111
Prior Symbol: 'a' Symbol: 'i' Code: 10111
Prior Symbol: 'a' Symbol: 'j' Code: 101011001
Prior Symbol: 'a' Symbol: 'k' Code: 101010
Prior Symbol: 'a' Symbol: 'l' Code: 001
Prior Symbol: 'a' Symbol: 'm' Code: 0101
Prior Symbol: 'a' Symbol: 'n' Code: 110
Prior Symbol: 'a' Symbol: 'p' Code: 111100
Prior Symbol: 'a' Symbol: 'r' Code: 100
Prior Symbol: 'a' Symbol: 's' Code: 1110
Prior Symbol: 'a' Symbol: 't' Code: 011
Prior Symbol: 'a' Symbol: 'u' Code: 1111011
Prior Symbol: 'a' Symbol: 'v' Code: 00011
Prior Symbol: 'a' Symbol: 'w' Code: 1010111
Prior Symbol: 'a' Symbol: 'x' Code: 111101010
Prior Symbol: 'a' Symbol: 'y' Code: 0000
Prior Symbol: 'a' Symbol: 'z' Code: 0100110
Prior Symbol: 'b' Symbol: 0 Code: 11111
Prior Symbol: 'b' Symbol: 27 Code: 111101
Prior Symbol: 'b' Symbol: ' ' Code: 0110
Prior Symbol: 'b' Symbol: 'a' Code: 00
Prior Symbol: 'b' Symbol: 'b' Code: 01111
Prior Symbol: 'b' Symbol: 'e' Code: 1010
Prior Symbol: 'b' Symbol: 'i' Code: 1110
Prior Symbol: 'b' Symbol: 'l' Code: 010
Prior Symbol: 'b' Symbol: 'o' Code: 110
Prior Symbol: 'b' Symbol: 'r' Code: 1011
Prior Symbol: 'b' Symbol: 's' Code: 111100
Prior Symbol: 'b' Symbol: 'u' Code: 01110
Prior Symbol: 'b' Symbol: 'y' Code: 100
Prior Symbol: 'c' Symbol: 0 Code: 010110
Prior Symbol: 'c' Symbol: 27 Code: 1000011
Prior Symbol: 'c' Symbol: ' ' Code: 0100
Prior Symbol: 'c' Symbol: 'C' Code: 0010110
Prior Symbol: 'c' Symbol: 'G' Code: 1000010
Prior Symbol: 'c' Symbol: 'L' Code: 0010111
Prior Symbol: 'c' Symbol: 'a' Code: 011
Prior Symbol: 'c' Symbol: 'c' Code: 001010
Prior Symbol: 'c' Symbol: 'e' Code: 111
Prior Symbol: 'c' Symbol: 'h' Code: 101
Prior Symbol: 'c' Symbol: 'i' Code: 0011
Prior Symbol: 'c' Symbol: 'k' Code: 110
Prior Symbol: 'c' Symbol: 'l' Code: 010111
Prior Symbol: 'c' Symbol: 'o' Code: 1001
Prior Symbol: 'c' Symbol: 'r' Code: 10001
Prior Symbol: 'c' Symbol: 's' Code: 00100
Prior Symbol: 'c' Symbol: 't' Code: 000
Prior Symbol: 'c' Symbol: 'u' Code: 01010
Prior Symbol: 'c' Symbol: 'y' Code: 100000
Prior Symbol: 'd' Symbol: 0 Code: 011
Prior Symbol: 'd' Symbol: 27 Code: 101110
Prior Symbol: 'd' Symbol: ' ' Code: 11
Prior Symbol: 'd' Symbol: '.' Code: 101101110
Prior Symbol: 'd' Symbol: 'a' Code: 1010
Prior Symbol: 'd' Symbol: 'd' Code: 100000
Prior Symbol: 'd' Symbol: 'e' Code: 00
Prior Symbol: 'd' Symbol: 'g' Code: 100001
Prior Symbol: 'd' Symbol: 'i' Code: 1001
Prior Symbol: 'd' Symbol: 'l' Code: 1011010
Prior Symbol: 'd' Symbol: 'o' Code: 101111
Prior Symbol: 'd' Symbol: 'r' Code: 101100
Prior Symbol: 'd' Symbol: 's' Code: 0101
Prior Symbol: 'd' Symbol: 'u' Code: 101101111
Prior Symbol: 'd' Symbol: 'v' Code: 10001
Prior Symbol: 'd' Symbol: 'w' Code: 10110110
Prior Symbol: 'd' Symbol: 'y' Code: 0100
Prior Symbol: 'e' Symbol: 0 Code: 001
Prior Symbol: 'e' Symbol: 27 Code: 1010111100
Prior Symbol: 'e' Symbol: ' ' Code: 01
Prior Symbol: 'e' Symbol: '!' Code: 1010111101
Prior Symbol: 'e' Symbol: ''' Code: 10101100
Prior Symbol: 'e' Symbol: '-' Code: 1010111110
Prior Symbol: 'e' Symbol: ':' Code: 00010010
Prior Symbol: 'e' Symbol: 'a' Code: 1000
Prior Symbol: 'e' Symbol: 'b' Code: 10101101
Prior Symbol: 'e' Symbol: 'c' Code: 100111
Prior Symbol: 'e' Symbol: 'd' Code: 00011
Prior Symbol: 'e' Symbol: 'e' Code: 10100
Prior Symbol: 'e' Symbol: 'f' Code: 1001100
Prior Symbol: 'e' Symbol: 'g' Code: 1010100
Prior Symbol: 'e' Symbol: 'h' Code: 1010111111
Prior Symbol: 'e' Symbol: 'i' Code: 10101110
Prior Symbol: 'e' Symbol: 'j' Code: 000100000
Prior Symbol: 'e' Symbol: 'k' Code: 1010101
Prior Symbol: 'e' Symbol: 'l' Code: 10010
Prior Symbol: 'e' Symbol: 'm' Code: 1001101
Prior Symbol: 'e' Symbol: 'n' Code: 1110
Prior Symbol: 'e' Symbol: 'o' Code: 000101
Prior Symbol: 'e' Symbol: 'p' Code: 000001
Prior Symbol: 'e' Symbol: 'q' Code: 000100001
Prior Symbol: 'e' Symbol: 'r' Code: 110
Prior Symbol: 'e' Symbol: 's' Code: 1111
Prior Symbol: 'e' Symbol: 't' Code: 10110
Prior Symbol: 'e' Symbol: 'u' Code: 000100010
Prior Symbol: 'e' Symbol: 'v' Code: 000000
Prior Symbol: 'e' Symbol: 'w' Code: 10111
Prior Symbol: 'e' Symbol: 'x' Code: 00010011
Prior Symbol: 'e' Symbol: 'y' Code: 00001
Prior Symbol: 'e' Symbol: 'z' Code: 000100011
Prior Symbol: 'f' Symbol: 0 Code: 11100
Prior Symbol: 'f' Symbol: 27 Code: 1111001
Prior Symbol: 'f' Symbol: ' ' Code: 0
Prior Symbol: 'f' Symbol: 'a' Code: 11101
Prior Symbol: 'f' Symbol: 'e' Code: 110
Prior Symbol: 'f' Symbol: 'f' Code: 1011
Prior Symbol: 'f' Symbol: 'i' Code: 1001
Prior Symbol: 'f' Symbol: 'l' Code: 111101
Prior Symbol: 'f' Symbol: 'o' Code: 1010
Prior Symbol: 'f' Symbol: 'r' Code: 111111
Prior Symbol: 'f' Symbol: 's' Code: 111110
Prior Symbol: 'f' Symbol: 't' Code: 1000
Prior Symbol: 'f' Symbol: 'u' Code: 1111000
Prior Symbol: 'g' Symbol: 0 Code: 110
Prior Symbol: 'g' Symbol: 27 Code: 1110000
Prior Symbol: 'g' Symbol: ' ' Code: 01
Prior Symbol: 'g' Symbol: ''' Code: 1001100
Prior Symbol: 'g' Symbol: ':' Code: 11100010
Prior Symbol: 'g' Symbol: 'a' Code: 1000
Prior Symbol: 'g' Symbol: 'e' Code: 101
Prior Symbol: 'g' Symbol: 'g' Code: 1111010
Prior Symbol: 'g' Symbol: 'h' Code: 00
Prior Symbol: 'g' Symbol: 'i' Code: 11101
Prior Symbol: 'g' Symbol: 'l' Code: 1111011
Prior Symbol: 'g' Symbol: 'n' Code: 100111
Prior Symbol: 'g' Symbol: 'o' Code: 111001
Prior Symbol: 'g' Symbol: 'r' Code: 10010
Prior Symbol: 'g' Symbol: 's' Code: 11111
Prior Symbol: 'g' Symbol: 't' Code: 1001101
Prior Symbol: 'g' Symbol: 'u' Code: 111100
Prior Symbol: 'g' Symbol: 'y' Code: 11100011
Prior Symbol: 'h' Symbol: 0 Code: 11101
Prior Symbol: 'h' Symbol: 27 Code: 1110001
Prior Symbol: 'h' Symbol: ' ' Code: 1011
Prior Symbol: 'h' Symbol: 'a' Code: 1100
Prior Symbol: 'h' Symbol: 'b' Code: 11100110
Prior Symbol: 'h' Symbol: 'e' Code: 0
Prior Symbol: 'h' Symbol: 'i' Code: 100
Prior Symbol: 'h' Symbol: 'l' Code: 1110010
Prior Symbol: 'h' Symbol: 'n' Code: 101001
Prior Symbol: 'h' Symbol: 'o' Code: 1101
Prior Symbol: 'h' Symbol: 'r' Code: 10101
Prior Symbol: 'h' Symbol: 't' Code: 1111
Prior Symbol: 'h' Symbol: 'u' Code: 11100111
Prior Symbol: 'h' Symbol: 'w' Code: 1110000
Prior Symbol: 'h' Symbol: 'y' Code: 101000
Prior Symbol: 'i' Symbol: 0 Code: 00110101
Prior Symbol: 'i' Symbol: 27 Code: 00110110
Prior Symbol: 'i' Symbol: ' ' Code: 000100
Prior Symbol: 'i' Symbol: '!' Code: 001101000
Prior Symbol: 'i' Symbol: 'a' Code: 00011
Prior Symbol: 'i' Symbol: 'b' Code: 0011000
Prior Symbol: 'i' Symbol: 'c' Code: 1111
Prior Symbol: 'i' Symbol: 'd' Code: 0010
Prior Symbol: 'i' Symbol: 'e' Code: 1101
Prior Symbol: 'i' Symbol: 'f' Code: 00111
Prior Symbol: 'i' Symbol: 'g' Code: 1100
Prior Symbol: 'i' Symbol: 'i' Code: 00110010
Prior Symbol: 'i' Symbol: 'k' Code: 00110011
Prior Symbol: 'i' Symbol: 'l' Code: 0110
Prior Symbol: 'i' Symbol: 'm' Code: 11101
Prior Symbol: 'i' Symbol: 'n' Code: 10
Prior Symbol: 'i' Symbol: 'o' Code: 0100
Prior Symbol: 'i' Symbol: 'p' Code: 000101
Prior Symbol: 'i' Symbol: 'r' Code: 11100
Prior Symbol: 'i' Symbol: 's' Code: 0111
Prior Symbol: 'i' Symbol: 't' Code: 0101
Prior Symbol: 'i' Symbol: 'v' Code: 0000
Prior Symbol: 'i' Symbol: 'x' Code: 001101001
Prior Symbol: 'i' Symbol: 'z' Code: 00110111
Prior Symbol: 'j' Symbol: 27 Code: 10
Prior Symbol: 'j' Symbol: 'a' Code: 11
Prior Symbol: 'j' Symbol: 'o' Code: 0
Prior Symbol: 'k' Symbol: 0 Code: 01
Prior Symbol: 'k' Symbol: 27 Code: 00011
Prior Symbol: 'k' Symbol: ' ' Code: 111
Prior Symbol: 'k' Symbol: ':' Code: 00001
Prior Symbol: 'k' Symbol: 'T' Code: 000000
Prior Symbol: 'k' Symbol: 'a' Code: 001111
Prior Symbol: 'k' Symbol: 'e' Code: 10
Prior Symbol: 'k' Symbol: 'f' Code: 000100
Prior Symbol: 'k' Symbol: 'i' Code: 110
Prior Symbol: 'k' Symbol: 'l' Code: 000101
Prior Symbol: 'k' Symbol: 'o' Code: 000001
Prior Symbol: 'k' Symbol: 's' Code: 0010
Prior Symbol: 'k' Symbol: 'w' Code: 001110
Prior Symbol: 'k' Symbol: 'y' Code: 00110
Prior Symbol: 'l' Symbol: 0 Code: 1000
Prior Symbol: 'l' Symbol: 27 Code: 0111001
Prior Symbol: 'l' Symbol: ' ' Code: 010
Prior Symbol: 'l' Symbol: ''' Code: 01100010
Prior Symbol: 'l' Symbol: '-' Code: 11110011
Prior Symbol: 'l' Symbol: ':' Code: 01100011
Prior Symbol: 'l' Symbol: 'a' Code: 1110
Prior Symbol: 'l' Symbol: 'b' Code: 0110000
Prior Symbol: 'l' Symbol: 'c' Code: 01110000
Prior Symbol: 'l' Symbol: 'd' Code: 000
Prior Symbol: 'l' Symbol: 'e' Code: 110
Prior Symbol: 'l' Symbol: 'f' Code: 1111000
Prior Symbol: 'l' Symbol: 'i' Code: 001
Prior Symbol: 'l' Symbol: 'k' Code: 011001
Prior Symbol: 'l' Symbol: 'l' Code: 101
Prior Symbol: 'l' Symbol: 'm' Code: 1111010
Prior Symbol: 'l' Symbol: 'o' Code: 11111
Prior Symbol: 'l' Symbol: 'r' Code: 11110010
Prior Symbol: 'l' Symbol: 's' Code: 01101
Prior Symbol: 'l' Symbol: 't' Code: 011101
Prior Symbol: 'l' Symbol: 'u' Code: 01111
Prior Symbol: 'l' Symbol: 'v' Code: 1111011
Prior Symbol: 'l' Symbol: 'w' Code: 01110001
Prior Symbol: 'l' Symbol: 'y' Code: 1001
Prior Symbol: 'm' Symbol: 0 Code: 0100
Prior Symbol: 'm' Symbol: 27 Code: 010101
Prior Symbol: 'm' Symbol: ' ' Code: 001
Prior Symbol: 'm' Symbol: 'a' Code: 101
Prior Symbol: 'm' Symbol: 'b' Code: 0000
Prior Symbol: 'm' Symbol: 'e' Code: 11
Prior Symbol: 'm' Symbol: 'i' Code: 011
Prior Symbol: 'm' Symbol: 'm' Code: 0001
Prior Symbol: 'm' Symbol: 'o' Code: 1001
Prior Symbol: 'm' Symbol: 'p' Code: 1000
Prior Symbol: 'm' Symbol: 's' Code: 010111
Prior Symbol: 'm' Symbol: 'u' Code: 010110
Prior Symbol: 'm' Symbol: 'y' Code: 010100
Prior Symbol: 'n' Symbol: 0 Code: 000
Prior Symbol: 'n' Symbol: 27 Code: 01110011
Prior Symbol: 'n' Symbol: ' ' Code: 110
Prior Symbol: 'n' Symbol: ''' Code: 011101
Prior Symbol: 'n' Symbol: ':' Code: 1001010
Prior Symbol: 'n' Symbol: 'a' Code: 11100
Prior Symbol: 'n' Symbol: 'b' Code: 111010000

Prior Symbol: 'n' Symbol: 'c' Code: 01111
Prior Symbol: 'n' Symbol: 'd' Code: 001
Prior Symbol: 'n' Symbol: 'e' Code: 010
Prior Symbol: 'n' Symbol: 'f' Code: 1001011
Prior Symbol: 'n' Symbol: 'g' Code: 101
Prior Symbol: 'n' Symbol: 'h' Code: 111010101
Prior Symbol: 'n' Symbol: 'i' Code: 1000
Prior Symbol: 'n' Symbol: 'j' Code: 111010001
Prior Symbol: 'n' Symbol: 'k' Code: 1110110
Prior Symbol: 'n' Symbol: 'l' Code: 111010110
Prior Symbol: 'n' Symbol: 'm' Code: 111010111
Prior Symbol: 'n' Symbol: 'n' Code: 10011
Prior Symbol: 'n' Symbol: 'o' Code: 1110111
Prior Symbol: 'n' Symbol: 'r' Code: 111010100
Prior Symbol: 'n' Symbol: 's' Code: 0110
Prior Symbol: 'n' Symbol: 't' Code: 1111
Prior Symbol: 'n' Symbol: 'u' Code: 11101001
Prior Symbol: 'n' Symbol: 'v' Code: 0111000
Prior Symbol: 'n' Symbol: 'y' Code: 100100
Prior Symbol: 'n' Symbol: 'z' Code: 01110010
Prior Symbol: 'o' Symbol: 0 Code: 00101
Prior Symbol: 'o' Symbol: 27 Code: 01110001
Prior Symbol: 'o' Symbol: ' ' Code: 0101
Prior Symbol: 'o' Symbol: ''' Code: 01110000
Prior Symbol: 'o' Symbol: '.' Code: 0111011010
Prior Symbol: 'o' Symbol: '?' Code: 011101100
Prior Symbol: 'o' Symbol: 'a' Code: 1100010
Prior Symbol: 'o' Symbol: 'b' Code: 001001
Prior Symbol: 'o' Symbol: 'c' Code: 110000
Prior Symbol: 'o' Symbol: 'd' Code: 01111
Prior Symbol: 'o' Symbol: 'e' Code: 0111001
Prior Symbol: 'o' Symbol: 'f' Code: 1001
Prior Symbol: 'o' Symbol: 'g' Code: 00010
Prior Symbol: 'o' Symbol: 'h' Code: 0111010
Prior Symbol: 'o' Symbol: 'i' Code: 01110111
Prior Symbol: 'o' Symbol: 'k' Code: 1100011
Prior Symbol: 'o' Symbol: 'l' Code: 0100
Prior Symbol: 'o' Symbol: 'm' Code: 1000
Prior Symbol: 'o' Symbol: 'n' Code: 111
Prior Symbol: 'o' Symbol: 'o' Code: 0011
Prior Symbol: 'o' Symbol: 'p' Code: 01101
Prior Symbol: 'o' Symbol: 'r' Code: 101
Prior Symbol: 'o' Symbol: 's' Code: 11001
Prior Symbol: 'o' Symbol: 't' Code: 00011
Prior Symbol: 'o' Symbol: 'u' Code: 1101
Prior Symbol: 'o' Symbol: 'v' Code: 01100
Prior Symbol: 'o' Symbol: 'w' Code: 0000
Prior Symbol: 'o' Symbol: 'x' Code: 0010000
Prior Symbol: 'o' Symbol: 'y' Code: 0010001
Prior Symbol: 'o' Symbol: 'z' Code: 0111011011
Prior Symbol: 'p' Symbol: 0 Code: 1101
Prior Symbol: 'p' Symbol: 27 Code: 101110
Prior Symbol: 'p' Symbol: ' ' Code: 010
Prior Symbol: 'p' Symbol: ''' Code: 1100101
Prior Symbol: 'p' Symbol: 'a' Code: 1001
Prior Symbol: 'p' Symbol: 'd' Code: 101111
Prior Symbol: 'p' Symbol: 'e' Code: 111
Prior Symbol: 'p' Symbol: 'h' Code: 11000
Prior Symbol: 'p' Symbol: 'i' Code: 1010
Prior Symbol: 'p' Symbol: 'l' Code: 0110
Prior Symbol: 'p' Symbol: 'm' Code: 1100100
Prior Symbol: 'p' Symbol: 'o' Code: 00
Prior Symbol: 'p' Symbol: 'p' Code: 0111
Prior Symbol: 'p' Symbol: 'r' Code: 10001
Prior Symbol: 'p' Symbol: 's' Code: 10000
Prior Symbol: 'p' Symbol: 't' Code: 10110
Prior Symbol: 'p' Symbol: 'y' Code: 110011
Prior Symbol: 'q' Symbol: 27 Code: 0
Prior Symbol: 'q' Symbol: 'u' Code: 1
Prior Symbol: 'r' Symbol: 0 Code: 1001
Prior Symbol: 'r' Symbol: 27 Code: 01100101
Prior Symbol: 'r' Symbol: ' ' Code: 1111
Prior Symbol: 'r' Symbol: ''' Code: 0110011
Prior Symbol: 'r' Symbol: ',' Code: 110011101
Prior Symbol: 'r' Symbol: '.' Code: 0111100
Prior Symbol: 'r' Symbol: ':' Code: 110011100
Prior Symbol: 'r' Symbol: 'a' Code: 000
Prior Symbol: 'r' Symbol: 'b' Code: 01111101
Prior Symbol: 'r' Symbol: 'c' Code: 0111111
Prior Symbol: 'r' Symbol: 'd' Code: 11000
Prior Symbol: 'r' Symbol: 'e' Code: 101
Prior Symbol: 'r' Symbol: 'f' Code: 11001111
Prior Symbol: 'r' Symbol: 'g' Code: 0111101
Prior Symbol: 'r' Symbol: 'i' Code: 010
Prior Symbol: 'r' Symbol: 'k' Code: 110010
Prior Symbol: 'r' Symbol: 'l' Code: 0011
Prior Symbol: 'r' Symbol: 'm' Code: 011000
Prior Symbol: 'r' Symbol: 'n' Code: 01101
Prior Symbol: 'r' Symbol: 'o' Code: 1101
Prior Symbol: 'r' Symbol: 'p' Code: 01111100
Prior Symbol: 'r' Symbol: 'r' Code: 01110
Prior Symbol: 'r' Symbol: 's' Code: 1110
Prior Symbol: 'r' Symbol: 't' Code: 1000
Prior Symbol: 'r' Symbol: 'u' Code: 1100110
Prior Symbol: 'r' Symbol: 'v' Code: 01100100
Prior Symbol: 'r' Symbol: 'y' Code: 0010
Prior Symbol: 's' Symbol: 0 Code: 11
Prior Symbol: 's' Symbol: 27 Code: 0010011
Prior Symbol: 's' Symbol: ' ' Code: 01
Prior Symbol: 's' Symbol: ''' Code: 001011010
Prior Symbol: 's' Symbol: ',' Code: 001011011
Prior Symbol: 's' Symbol: '.' Code: 00100101
Prior Symbol: 's' Symbol: ':' Code: 0000001
Prior Symbol: 's' Symbol: '?' Code: 001011100
Prior Symbol: 's' Symbol: 'C' Code: 001011101
Prior Symbol: 's' Symbol: 'H' Code: 001011110
Prior Symbol: 's' Symbol: 'a' Code: 101010
Prior Symbol: 's' Symbol: 'c' Code: 101011
Prior Symbol: 's' Symbol: 'd' Code: 001011111
Prior Symbol: 's' Symbol: 'e' Code: 1011
Prior Symbol: 's' Symbol: 'f' Code: 00000000
Prior Symbol: 's' Symbol: 'h' Code: 00001
Prior Symbol: 's' Symbol: 'i' Code: 0011
Prior Symbol: 's' Symbol: 'k' Code: 000001
Prior Symbol: 's' Symbol: 'l' Code: 00101010
Prior Symbol: 's' Symbol: 'm' Code: 00000001
Prior Symbol: 's' Symbol: 'n' Code: 00101011
Prior Symbol: 's' Symbol: 'o' Code: 10100
Prior Symbol: 's' Symbol: 'p' Code: 001000
Prior Symbol: 's' Symbol: 'r' Code: 00100100
Prior Symbol: 's' Symbol: 's' Code: 0001
Prior Symbol: 's' Symbol: 't' Code: 100
Prior Symbol: 's' Symbol: 'u' Code: 0010100
Prior Symbol: 's' Symbol: 'y' Code: 00101100
Prior Symbol: 't' Symbol: 0 Code: 010
Prior Symbol: 't' Symbol: 27 Code: 11000010
Prior Symbol: 't' Symbol: ' ' Code: 101
Prior Symbol: 't' Symbol: ''' Code: 11000011
Prior Symbol: 't' Symbol: ':' Code: 110110000
Prior Symbol: 't' Symbol: '?' Code: 110110001
Prior Symbol: 't' Symbol: 'a' Code: 0000
Prior Symbol: 't' Symbol: 'b' Code: 100000
Prior Symbol: 't' Symbol: 'c' Code: 1101101
Prior Symbol: 't' Symbol: 'd' Code: 11000000
Prior Symbol: 't' Symbol: 'e' Code: 011
Prior Symbol: 't' Symbol: 'h' Code: 111
Prior Symbol: 't' Symbol: 'i' Code: 001
Prior Symbol: 't' Symbol: 'l' Code: 10001
Prior Symbol: 't' Symbol: 'm' Code: 100001
Prior Symbol: 't' Symbol: 'n' Code: 11011001
Prior Symbol: 't' Symbol: 'o' Code: 1001
Prior Symbol: 't' Symbol: 'r' Code: 11010
Prior Symbol: 't' Symbol: 's' Code: 0001
Prior Symbol: 't' Symbol: 't' Code: 110111
Prior Symbol: 't' Symbol: 'u' Code: 11001
Prior Symbol: 't' Symbol: 'w' Code: 11000001
Prior Symbol: 't' Symbol: 'y' Code: 110001
Prior Symbol: 'u' Symbol: 0 Code: 0011110
Prior Symbol: 'u' Symbol: 27 Code: 000100
Prior Symbol: 'u' Symbol: ' ' Code: 001110
Prior Symbol: 'u' Symbol: 'a' Code: 00110
Prior Symbol: 'u' Symbol: 'b' Code: 10011
Prior Symbol: 'u' Symbol: 'c' Code: 11100
Prior Symbol: 'u' Symbol: 'd' Code: 10000
Prior Symbol: 'u' Symbol: 'e' Code: 0010
Prior Symbol: 'u' Symbol: 'f' Code: 0011111
Prior Symbol: 'u' Symbol: 'g' Code: 11101
Prior Symbol: 'u' Symbol: 'i' Code: 00011
Prior Symbol: 'u' Symbol: 'k' Code: 0001010
Prior Symbol: 'u' Symbol: 'l' Code: 0000
Prior Symbol: 'u' Symbol: 'm' Code: 10010
Prior Symbol: 'u' Symbol: 'n' Code: 110
Prior Symbol: 'u' Symbol: 'p' Code: 10001
Prior Symbol: 'u' Symbol: 'r' Code: 01
Prior Symbol: 'u' Symbol: 's' Code: 101
Prior Symbol: 'u' Symbol: 't' Code: 1111
Prior Symbol: 'u' Symbol: 'z' Code: 0001011
Prior Symbol: 'v' Symbol: 27 Code: 0010
Prior Symbol: 'v' Symbol: 'a' Code: 000
Prior Symbol: 'v' Symbol: 'e' Code: 1
Prior Symbol: 'v' Symbol: 'i' Code: 01
Prior Symbol: 'v' Symbol: 'o' Code: 00111
Prior Symbol: 'v' Symbol: 's' Code: 00110
Prior Symbol: 'w' Symbol: 0 Code: 001
Prior Symbol: 'w' Symbol: 27 Code: 01010
Prior Symbol: 'w' Symbol: ' ' Code: 011
Prior Symbol: 'w' Symbol: ''' Code: 010010
Prior Symbol: 'w' Symbol: 'a' Code: 000
Prior Symbol: 'w' Symbol: 'b' Code: 010011
Prior Symbol: 'w' Symbol: 'c' Code: 010111
Prior Symbol: 'w' Symbol: 'e' Code: 1111
Prior Symbol: 'w' Symbol: 'i' Code: 1100
Prior Symbol: 'w' Symbol: 'l' Code: 010110
Prior Symbol: 'w' Symbol: 'n' Code: 1110
Prior Symbol: 'w' Symbol: 'o' Code: 1101
Prior Symbol: 'w' Symbol: 'r' Code: 01000
Prior Symbol: 'w' Symbol: 's' Code: 10
Prior Symbol: 'x' Symbol: 0 Code: 110
Prior Symbol: 'x' Symbol: 27 Code: 1010
Prior Symbol: 'x' Symbol: ' ' Code: 1011
Prior Symbol: 'x' Symbol: 'a' Code: 000
Prior Symbol: 'x' Symbol: 'e' Code: 001
Prior Symbol: 'x' Symbol: 'i' Code: 100
Prior Symbol: 'x' Symbol: 'p' Code: 111
Prior Symbol: 'x' Symbol: 't' Code: 01
Prior Symbol: 'y' Symbol: 0 Code: 10
Prior Symbol: 'y' Symbol: 27 Code: 111110
Prior Symbol: 'y' Symbol: ' ' Code: 0
Prior Symbol: 'y' Symbol: '!' Code: 1101101
Prior Symbol: 'y' Symbol: ''' Code: 110101
Prior Symbol: 'y' Symbol: '-' Code: 11110101
Prior Symbol: 'y' Symbol: 'a' Code: 1101110
Prior Symbol: 'y' Symbol: 'b' Code: 1111011
Prior Symbol: 'y' Symbol: 'c' Code: 11110100
Prior Symbol: 'y' Symbol: 'd' Code: 1100000
Prior Symbol: 'y' Symbol: 'e' Code: 11001
Prior Symbol: 'y' Symbol: 'i' Code: 1100001
Prior Symbol: 'y' Symbol: 'l' Code: 111111
Prior Symbol: 'y' Symbol: 'm' Code: 1101111
Prior Symbol: 'y' Symbol: 'n' Code: 1100010
Prior Symbol: 'y' Symbol: 'o' Code: 1100011
Prior Symbol: 'y' Symbol: 'p' Code: 1101000
Prior Symbol: 'y' Symbol: 's' Code: 1110
Prior Symbol: 'y' Symbol: 't' Code: 1101001
Prior Symbol: 'y' Symbol: 'v' Code: 1101100
Prior Symbol: 'y' Symbol: 'w' Code: 111100
Prior Symbol: 'z' Symbol: 0 Code: 110
Prior Symbol: 'z' Symbol: 27 Code: 100
Prior Symbol: 'z' Symbol: ' ' Code: 000
Prior Symbol: 'z' Symbol: 'a' Code: 01
Prior Symbol: 'z' Symbol: 'e' Code: 1010
Prior Symbol: 'z' Symbol: 'i' Code: 111
Prior Symbol: 'z' Symbol: 'y' Code: 001
Prior Symbol: 'z' Symbol: 'z' Code: 1011
Prior Symbol: '{' Symbol: 27 Code: 1
Prior Symbol: '|' Symbol: 27 Code: 1
Prior Symbol: '}' Symbol: 27 Code: 1
Prior Symbol: '~' Symbol: 27 Code: 1
Prior Symbol: 127 Symbol: 27 Code: 1

Table C5 English-language Program Title Decode Table

0	1	76	1	152	3	228	6	304	212	380	185	456	35
1	0	77	218	153	50	229	102	305	18	381	1	457	36
2	1	78	1	154	3	230	6	306	19	382	167	458	37
3	58	79	220	155	62	231	154	307	20	383	177	459	38
4	1	80	1	156	3	232	6	308	21	384	236	460	39
5	60	81	230	157	82	233	208	309	22	385	209	461	40
6	1	82	1	158	3	234	6	310	23	386	2	462	1
7	62	83	232	159	100	235	252	311	24	387	173	463	128
8	1	84	1	160	3	236	7	312	25	388	178	464	160
9	64	85	234	161	122	237	34	313	26	389	218	465	155
10	1	86	1	162	3	238	7	314	155	390	227	466	155
11	66	87	240	163	148	239	44	315	155	391	179	467	155
12	1	88	1	164	3	240	7	316	155	392	3	468	155
13	68	89	242	165	152	241	70	317	155	393	228	469	155
14	1	90	1	166	3	242	7	318	155	394	230	470	177
15	70	91	244	167	164	243	84	319	155	395	4	471	155
16	1	92	2	168	3	244	7	320	155	396	155	472	155
17	72	93	6	169	200	245	124	321	155	397	226	473	155
18	1	94	2	170	3	246	7	322	155	398	5	474	155
19	74	95	18	171	222	247	138	323	155	399	6	475	160
20	1	96	2	172	3	248	7	324	155	400	7	476	4
21	76	97	20	173	230	249	140	325	155	401	8	477	243
22	1	98	2	174	3	250	7	326	155	402	9	478	228
23	78	99	28	175	244	251	142	327	155	403	213	479	185
24	1	100	2	176	4	252	7	328	155	404	10	480	1
25	80	101	40	177	4	253	144	329	155	405	214	481	244
26	1	102	2	178	4	254	7	330	155	406	11	482	160
27	82	103	48	179	6	255	146	331	155	407	217	483	155
28	1	104	2	180	4	256	27	332	155	408	12	484	2
29	84	105	52	181	12	257	28	333	155	409	166	485	3
30	1	106	2	182	4	258	180	334	155	410	233	486	155
31	86	107	54	183	16	259	164	335	155	411	203	487	155
32	1	108	2	184	4	260	178	336	155	412	197	488	155
33	88	109	56	185	18	261	183	337	155	413	207	489	155
34	1	110	2	186	4	262	218	338	155	414	13	490	1
35	90	111	58	187	20	263	1	339	155	415	14	491	2
36	1	112	2	188	4	264	209	340	155	416	202	492	155
37	92	113	60	189	22	265	2	341	155	417	201	493	193
38	1	114	2	190	4	266	3	342	155	418	15	494	200
39	94	115	62	191	24	267	155	343	155	419	199	495	211
40	1	116	2	192	4	268	4	344	155	420	16	496	155
41	96	117	70	193	26	269	213	345	155	421	17	497	155
42	1	118	2	194	4	270	217	346	155	422	225	498	155
43	98	119	72	195	28	271	5	347	155	423	18	499	160
44	1	120	2	196	4	272	203	348	155	424	19	500	7
45	100	121	74	197	82	273	214	349	155	425	198	501	8
46	1	122	2	198	4	274	6	350	155	426	210	502	177
47	102	123	76	199	106	275	207	351	155	427	200	503	210
48	1	124	2	200	4	276	7	352	155	428	206	504	211
49	104	125	78	201	142	277	8	353	155	429	193	505	212
50	1	126	2	202	4	278	202	354	155	430	196	506	213
51	106	127	80	203	174	279	9	355	155	431	208	507	173
52	1	128	2	204	4	280	201	356	155	432	204	508	205
53	108	129	82	205	238	281	197	357	155	433	20	509	193
54	1	130	2	206	5	282	198	358	155	434	21	510	1
55	110	131	84	207	6	283	10	359	155	435	239	511	2
56	1	132	2	208	5	284	210	360	155	436	194	512	3
57	112	133	126	209	40	285	196	361	155	437	215	513	160
58	1	134	2	210	5	286	199	362	155	438	22	514	4
59	114	135	146	211	68	287	204	363	155	439	205	515	155
60	1	136	2	212	5	288	208	364	155	440	23	516	5
61	116	137	172	213	114	289	200	365	155	441	244	517	6
62	1	138	2	214	5	290	215	366	155	442	212	518	160
63	118	139	186	215	118	291	206	367	155	443	24	519	5
64	1	140	2	216	5	292	11	368	155	444	25	520	201
65	120	141	210	217	144	293	193	369	155	445	26	521	215
66	1	142	2	218	5	294	12	370	155	446	195	522	211
67	206	143	228	219	190	295	194	371	155	447	211	523	1
68	1	144	2	220	5	296	205	372	155	448	27	524	2
69	210	145	250	221	214	297	195	373	155	449	28	525	155
70	1	146	3	222	6	298	13	374	155	450	29	526	174
71	212	147	6	223	10	299	14	375	155	451	30	527	128
72	1	148	3	224	6	300	15	376	41	452	31	528	3
73	214	149	30	225	68	301	16	377	42	453	32	529	4
74	1	150	3	226	6	302	211	378	216	454	33	530	155
75	216	151	38	227	100	303	17	379	229	455	34	531	155

532	2	612	227	692	233	772	229	852	206	932	16	1012	6
533	3	613	230	693	229	773	4	853	160	933	17	1013	7
534	173	614	247	694	239	774	238	854	198	934	170	1014	198
535	155	615	3	695	3	775	11	855	245	935	236	1015	215
536	1	616	245	696	225	776	186	856	1	936	241	1016	1
537	128	617	4	697	4	777	212	857	2	937	174	1017	155
538	160	618	5	698	10	778	174	858	155	938	160	1018	242
539	176	619	6	699	11	779	242	859	194	939	247	1019	2
540	4	620	242	700	241	780	227	860	3	940	237	1020	3
541	5	621	7	701	245	781	1	861	225	941	238	1021	232
542	128	622	8	702	243	782	160	862	4	942	1	1022	229
543	155	623	9	703	1	783	2	863	239	943	2	1023	225
544	177	624	10	704	237	784	128	864	5	944	155	1024	4
545	178	625	11	705	249	785	155	865	233	945	235	1025	233
546	160	626	12	706	195	786	237	866	6	946	3	1026	239
547	176	627	228	707	2	787	3	867	7	947	4	1027	5
548	185	628	160	708	236	788	201	868	9	948	5	1028	155
549	1	629	13	709	238	789	243	869	10	949	6	1029	155
550	2	630	236	710	228	790	244	870	228	950	227	1030	2
551	3	631	238	711	248	791	4	871	243	951	7	1031	239
552	2	632	14	712	3	792	5	872	230	952	239	1032	225
553	3	633	237	713	155	793	6	873	246	953	8	1033	155
554	177	634	15	714	246	794	7	874	247	954	233	1034	1
555	186	635	16	715	4	795	8	875	240	955	245	1035	229
556	1	636	17	716	5	796	9	876	242	956	9	1036	1
557	176	637	18	717	225	797	10	877	1	957	225	1037	239
558	155	638	8	718	6	798	2	878	236	958	229	1038	155
559	128	639	9	719	7	799	3	879	2	959	240	1039	225
560	128	640	193	720	8	800	155	880	3	960	232	1040	155
561	1	641	211	721	9	801	245	881	160	961	10	1041	155
562	176	642	155	722	7	802	1	882	155	962	11	1042	155
563	155	643	1	723	8	803	225	883	4	963	12	1043	155
564	155	644	195	724	160	804	239	884	5	964	13	1044	155
565	184	645	2	725	155	805	229	885	245	965	244	1045	155
566	155	646	233	726	204	806	5	886	6	966	14	1046	155
567	155	647	236	727	1	807	233	887	7	967	15	1047	155
568	155	648	3	728	229	808	225	888	238	968	232	1048	155
569	155	649	242	729	2	809	239	889	8	969	10	1049	155
570	155	650	245	730	236	810	245	890	11	970	173	1050	155
571	176	651	4	731	245	811	238	891	12	971	206	1051	155
572	155	652	239	732	239	812	155	892	160	972	155	1052	25
573	160	653	225	733	3	813	229	893	243	973	1	1053	26
574	2	654	5	734	233	814	1	894	249	974	214	1054	155
575	3	655	229	735	242	815	2	895	174	975	2	1055	186
576	177	656	6	736	4	816	3	896	210	976	245	1056	229
577	179	657	7	737	5	817	4	897	199	977	247	1057	234
578	185	658	11	738	225	818	4	898	1	978	3	1058	248
579	176	659	12	739	6	819	5	899	155	979	4	1059	1
580	1	660	193	740	9	820	160	900	2	980	225	1060	2
581	155	661	249	741	10	821	155	901	245	981	229	1061	230
582	155	662	1	742	174	822	1	902	3	982	233	1062	167
583	160	663	194	743	236	823	245	903	4	983	5	1063	3
584	155	664	207	744	249	824	2	904	5	984	242	1064	250
585	155	665	229	745	193	825	229	905	233	985	6	1065	232
586	155	666	245	746	232	826	239	906	236	986	239	1066	4
587	155	667	155	747	1	827	3	907	6	987	7	1067	247
588	155	668	233	748	155	828	225	908	229	988	8	1068	5
589	155	669	2	749	2	829	233	909	7	989	9	1069	245
590	155	670	160	750	3	830	8	910	239	990	238	1070	226
591	155	671	3	751	4	831	9	911	8	991	3	1071	6
592	155	672	4	752	225	832	170	912	225	992	236	1072	235
593	128	673	5	753	245	833	212	913	9	993	174	1073	7
594	155	674	242	754	233	834	1	914	242	994	1	1074	240
595	155	675	6	755	5	835	155	915	10	995	155	1075	8
596	19	676	236	756	229	836	227	916	1	996	2	1076	128
597	20	677	7	757	6	837	2	917	245	997	240	1077	246
598	170	678	225	758	242	838	242	918	155	998	6	1078	231
599	173	679	8	759	239	839	3	919	214	999	233	1079	9
600	174	680	9	760	7	840	229	920	4	1000	160	1080	228
601	246	681	232	761	8	841	4	921	5	1001	195	1081	10
602	231	682	10	762	239	842	245	922	232	1002	239	1082	160
603	244	683	239	763	5	843	249	923	155	1003	155	1083	233
604	226	684	5	764	128	844	233	924	1	1004	229	1084	11
605	233	685	6	765	155	845	5	925	245	1005	1	1085	227
606	1	686	249	766	245	846	239	926	2	1006	128	1086	249
607	2	687	155	767	1	847	6	927	225	1007	2	1087	12
608	194	688	1	768	2	848	7	928	233	1008	3	1088	13
609	240	689	245	769	233	849	225	929	239	1009	225	1089	237
610	155	690	2	770	225	850	229	930	3	1010	4	1090	14
611	243	691	242	771	3	851	8	931	229	1011	5	1091	15

1092	243	1172	236	1252	23	1332	4	1412	4	1492	9	1572	9
1093	16	1173	2	1253	128	1333	242	1413	5	1493	229	1573	128
1094	17	1174	228	1254	24	1334	5	1414	243	1494	24	1574	246
1095	236	1175	231	1255	25	1335	128	1415	6	1495	25	1575	240
1096	18	1176	242	1256	242	1336	6	1416	7	1496	226	1576	10
1097	244	1177	3	1257	26	1337	160	1417	8	1497	234	1577	228
1098	242	1178	155	1258	27	1338	225	1418	233	1498	242	1578	11
1099	19	1179	239	1259	160	1339	239	1419	160	1499	232	1579	243
1100	238	1180	4	1260	28	1340	7	1420	9	1500	236	1580	247
1101	20	1181	246	1261	29	1341	244	1421	128	1501	237	1581	12
1102	21	1182	5	1262	160	1342	233	1422	229	1502	250	1582	13
1103	22	1183	6	1263	11	1343	8	1423	10	1503	155	1583	239
1104	23	1184	249	1264	245	1344	9	1424	21	1504	1	1584	236
1105	24	1185	243	1265	155	1345	10	1425	22	1505	245	1585	160
1106	10	1186	7	1266	1	1346	11	1426	167	1506	2	1586	14
1107	11	1187	233	1267	236	1347	12	1427	186	1507	3	1587	15
1108	243	1188	225	1268	243	1348	21	1428	227	1508	246	1588	237
1109	155	1189	8	1269	242	1349	22	1429	247	1509	4	1589	230
1110	245	1190	9	1270	128	1350	161	1430	242	1510	186	1590	16
1111	226	1191	128	1271	225	1351	248	1431	173	1511	230	1591	245
1112	1	1192	10	1272	2	1352	233	1432	226	1512	5	1592	17
1113	128	1193	11	1273	3	1353	235	1433	1	1513	6	1593	18
1114	160	1194	229	1274	244	1354	1	1434	2	1514	235	1594	19
1115	2	1195	12	1275	233	1355	128	1435	155	1515	239	1595	20
1116	229	1196	13	1276	239	1356	155	1436	230	1516	7	1596	21
1117	242	1197	160	1277	230	1357	250	1437	3	1517	167	1597	242
1118	233	1198	30	1278	4	1358	226	1438	237	1518	249	1598	22
1119	3	1199	31	1279	5	1359	2	1439	246	1519	8	1599	238
1120	236	1200	155	1280	6	1360	3	1440	4	1520	9	1600	23
1121	4	1201	161	1281	7	1361	4	1441	235	1521	10	1601	24
1122	249	1202	173	1282	229	1362	160	1442	5	1522	11	1602	25
1123	5	1203	232	1283	8	1363	240	1443	244	1523	227	1603	26
1124	239	1204	234	1284	9	1364	5	1444	6	1524	12	1604	14
1125	6	1205	241	1285	10	1365	6	1445	7	1525	238	1605	15
1126	225	1206	245	1286	15	1366	7	1446	8	1526	225	1606	237
1127	7	1207	250	1287	16	1367	225	1447	243	1527	13	1607	167
1128	8	1208	1	1288	186	1368	8	1448	9	1528	243	1608	155
1129	9	1209	2	1289	249	1369	230	1449	245	1529	14	1609	228
1130	16	1210	3	1290	167	1370	242	1450	10	1530	233	1610	1
1131	17	1211	4	1291	244	1371	237	1451	239	1531	15	1611	249
1132	195	1212	186	1292	155	1372	246	1452	11	1532	16	1612	243
1133	204	1213	248	1293	1	1373	9	1453	12	1533	244	1613	242
1134	199	1214	167	1294	231	1374	228	1454	128	1534	128	1614	244
1135	155	1215	226	1295	236	1375	10	1455	249	1535	228	1615	2
1136	227	1216	233	1296	2	1376	239	1456	225	1536	229	1616	232
1137	1	1217	5	1297	238	1377	244	1457	13	1537	17	1617	3
1138	128	1218	6	1298	3	1378	236	1458	228	1538	18	1618	236
1139	236	1219	7	1299	239	1379	243	1459	233	1539	231	1619	240
1140	249	1220	230	1300	245	1380	231	1460	160	1540	160	1620	4
1141	2	1221	237	1301	4	1381	229	1461	14	1541	19	1621	225
1142	243	1222	231	1302	242	1382	11	1462	15	1542	20	1622	233
1143	3	1223	235	1303	5	1383	227	1463	236	1543	21	1623	5
1144	245	1224	8	1304	6	1384	12	1464	229	1544	22	1624	6
1145	4	1225	9	1305	233	1385	13	1465	16	1545	23	1625	128
1146	5	1226	246	1306	7	1386	14	1466	17	1546	27	1626	160
1147	242	1227	240	1307	243	1387	15	1467	18	1547	28	1627	7
1148	6	1228	10	1308	225	1388	16	1468	19	1548	174	1628	8
1149	233	1229	239	1309	8	1389	17	1469	20	1549	250	1629	9
1150	160	1230	11	1310	9	1390	18	1470	10	1550	191	1630	10
1151	7	1231	227	1311	10	1391	19	1471	11	1551	1	1631	229
1152	8	1232	12	1312	11	1392	238	1472	249	1552	167	1632	239
1153	239	1233	13	1313	229	1393	20	1473	155	1553	155	1633	11
1154	244	1234	14	1314	128	1394	239	1474	245	1554	2	1634	12
1155	9	1235	249	1315	12	1395	1	1475	243	1555	233	1635	13
1156	10	1236	15	1316	232	1396	155	1476	1	1556	248	1636	155
1157	225	1237	228	1317	160	1397	225	1477	2	1557	249	1637	245
1158	11	1238	236	1318	13	1398	11	1478	226	1558	3	1638	24
1159	232	1239	16	1319	14	1399	12	1479	237	1559	229	1639	25
1160	235	1240	229	1320	229	1400	212	1480	128	1560	232	1640	186
1161	229	1241	17	1321	13	1401	239	1481	3	1561	4	1641	172
1162	12	1242	244	1322	226	1402	230	1482	240	1562	225	1642	246
1163	13	1243	247	1323	245	1403	236	1483	239	1563	235	1643	155
1164	14	1244	18	1324	247	1404	247	1484	4	1564	5	1644	240
1165	15	1245	19	1325	155	1405	225	1485	160	1565	226	1645	226
1166	14	1246	225	1326	236	1406	1	1486	5	1566	6	1646	1
1167	15	1247	20	1327	1	1407	186	1487	233	1567	7	1647	230
1168	174	1248	21	1328	249	1408	2	1488	6	1568	227	1648	2
1169	245	1249	22	1329	238	1409	155	1489	225	1569	8	1649	167
1170	247	1250	238	1330	2	1410	249	1490	7	1570	231	1650	174
1171	1	1251	243	1331	3	1411	3	1491	8	1571	244	1651	231

1652	3	1694	191	1736	20	1778	128	1820	238	1862	5	1904	9
1653	227	1695	195	1737	21	1779	229	1821	13	1863	6	1905	10
1654	245	1696	200	1738	244	1780	14	1822	14	1864	155	1906	11
1655	4	1697	228	1739	22	1781	160	1823	242	1865	160	1907	12
1656	237	1698	230	1740	23	1782	15	1824	15	1866	225	1908	13
1657	5	1699	237	1741	160	1783	232	1825	16	1867	229	1909	14
1658	6	1700	242	1742	24	1784	16	1826	4	1868	233	1910	243
1659	7	1701	174	1743	128	1785	17	1827	229	1869	1	1911	15
1660	235	1702	236	1744	20	1786	18	1828	243	1870	128	1912	16
1661	8	1703	238	1745	21	1787	19	1829	239	1871	240	1913	17
1662	9	1704	249	1746	186	1788	17	1830	155	1872	2	1914	128
1663	238	1705	1	1747	191	1789	18	1831	1	1873	244	1915	18
1664	242	1706	2	1748	228	1790	235	1832	225	1874	3	1916	5
1665	10	1707	3	1749	247	1791	250	1833	2	1875	4	1917	6
1666	228	1708	4	1750	155	1792	128	1834	3	1876	160	1918	229
1667	11	1709	186	1751	167	1793	230	1835	233	1877	19	1919	250
1668	249	1710	5	1752	1	1794	155	1836	11	1878	227	1920	160
1669	236	1711	155	1753	238	1795	1	1837	12	1879	173	1921	249
1670	12	1712	245	1754	2	1796	160	1838	167	1880	228	1922	155
1671	13	1713	6	1755	3	1797	2	1839	226	1881	233	1923	1
1672	244	1714	7	1756	4	1798	3	1840	236	1882	238	1924	128
1673	128	1715	8	1757	227	1799	233	1841	227	1883	239	1925	233
1674	14	1716	9	1758	226	1800	225	1842	242	1884	240	1926	2
1675	239	1717	235	1759	237	1801	4	1843	1	1885	244	1927	225
1676	243	1718	240	1760	5	1802	228	1844	155	1886	246	1928	3
1677	160	1719	10	1761	249	1803	240	1845	2	1887	161	1929	4
1678	225	1720	11	1762	6	1804	237	1846	3	1888	225	1930	155
1679	15	1721	12	1763	244	1805	226	1847	4	1889	237	1931	155
1680	233	1722	225	1764	7	1806	227	1848	233	1890	1	1932	155
1681	16	1723	227	1765	236	1807	231	1849	239	1891	226	1933	155
1682	17	1724	13	1766	8	1808	236	1850	238	1892	2	1934	155
1683	229	1725	232	1767	245	1809	5	1851	229	1893	3	1935	155
1684	18	1726	14	1768	242	1810	229	1852	225	1894	4	1936	155
1685	19	1727	15	1769	9	1811	6	1853	128	1895	167	1937	155
1686	20	1728	239	1770	225	1812	7	1854	5	1896	5	1938	155
1687	21	1729	16	1771	243	1813	8	1855	160	1897	6	1939	155
1688	22	1730	17	1772	10	1814	9	1856	6	1898	247		
1689	23	1731	243	1773	239	1815	244	1857	7	1899	7		
1690	25	1732	18	1774	11	1816	10	1858	8	1900	155		
1691	26	1733	233	1775	12	1817	11	1859	9	1901	236		
1692	167	1734	19	1776	13	1818	12	1860	243	1902	8		
1693	172	1735	229	1777	233	1819	243	1861	10	1903	229		

4. STANDARD COMPRESSION TYPE 2 HUFFMAN ENCODE/DECODE TABLES

The following encode/decode tables are optimized for English-language program description text. These tables correspond to multiple_string_structure() with compression_type value 0x02, and mode equal to 0xFF.

Table C6 English-language Program Description Encode Table

Prior Symbol: 0	Symbol: 27	Code: 1110000
Prior Symbol: 0	Symbol: '"'	Code: 111001
Prior Symbol: 0	Symbol: 'A'	Code: 010
Prior Symbol: 0	Symbol: 'B'	Code: 0011
Prior Symbol: 0	Symbol: 'C'	Code: 0111
Prior Symbol: 0	Symbol: 'D'	Code: 11101
Prior Symbol: 0	Symbol: 'E'	Code: 10010
Prior Symbol: 0	Symbol: 'F'	Code: 10110
Prior Symbol: 0	Symbol: 'G'	Code: 011011
Prior Symbol: 0	Symbol: 'H'	Code: 10111
Prior Symbol: 0	Symbol: 'I'	Code: 011000
Prior Symbol: 0	Symbol: 'J'	Code: 1100
Prior Symbol: 0	Symbol: 'K'	Code: 00101
Prior Symbol: 0	Symbol: 'L'	Code: 10011
Prior Symbol: 0	Symbol: 'M'	Code: 1111
Prior Symbol: 0	Symbol: 'N'	Code: 00100
Prior Symbol: 0	Symbol: 'O'	Code: 011001
Prior Symbol: 0	Symbol: 'P'	Code: 000
Prior Symbol: 0	Symbol: 'R'	Code: 1000
Prior Symbol: 0	Symbol: 'S'	Code: 1010
Prior Symbol: 0	Symbol: 'T'	Code: 1101
Prior Symbol: 0	Symbol: 'V'	Code: 1110001
Prior Symbol: 0	Symbol: 'W'	Code: 011010
Prior Symbol: 1	Symbol: 27	Code: 1
Prior Symbol: 2	Symbol: 27	Code: 1
Prior Symbol: 3	Symbol: 27	Code: 1
Prior Symbol: 4	Symbol: 27	Code: 1
Prior Symbol: 5	Symbol: 27	Code: 1
Prior Symbol: 6	Symbol: 27	Code: 1
Prior Symbol: 7	Symbol: 27	Code: 1
Prior Symbol: 8	Symbol: 27	Code: 1
Prior Symbol: 9	Symbol: 27	Code: 1
Prior Symbol: 10	Symbol: 27	Code: 1
Prior Symbol: 11	Symbol: 27	Code: 1
Prior Symbol: 12	Symbol: 27	Code: 1
Prior Symbol: 13	Symbol: 27	Code: 1
Prior Symbol: 14	Symbol: 27	Code: 1
Prior Symbol: 15	Symbol: 27	Code: 1
Prior Symbol: 16	Symbol: 27	Code: 1
Prior Symbol: 17	Symbol: 27	Code: 1
Prior Symbol: 18	Symbol: 27	Code: 1
Prior Symbol: 19	Symbol: 27	Code: 1
Prior Symbol: 20	Symbol: 27	Code: 1
Prior Symbol: 21	Symbol: 27	Code: 1
Prior Symbol: 22	Symbol: 27	Code: 1
Prior Symbol: 23	Symbol: 27	Code: 1
Prior Symbol: 24	Symbol: 27	Code: 1
Prior Symbol: 25	Symbol: 27	Code: 1
Prior Symbol: 26	Symbol: 27	Code: 1
Prior Symbol: 27	Symbol: 27	Code: 1
Prior Symbol: 28	Symbol: 27	Code: 1
Prior Symbol: 29	Symbol: 27	Code: 1
Prior Symbol: 30	Symbol: 27	Code: 1
Prior Symbol: 31	Symbol: 27	Code: 1
Prior Symbol: ' '	Symbol: 27	Code: 101000001
Prior Symbol: ' '	Symbol: '"'	Code: 111111010
Prior Symbol: ' '	Symbol: '('	Code: 1111111100
Prior Symbol: ' '	Symbol: '-'	Code: 11111111110
Prior Symbol: ' '	Symbol: '/'	Code: 11111111111
Prior Symbol: ' '	Symbol: '1'	Code: 0101011
Prior Symbol: ' '	Symbol: '2'	Code: 0100010
Prior Symbol: ' '	Symbol: '3'	Code: 1111111101
Prior Symbol: ' '	Symbol: '4'	Code: 110010100
Prior Symbol: ' '	Symbol: '5'	Code: 1111111110
Prior Symbol: ' '	Symbol: '7'	Code: 1010000000
Prior Symbol: ' '	Symbol: 'A'	Code: 10010
Prior Symbol: ' '	Symbol: 'B'	Code: 010100
Prior Symbol: ' '	Symbol: 'C'	Code: 111100
Prior Symbol: ' '	Symbol: 'D'	Code: 1111010
Prior Symbol: ' '	Symbol: 'E'	Code: 0100011
Prior Symbol: ' '	Symbol: 'F'	Code: 0101010
Prior Symbol: ' '	Symbol: 'G'	Code: 000010
Prior Symbol: ' '	Symbol: 'H'	Code: 1111011
Prior Symbol: ' '	Symbol: 'I'	Code: 11001011
Prior Symbol: ' '	Symbol: 'J'	Code: 000011
Prior Symbol: ' '	Symbol: 'K'	Code: 1100100
Prior Symbol: ' '	Symbol: 'L'	Code: 010110
Prior Symbol: ' '	Symbol: 'M'	Code: 101001
Prior Symbol: ' '	Symbol: 'N'	Code: 001100
Prior Symbol: ' '	Symbol: 'O'	Code: 10100001
Prior Symbol: ' '	Symbol: 'P'	Code: 001101
Prior Symbol: ' '	Symbol: 'R'	Code: 1111100
Prior Symbol: ' '	Symbol: 'S'	Code: 01001
Prior Symbol: ' '	Symbol: 'T'	Code: 1100110
Prior Symbol: ' '	Symbol: 'U'	Code: 111111011
Prior Symbol: ' '	Symbol: 'V'	Code: 111111100
Prior Symbol: ' '	Symbol: 'W'	Code: 010000
Prior Symbol: ' '	Symbol: 'Y'	Code: 111111101
Prior Symbol: ' '	Symbol: 'Z'	Code: 1010000001
Prior Symbol: ' '	Symbol: 'a'	Code: 011
Prior Symbol: ' '	Symbol: 'b'	Code: 10111
Prior Symbol: ' '	Symbol: 'c'	Code: 10011
Prior Symbol: ' '	Symbol: 'd'	Code: 10000
Prior Symbol: ' '	Symbol: 'e'	Code: 100010
Prior Symbol: ' '	Symbol: 'f'	Code: 11101
Prior Symbol: ' '	Symbol: 'g'	Code: 100011
Prior Symbol: ' '	Symbol: 'h'	Code: 0001
Prior Symbol: ' '	Symbol: 'i'	Code: 10101
Prior Symbol: ' '	Symbol: 'j'	Code: 11001111
Prior Symbol: ' '	Symbol: 'k'	Code: 11111010
Prior Symbol: ' '	Symbol: 'l'	Code: 010111
Prior Symbol: ' '	Symbol: 'm'	Code: 00000
Prior Symbol: ' '	Symbol: 'n'	Code: 1010001
Prior Symbol: ' '	Symbol: 'o'	Code: 0010
Prior Symbol: ' '	Symbol: 'p'	Code: 10110
Prior Symbol: ' '	Symbol: 'q'	Code: 110010101
Prior Symbol: ' '	Symbol: 'r'	Code: 00111
Prior Symbol: ' '	Symbol: 's'	Code: 11100
Prior Symbol: ' '	Symbol: 't'	Code: 1101
Prior Symbol: ' '	Symbol: 'u'	Code: 11111011
Prior Symbol: ' '	Symbol: 'v'	Code: 11111100
Prior Symbol: ' '	Symbol: 'w'	Code: 11000
Prior Symbol: ' '	Symbol: 'y'	Code: 11001110
Prior Symbol: '!'	Symbol: 27	Code: 1
Prior Symbol: '"'	Symbol: 0	Code: 000
Prior Symbol: '"'	Symbol: 27	Code: 10
Prior Symbol: '"'	Symbol: ' '	Code: 11
Prior Symbol: '"'	Symbol: '.'	Code: 001
Prior Symbol: '"'	Symbol: 'H'	Code: 010
Prior Symbol: '"'	Symbol: 'T'	Code: 011
Prior Symbol: '#'	Symbol: 27	Code: 1
Prior Symbol: '$'	Symbol: 27	Code: 1
Prior Symbol: '%'	Symbol: 27	Code: 1
Prior Symbol: '&'	Symbol: 27	Code: 1
Prior Symbol: '''	Symbol: 27	Code: 00
Prior Symbol: '''	Symbol: ' '	Code: 010
Prior Symbol: '''	Symbol: 's'	Code: 1
Prior Symbol: '''	Symbol: 't'	Code: 011
Prior Symbol: '('	Symbol: 27	Code: 1
Prior Symbol: ')'	Symbol: 27	Code: 1
Prior Symbol: ')'	Symbol: ','	Code: 0
Prior Symbol: '*'	Symbol: 27	Code: 1
Prior Symbol: '+'	Symbol: 27	Code: 1
Prior Symbol: ','	Symbol: 27	Code: 00
Prior Symbol: ','	Symbol: ' '	Code: 1
Prior Symbol: ','	Symbol: '"'	Code: 01
Prior Symbol: '-'	Symbol: 27	Code: 10
Prior Symbol: '-'	Symbol: ' '	Code: 1110
Prior Symbol: '-'	Symbol: 'a'	Code: 000
Prior Symbol: '-'	Symbol: 'b'	Code: 0010
Prior Symbol: '-'	Symbol: 'c'	Code: 110
Prior Symbol: '-'	Symbol: 'd'	Code: 0011
Prior Symbol: '-'	Symbol: 'e'	Code: 0100
Prior Symbol: '-'	Symbol: 'f'	Code: 0101
Prior Symbol: '-'	Symbol: 'r'	Code: 1111
Prior Symbol: '-'	Symbol: 's'	Code: 011
Prior Symbol: '.'	Symbol: 0	Code: 1
Prior Symbol: '.'	Symbol: 27	Code: 000
Prior Symbol: '.'	Symbol: ' '	Code: 01
Prior Symbol: '.'	Symbol: '"'	Code: 0010
Prior Symbol: '.'	Symbol: 'J'	Code: 00110
Prior Symbol: '.'	Symbol: 'S'	Code: 00111
Prior Symbol: '/'	Symbol: 27	Code: 0
Prior Symbol: '/'	Symbol: ' '	Code: 1
Prior Symbol: '0'	Symbol: 27	Code: 100
Prior Symbol: '0'	Symbol: ' '	Code: 111
Prior Symbol: '0'	Symbol: '0'	Code: 00
Prior Symbol: '0'	Symbol: '7'	Code: 101
Prior Symbol: '0'	Symbol: 's'	Code: 01
Prior Symbol: '0'	Symbol: 't'	Code: 110
Prior Symbol: '1'	Symbol: 27	Code: 111
Prior Symbol: '1'	Symbol: ' '	Code: 10
Prior Symbol: '1'	Symbol: '8'	Code: 110
Prior Symbol: '1'	Symbol: '9'	Code: 0
Prior Symbol: '2'	Symbol: 27	Code: 101
Prior Symbol: '2'	Symbol: ' '	Code: 11
Prior Symbol: '2'	Symbol: '.'	Code: 0
Prior Symbol: '2'	Symbol: '6'	Code: 100
Prior Symbol: '3'	Symbol: 27	Code: 10
Prior Symbol: '3'	Symbol: ' '	Code: 0
Prior Symbol: '3'	Symbol: '0'	Code: 11
Prior Symbol: '4'	Symbol: 27	Code: 10
Prior Symbol: '4'	Symbol: ' '	Code: 11
Prior Symbol: '4'	Symbol: '.'	Code: 0
Prior Symbol: '5'	Symbol: 27	Code: 11
Prior Symbol: '5'	Symbol: ' '	Code: 10
Prior Symbol: '5'	Symbol: '.'	Code: 0
Prior Symbol: '6'	Symbol: 27	Code: 1
Prior Symbol: '7'	Symbol: 27	Code: 0
Prior Symbol: '7'	Symbol: ','	Code: 10
Prior Symbol: '7'	Symbol: '.'	Code: 11
Prior Symbol: '8'	Symbol: 27	Code: 1
Prior Symbol: '9'	Symbol: 27	Code: 110
Prior Symbol: '9'	Symbol: ' '	Code: 111
Prior Symbol: '9'	Symbol: '5'	Code: 00
Prior Symbol: '9'	Symbol: '6'	Code: 01
Prior Symbol: '9'	Symbol: '8'	Code: 10
Prior Symbol: ':'	Symbol: 27	Code: 0
Prior Symbol: ':'	Symbol: ' '	Code: 1
Prior Symbol: ';'	Symbol: 27	Code: 0
Prior Symbol: ';'	Symbol: ' '	Code: 1
Prior Symbol: '<'	Symbol: 27	Code: 1
Prior Symbol: '='	Symbol: 27	Code: 1
Prior Symbol: '>'	Symbol: 27	Code: 1
Prior Symbol: '?'	Symbol: 27	Code: 0
Prior Symbol: '?'	Symbol: ' '	Code: 1
Prior Symbol: '@'	Symbol: 27	Code: 1
Prior Symbol: 'A'	Symbol: 27	Code: 10010
Prior Symbol: 'A'	Symbol: ' '	Code: 11
Prior Symbol: 'A'	Symbol: 'd'	Code: 10011
Prior Symbol: 'A'	Symbol: 'f'	Code: 101000

Prior Symbol: 'A' Symbol: 'l' Code: 00
Prior Symbol: 'A' Symbol: 'm' Code: 10101
Prior Symbol: 'A' Symbol: 'n' Code: 01
Prior Symbol: 'A' Symbol: 'r' Code: 1011
Prior Symbol: 'A' Symbol: 's' Code: 10000
Prior Symbol: 'A' Symbol: 't' Code: 10001
Prior Symbol: 'A' Symbol: 'u' Code: 101001
Prior Symbol: 'B' Symbol: 27 Code: 10010
Prior Symbol: 'B' Symbol: 'a' Code: 101
Prior Symbol: 'B' Symbol: 'e' Code: 111
Prior Symbol: 'B' Symbol: 'i' Code: 00
Prior Symbol: 'B' Symbol: 'l' Code: 10011
Prior Symbol: 'B' Symbol: 'o' Code: 110
Prior Symbol: 'B' Symbol: 'r' Code: 01
Prior Symbol: 'B' Symbol: 'u' Code: 1000
Prior Symbol: 'C' Symbol: 27 Code: 01110
Prior Symbol: 'C' Symbol: 'a' Code: 00
Prior Symbol: 'C' Symbol: 'h' Code: 10
Prior Symbol: 'C' Symbol: 'i' Code: 01111
Prior Symbol: 'C' Symbol: 'l' Code: 110
Prior Symbol: 'C' Symbol: 'o' Code: 111
Prior Symbol: 'C' Symbol: 'r' Code: 0101
Prior Symbol: 'C' Symbol: 'u' Code: 0110
Prior Symbol: 'C' Symbol: 'y' Code: 0100
Prior Symbol: 'D' Symbol: 27 Code: 1111
Prior Symbol: 'D' Symbol: 'a' Code: 01
Prior Symbol: 'D' Symbol: 'e' Code: 100
Prior Symbol: 'D' Symbol: 'i' Code: 00
Prior Symbol: 'D' Symbol: 'o' Code: 101
Prior Symbol: 'D' Symbol: 'r' Code: 1101
Prior Symbol: 'D' Symbol: 'u' Code: 1110
Prior Symbol: 'D' Symbol: 'y' Code: 1100
Prior Symbol: 'E' Symbol: 27 Code: 10
Prior Symbol: 'E' Symbol: 'a' Code: 0110
Prior Symbol: 'E' Symbol: 'd' Code: 000
Prior Symbol: 'E' Symbol: 'i' Code: 0111
Prior Symbol: 'E' Symbol: 'l' Code: 001
Prior Symbol: 'E' Symbol: 'n' Code: 1100
Prior Symbol: 'E' Symbol: 'r' Code: 111
Prior Symbol: 'E' Symbol: 's' Code: 010
Prior Symbol: 'E' Symbol: 'v' Code: 1101
Prior Symbol: 'F' Symbol: 27 Code: 00
Prior Symbol: 'F' Symbol: 'e' Code: 100
Prior Symbol: 'F' Symbol: 'l' Code: 101
Prior Symbol: 'F' Symbol: 'o' Code: 01
Prior Symbol: 'F' Symbol: 'r' Code: 11
Prior Symbol: 'G' Symbol: 27 Code: 000
Prior Symbol: 'G' Symbol: 'a' Code: 110
Prior Symbol: 'G' Symbol: 'e' Code: 01
Prior Symbol: 'G' Symbol: 'i' Code: 100
Prior Symbol: 'G' Symbol: 'l' Code: 001
Prior Symbol: 'G' Symbol: 'o' Code: 1011
Prior Symbol: 'G' Symbol: 'r' Code: 111
Prior Symbol: 'G' Symbol: 'u' Code: 1010
Prior Symbol: 'H' Symbol: 27 Code: 010
Prior Symbol: 'H' Symbol: 'a' Code: 00
Prior Symbol: 'H' Symbol: 'e' Code: 011
Prior Symbol: 'H' Symbol: 'i' Code: 110
Prior Symbol: 'H' Symbol: 'o' Code: 10
Prior Symbol: 'H' Symbol: 'u' Code: 111
Prior Symbol: 'I' Symbol: 27 Code: 011
Prior Symbol: 'I' Symbol: ' ' Code: 000
Prior Symbol: 'I' Symbol: '.' Code: 100
Prior Symbol: 'I' Symbol: 'l' Code: 001
Prior Symbol: 'I' Symbol: 'n' Code: 11
Prior Symbol: 'I' Symbol: 'r' Code: 101
Prior Symbol: 'I' Symbol: 's' Code: 010
Prior Symbol: 'J' Symbol: 27 Code: 1000
Prior Symbol: 'J' Symbol: '.' Code: 1001
Prior Symbol: 'J' Symbol: 'a' Code: 111
Prior Symbol: 'J' Symbol: 'e' Code: 1101
Prior Symbol: 'J' Symbol: 'i' Code: 1100
Prior Symbol: 'J' Symbol: 'o' Code: 0
Prior Symbol: 'J' Symbol: 'u' Code: 101
Prior Symbol: 'K' Symbol: 27 Code: 111
Prior Symbol: 'K' Symbol: 'a' Code: 100
Prior Symbol: 'K' Symbol: 'e' Code: 0
Prior Symbol: 'K' Symbol: 'i' Code: 101
Prior Symbol: 'K' Symbol: 'r' Code: 110
Prior Symbol: 'L' Symbol: 27 Code: 0110
Prior Symbol: 'L' Symbol: 'a' Code: 11
Prior Symbol: 'L' Symbol: 'e' Code: 00
Prior Symbol: 'L' Symbol: 'i' Code: 0111
Prior Symbol: 'L' Symbol: 'o' Code: 10
Prior Symbol: 'L' Symbol: 'u' Code: 010
Prior Symbol: 'M' Symbol: 27 Code: 11010
Prior Symbol: 'M' Symbol: 'a' Code: 0
Prior Symbol: 'M' Symbol: 'c' Code: 11011
Prior Symbol: 'M' Symbol: 'e' Code: 1111
Prior Symbol: 'M' Symbol: 'i' Code: 10
Prior Symbol: 'M' Symbol: 'o' Code: 1100
Prior Symbol: 'M' Symbol: 'u' Code: 1110
Prior Symbol: 'N' Symbol: 27 Code: 1100
Prior Symbol: 'N' Symbol: 'a' Code: 111
Prior Symbol: 'N' Symbol: 'e' Code: 0
Prior Symbol: 'N' Symbol: 'i' Code: 1101
Prior Symbol: 'N' Symbol: 'o' Code: 10
Prior Symbol: 'O' Symbol: 27 Code: 10
Prior Symbol: 'O' Symbol: ''' Code: 010
Prior Symbol: 'O' Symbol: 'l' Code: 110
Prior Symbol: 'O' Symbol: 'n' Code: 011
Prior Symbol: 'O' Symbol: 'r' Code: 111
Prior Symbol: 'O' Symbol: 's' Code: 00
Prior Symbol: 'P' Symbol: 27 Code: 10010
Prior Symbol: 'P' Symbol: 'a' Code: 0
Prior Symbol: 'P' Symbol: 'e' Code: 111
Prior Symbol: 'P' Symbol: 'h' Code: 10011
Prior Symbol: 'P' Symbol: 'i' Code: 1000
Prior Symbol: 'P' Symbol: 'l' Code: 1101
Prior Symbol: 'P' Symbol: 'o' Code: 101
Prior Symbol: 'P' Symbol: 'r' Code: 1100
Prior Symbol: 'Q' Symbol: 27 Code: 1
Prior Symbol: 'R' Symbol: 27 Code: 0000
Prior Symbol: 'R' Symbol: '.' Code: 0001
Prior Symbol: 'R' Symbol: 'a' Code: 01
Prior Symbol: 'R' Symbol: 'e' Code: 10
Prior Symbol: 'R' Symbol: 'i' Code: 001
Prior Symbol: 'R' Symbol: 'o' Code: 11
Prior Symbol: 'S' Symbol: 27 Code: 1011
Prior Symbol: 'S' Symbol: '.' Code: 0001
Prior Symbol: 'S' Symbol: 'a' Code: 100
Prior Symbol: 'S' Symbol: 'c' Code: 0010
Prior Symbol: 'S' Symbol: 'e' Code: 1110
Prior Symbol: 'S' Symbol: 'h' Code: 110
Prior Symbol: 'S' Symbol: 'i' Code: 0011
Prior Symbol: 'S' Symbol: 'o' Code: 1111
Prior Symbol: 'S' Symbol: 't' Code: 01
Prior Symbol: 'S' Symbol: 'u' Code: 1010
Prior Symbol: 'S' Symbol: 'v' Code: 00000
Prior Symbol: 'S' Symbol: 'y' Code: 00001
Prior Symbol: 'T' Symbol: 27 Code: 1010
Prior Symbol: 'T' Symbol: 'V' Code: 1000
Prior Symbol: 'T' Symbol: 'a' Code: 1001
Prior Symbol: 'T' Symbol: 'e' Code: 11010
Prior Symbol: 'T' Symbol: 'h' Code: 0
Prior Symbol: 'T' Symbol: 'i' Code: 1011
Prior Symbol: 'T' Symbol: 'o' Code: 111
Prior Symbol: 'T' Symbol: 'r' Code: 1100
Prior Symbol: 'T' Symbol: 'w' Code: 11011
Prior Symbol: 'U' Symbol: 27 Code: 10
Prior Symbol: 'U' Symbol: '.' Code: 0
Prior Symbol: 'U' Symbol: 'n' Code: 11
Prior Symbol: 'V' Symbol: 27 Code: 111
Prior Symbol: 'V' Symbol: ' ' Code: 10
Prior Symbol: 'V' Symbol: 'e' Code: 110
Prior Symbol: 'V' Symbol: 'i' Code: 0
Prior Symbol: 'W' Symbol: 27 Code: 010
Prior Symbol: 'W' Symbol: 'a' Code: 111
Prior Symbol: 'W' Symbol: 'e' Code: 110
Prior Symbol: 'W' Symbol: 'h' Code: 011
Prior Symbol: 'W' Symbol: 'i' Code: 10
Prior Symbol: 'W' Symbol: 'o' Code: 00
Prior Symbol: 'X' Symbol: 27 Code: 1
Prior Symbol: 'Y' Symbol: 27 Code: 0
Prior Symbol: 'Y' Symbol: 'o' Code: 1
Prior Symbol: 'Z' Symbol: 27 Code: 1
Prior Symbol: '[' Symbol: 27 Code: 1
Prior Symbol: '\' Symbol: 27 Code: 1
Prior Symbol: ']' Symbol: 27 Code: 1
Prior Symbol: '^' Symbol: 27 Code: 1
Prior Symbol: '_' Symbol: 27 Code: 1
Prior Symbol: '`' Symbol: 27 Code: 1
Prior Symbol: 'a' Symbol: 27 Code: 111001101
Prior Symbol: 'a' Symbol: ' ' Code: 101
Prior Symbol: 'a' Symbol: ''' Code: 111001110
Prior Symbol: 'a' Symbol: '.' Code: 1110010
Prior Symbol: 'a' Symbol: 'b' Code: 001011
Prior Symbol: 'a' Symbol: 'c' Code: 11001
Prior Symbol: 'a' Symbol: 'd' Code: 00111
Prior Symbol: 'a' Symbol: 'e' Code: 0011001
Prior Symbol: 'a' Symbol: 'f' Code: 001010
Prior Symbol: 'a' Symbol: 'g' Code: 00100
Prior Symbol: 'a' Symbol: 'h' Code: 001100010
Prior Symbol: 'a' Symbol: 'i' Code: 111000
Prior Symbol: 'a' Symbol: 'k' Code: 110000
Prior Symbol: 'a' Symbol: 'l' Code: 1101
Prior Symbol: 'a' Symbol: 'm' Code: 11101
Prior Symbol: 'a' Symbol: 'n' Code: 01
Prior Symbol: 'a' Symbol: 'o' Code: 001100011
Prior Symbol: 'a' Symbol: 'p' Code: 00000
Prior Symbol: 'a' Symbol: 'r' Code: 100
Prior Symbol: 'a' Symbol: 's' Code: 0001
Prior Symbol: 'a' Symbol: 't' Code: 1111
Prior Symbol: 'a' Symbol: 'u' Code: 110001
Prior Symbol: 'a' Symbol: 'v' Code: 001101
Prior Symbol: 'a' Symbol: 'w' Code: 111001111
Prior Symbol: 'a' Symbol: 'x' Code: 111001100
Prior Symbol: 'a' Symbol: 'y' Code: 00001
Prior Symbol: 'a' Symbol: 'z' Code: 00110000
Prior Symbol: 'b' Symbol: 27 Code: 101000
Prior Symbol: 'b' Symbol: ' ' Code: 0101
Prior Symbol: 'b' Symbol: '.' Code: 101001
Prior Symbol: 'b' Symbol: 'a' Code: 100
Prior Symbol: 'b' Symbol: 'b' Code: 101010
Prior Symbol: 'b' Symbol: 'd' Code: 1010110
Prior Symbol: 'b' Symbol: 'e' Code: 00
Prior Symbol: 'b' Symbol: 'i' Code: 1011
Prior Symbol: 'b' Symbol: 'l' Code: 0100
Prior Symbol: 'b' Symbol: 'o' Code: 110
Prior Symbol: 'b' Symbol: 'r' Code: 1110
Prior Symbol: 'b' Symbol: 's' Code: 1010111
Prior Symbol: 'b' Symbol: 'u' Code: 1111
Prior Symbol: 'b' Symbol: 'y' Code: 011
Prior Symbol: 'c' Symbol: 27 Code: 00010
Prior Symbol: 'c' Symbol: ' ' Code: 10000
Prior Symbol: 'c' Symbol: ',' Code: 010000
Prior Symbol: 'c' Symbol: '.' Code: 0100011
Prior Symbol: 'c' Symbol: 'D' Code: 0100110
Prior Symbol: 'c' Symbol: 'a' Code: 110
Prior Symbol: 'c' Symbol: 'c' Code: 010010
Prior Symbol: 'c' Symbol: 'e' Code: 011
Prior Symbol: 'c' Symbol: 'h' Code: 111
Prior Symbol: 'c' Symbol: 'i' Code: 0101
Prior Symbol: 'c' Symbol: 'k' Code: 1001
Prior Symbol: 'c' Symbol: 'l' Code: 10001
Prior Symbol: 'c' Symbol: 'o' Code: 101
Prior Symbol: 'c' Symbol: 'q' Code: 0100010
Prior Symbol: 'c' Symbol: 'r' Code: 00011
Prior Symbol: 'c' Symbol: 't' Code: 001
Prior Symbol: 'c' Symbol: 'u' Code: 0000
Prior Symbol: 'c' Symbol: 'y' Code: 0100111
Prior Symbol: 'd' Symbol: 27 Code: 1010001
Prior Symbol: 'd' Symbol: ' ' Code: 11
Prior Symbol: 'd' Symbol: ''' Code: 01111010
Prior Symbol: 'd' Symbol: ',' Code: 101011
Prior Symbol: 'd' Symbol: '.' Code: 0100
Prior Symbol: 'd' Symbol: ';' Code: 01111011
Prior Symbol: 'd' Symbol: 'a' Code: 1000
Prior Symbol: 'd' Symbol: 'd' Code: 01010
Prior Symbol: 'd' Symbol: 'e' Code: 00
Prior Symbol: 'd' Symbol: 'f' Code: 10100000
Prior Symbol: 'd' Symbol: 'g' Code: 10101011
Prior Symbol: 'd' Symbol: 'i' Code: 1011
Prior Symbol: 'd' Symbol: 'l' Code: 011111
Prior Symbol: 'd' Symbol: 'm' Code: 10100001
Prior Symbol: 'd' Symbol: 'n' Code: 1010100
Prior Symbol: 'd' Symbol: 'o' Code: 0110
Prior Symbol: 'd' Symbol: 'r' Code: 01110
Prior Symbol: 'd' Symbol: 's' Code: 1001
Prior Symbol: 'd' Symbol: 'u' Code: 101001

Prior Symbol: 'd' Symbol: 'v' Code: 0111100
Prior Symbol: 'd' Symbol: 'w' Code: 10101010
Prior Symbol: 'd' Symbol: 'y' Code: 01011
Prior Symbol: 'e' Symbol: 27 Code: 101110011
Prior Symbol: 'e' Symbol: ' ' Code: 111
Prior Symbol: 'e' Symbol: '"' Code: 10111010
Prior Symbol: 'e' Symbol: ')' Code: 100110000
Prior Symbol: 'e' Symbol: ',' Code: 000111
Prior Symbol: 'e' Symbol: '-' Code: 10011001
Prior Symbol: 'e' Symbol: '.' Code: 00110
Prior Symbol: 'e' Symbol: ';' Code: 10011010
Prior Symbol: 'e' Symbol: 'a' Code: 1000
Prior Symbol: 'e' Symbol: 'b' Code: 0001100
Prior Symbol: 'e' Symbol: 'c' Code: 10010
Prior Symbol: 'e' Symbol: 'd' Code: 0000
Prior Symbol: 'e' Symbol: 'e' Code: 10100
Prior Symbol: 'e' Symbol: 'f' Code: 10111011
Prior Symbol: 'e' Symbol: 'g' Code: 0001101
Prior Symbol: 'e' Symbol: 'h' Code: 100110001
Prior Symbol: 'e' Symbol: 'i' Code: 000100
Prior Symbol: 'e' Symbol: 'k' Code: 10011011
Prior Symbol: 'e' Symbol: 'l' Code: 0010
Prior Symbol: 'e' Symbol: 'm' Code: 100111
Prior Symbol: 'e' Symbol: 'n' Code: 010
Prior Symbol: 'e' Symbol: 'o' Code: 001110
Prior Symbol: 'e' Symbol: 'p' Code: 001111
Prior Symbol: 'e' Symbol: 'r' Code: 110
Prior Symbol: 'e' Symbol: 's' Code: 011
Prior Symbol: 'e' Symbol: 't' Code: 10101
Prior Symbol: 'e' Symbol: 'u' Code: 101110010
Prior Symbol: 'e' Symbol: 'v' Code: 101100
Prior Symbol: 'e' Symbol: 'w' Code: 101111
Prior Symbol: 'e' Symbol: 'x' Code: 000101
Prior Symbol: 'e' Symbol: 'y' Code: 101101
Prior Symbol: 'e' Symbol: 'z' Code: 10111000
Prior Symbol: 'f' Symbol: 27 Code: 1110111
Prior Symbol: 'f' Symbol: ' ' Code: 10
Prior Symbol: 'f' Symbol: '.' Code: 1110110
Prior Symbol: 'f' Symbol: 'a' Code: 1111
Prior Symbol: 'f' Symbol: 'e' Code: 000
Prior Symbol: 'f' Symbol: 'f' Code: 0101
Prior Symbol: 'f' Symbol: 'i' Code: 001
Prior Symbol: 'f' Symbol: 'l' Code: 111010
Prior Symbol: 'f' Symbol: 'o' Code: 110
Prior Symbol: 'f' Symbol: 'r' Code: 011
Prior Symbol: 'f' Symbol: 't' Code: 0100
Prior Symbol: 'f' Symbol: 'u' Code: 11100
Prior Symbol: 'g' Symbol: 27 Code: 1111010
Prior Symbol: 'g' Symbol: ' ' Code: 10
Prior Symbol: 'g' Symbol: '"' Code: 1111011
Prior Symbol: 'g' Symbol: ',' Code: 111110
Prior Symbol: 'g' Symbol: '-' Code: 0101010
Prior Symbol: 'g' Symbol: '.' Code: 01011
Prior Symbol: 'g' Symbol: 'a' Code: 1110
Prior Symbol: 'g' Symbol: 'e' Code: 00
Prior Symbol: 'g' Symbol: 'g' Code: 0101011
Prior Symbol: 'g' Symbol: 'h' Code: 011
Prior Symbol: 'g' Symbol: 'i' Code: 1101
Prior Symbol: 'g' Symbol: 'l' Code: 111100
Prior Symbol: 'g' Symbol: 'o' Code: 0100
Prior Symbol: 'g' Symbol: 'r' Code: 111111
Prior Symbol: 'g' Symbol: 's' Code: 11000
Prior Symbol: 'g' Symbol: 'u' Code: 11001
Prior Symbol: 'g' Symbol: 'y' Code: 010100
Prior Symbol: 'h' Symbol: 27 Code: 1011100
Prior Symbol: 'h' Symbol: ' ' Code: 100
Prior Symbol: 'h' Symbol: '"' Code: 10101000
Prior Symbol: 'h' Symbol: ',' Code: 10101001
Prior Symbol: 'h' Symbol: '-' Code: 10101011
Prior Symbol: 'h' Symbol: '.' Code: 101001
Prior Symbol: 'h' Symbol: 'a' Code: 011
Prior Symbol: 'h' Symbol: 'e' Code: 11
Prior Symbol: 'h' Symbol: 'i' Code: 00
Prior Symbol: 'h' Symbol: 'n' Code: 101011
Prior Symbol: 'h' Symbol: 'o' Code: 010
Prior Symbol: 'h' Symbol: 'r' Code: 101111
Prior Symbol: 'h' Symbol: 's' Code: 10101010
Prior Symbol: 'h' Symbol: 't' Code: 10110
Prior Symbol: 'h' Symbol: 'u' Code: 101000
Prior Symbol: 'h' Symbol: 'y' Code: 1011101
Prior Symbol: 'i' Symbol: 27 Code: 00011101
Prior Symbol: 'i' Symbol: ' ' Code: 0001111
Prior Symbol: 'i' Symbol: ',' Code: 100110100
Prior Symbol: 'i' Symbol: '.' Code: 10011000
Prior Symbol: 'i' Symbol: 'a' Code: 11010
Prior Symbol: 'i' Symbol: 'b' Code: 100110101
Prior Symbol: 'i' Symbol: 'c' Code: 1111
Prior Symbol: 'i' Symbol: 'd' Code: 10000
Prior Symbol: 'i' Symbol: 'e' Code: 1110
Prior Symbol: 'i' Symbol: 'f' Code: 100111
Prior Symbol: 'i' Symbol: 'g' Code: 10010
Prior Symbol: 'i' Symbol: 'k' Code: 10011011
Prior Symbol: 'i' Symbol: 'l' Code: 1100
Prior Symbol: 'i' Symbol: 'm' Code: 10001
Prior Symbol: 'i' Symbol: 'n' Code: 01
Prior Symbol: 'i' Symbol: 'o' Code: 11011
Prior Symbol: 'i' Symbol: 'p' Code: 000110
Prior Symbol: 'i' Symbol: 'r' Code: 0000
Prior Symbol: 'i' Symbol: 's' Code: 101
Prior Symbol: 'i' Symbol: 't' Code: 001
Prior Symbol: 'i' Symbol: 'v' Code: 00010
Prior Symbol: 'i' Symbol: 'x' Code: 00011100
Prior Symbol: 'i' Symbol: 'z' Code: 10011001
Prior Symbol: 'j' Symbol: 27 Code: 000
Prior Symbol: 'j' Symbol: 'a' Code: 001
Prior Symbol: 'j' Symbol: 'e' Code: 010
Prior Symbol: 'j' Symbol: 'o' Code: 1
Prior Symbol: 'j' Symbol: 'u' Code: 011
Prior Symbol: 'k' Symbol: 27 Code: 0000
Prior Symbol: 'k' Symbol: ' ' Code: 01
Prior Symbol: 'k' Symbol: '"' Code: 10000
Prior Symbol: 'k' Symbol: ',' Code: 10011
Prior Symbol: 'k' Symbol: '.' Code: 0001
Prior Symbol: 'k' Symbol: 'e' Code: 11
Prior Symbol: 'k' Symbol: 'i' Code: 101
Prior Symbol: 'k' Symbol: 'l' Code: 100100
Prior Symbol: 'k' Symbol: 'n' Code: 10001
Prior Symbol: 'k' Symbol: 's' Code: 001
Prior Symbol: 'k' Symbol: 'y' Code: 100101
Prior Symbol: 'l' Symbol: 27 Code: 0011100
Prior Symbol: 'l' Symbol: ' ' Code: 110
Prior Symbol: 'l' Symbol: '"' Code: 00111100
Prior Symbol: 'l' Symbol: ',' Code: 001101
Prior Symbol: 'l' Symbol: '-' Code: 00111101
Prior Symbol: 'l' Symbol: '.' Code: 00100
Prior Symbol: 'l' Symbol: 'a' Code: 000
Prior Symbol: 'l' Symbol: 'b' Code: 0011101
Prior Symbol: 'l' Symbol: 'c' Code: 00111111
Prior Symbol: 'l' Symbol: 'd' Code: 10111
Prior Symbol: 'l' Symbol: 'e' Code: 111
Prior Symbol: 'l' Symbol: 'f' Code: 010110
Prior Symbol: 'l' Symbol: 'i' Code: 011
Prior Symbol: 'l' Symbol: 'k' Code: 10110110
Prior Symbol: 'l' Symbol: 'l' Code: 100
Prior Symbol: 'l' Symbol: 'm' Code: 010111
Prior Symbol: 'l' Symbol: 'n' Code: 00111110
Prior Symbol: 'l' Symbol: 'o' Code: 1010
Prior Symbol: 'l' Symbol: 'p' Code: 00101
Prior Symbol: 'l' Symbol: 'r' Code: 10110111
Prior Symbol: 'l' Symbol: 's' Code: 01010
Prior Symbol: 'l' Symbol: 't' Code: 001100
Prior Symbol: 'l' Symbol: 'u' Code: 1011010
Prior Symbol: 'l' Symbol: 'v' Code: 101100
Prior Symbol: 'l' Symbol: 'y' Code: 0100
Prior Symbol: 'm' Symbol: 27 Code: 101010
Prior Symbol: 'm' Symbol: ' ' Code: 111
Prior Symbol: 'm' Symbol: '"' Code: 1010110
Prior Symbol: 'm' Symbol: '.' Code: 110101
Prior Symbol: 'm' Symbol: ';' Code: 1010111
Prior Symbol: 'm' Symbol: 'a' Code: 00
Prior Symbol: 'm' Symbol: 'b' Code: 10100
Prior Symbol: 'm' Symbol: 'e' Code: 01
Prior Symbol: 'm' Symbol: 'i' Code: 1100
Prior Symbol: 'm' Symbol: 'm' Code: 10110
Prior Symbol: 'm' Symbol: 'o' Code: 1000
Prior Symbol: 'm' Symbol: 'p' Code: 1001
Prior Symbol: 'm' Symbol: 's' Code: 10111
Prior Symbol: 'm' Symbol: 'u' Code: 11011
Prior Symbol: 'm' Symbol: 'y' Code: 110100
Prior Symbol: 'n' Symbol: 27 Code: 0100000
Prior Symbol: 'n' Symbol: ' ' Code: 10
Prior Symbol: 'n' Symbol: '"' Code: 0100011
Prior Symbol: 'n' Symbol: ',' Code: 111100
Prior Symbol: 'n' Symbol: '-' Code: 011011010
Prior Symbol: 'n' Symbol: '.' Code: 01100
Prior Symbol: 'n' Symbol: ';' Code: 011011011
Prior Symbol: 'n' Symbol: 'a' Code: 11111
Prior Symbol: 'n' Symbol: 'b' Code: 011011100
Prior Symbol: 'n' Symbol: 'c' Code: 01001
Prior Symbol: 'n' Symbol: 'd' Code: 110
Prior Symbol: 'n' Symbol: 'e' Code: 001
Prior Symbol: 'n' Symbol: 'f' Code: 01000101
Prior Symbol: 'n' Symbol: 'g' Code: 000
Prior Symbol: 'n' Symbol: 'i' Code: 01111
Prior Symbol: 'n' Symbol: 'j' Code: 011011101
Prior Symbol: 'n' Symbol: 'k' Code: 1111010
Prior Symbol: 'n' Symbol: 'l' Code: 01101100
Prior Symbol: 'n' Symbol: 'm' Code: 011011110
Prior Symbol: 'n' Symbol: 'n' Code: 01110
Prior Symbol: 'n' Symbol: 'o' Code: 1111011
Prior Symbol: 'n' Symbol: 'r' Code: 011011111
Prior Symbol: 'n' Symbol: 's' Code: 0101
Prior Symbol: 'n' Symbol: 't' Code: 1110
Prior Symbol: 'n' Symbol: 'u' Code: 0100001
Prior Symbol: 'n' Symbol: 'v' Code: 0110100
Prior Symbol: 'n' Symbol: 'y' Code: 0110101
Prior Symbol: 'n' Symbol: 'z' Code: 01000100
Prior Symbol: 'o' Symbol: 27 Code: 101010011
Prior Symbol: 'o' Symbol: ' ' Code: 001
Prior Symbol: 'o' Symbol: ',' Code: 01001111
Prior Symbol: 'o' Symbol: '-' Code: 01001110
Prior Symbol: 'o' Symbol: '.' Code: 0100110
Prior Symbol: 'o' Symbol: 'B' Code: 101010010
Prior Symbol: 'o' Symbol: 'a' Code: 100001
Prior Symbol: 'o' Symbol: 'b' Code: 110111
Prior Symbol: 'o' Symbol: 'c' Code: 100000
Prior Symbol: 'o' Symbol: 'd' Code: 110101
Prior Symbol: 'o' Symbol: 'e' Code: 1010101
Prior Symbol: 'o' Symbol: 'f' Code: 000
Prior Symbol: 'o' Symbol: 'g' Code: 1101000
Prior Symbol: 'o' Symbol: 'h' Code: 1101001
Prior Symbol: 'o' Symbol: 'i' Code: 1101101
Prior Symbol: 'o' Symbol: 'k' Code: 010010
Prior Symbol: 'o' Symbol: 'l' Code: 0101
Prior Symbol: 'o' Symbol: 'm' Code: 1100
Prior Symbol: 'o' Symbol: 'n' Code: 111
Prior Symbol: 'o' Symbol: 'o' Code: 10100
Prior Symbol: 'o' Symbol: 'p' Code: 01000
Prior Symbol: 'o' Symbol: 'r' Code: 011
Prior Symbol: 'o' Symbol: 's' Code: 10001
Prior Symbol: 'o' Symbol: 't' Code: 10010
Prior Symbol: 'o' Symbol: 'u' Code: 1011
Prior Symbol: 'o' Symbol: 'v' Code: 101011
Prior Symbol: 'o' Symbol: 'w' Code: 10011
Prior Symbol: 'o' Symbol: 'x' Code: 10101000
Prior Symbol: 'o' Symbol: 'y' Code: 1101100
Prior Symbol: 'p' Symbol: 27 Code: 011011
Prior Symbol: 'p' Symbol: ' ' Code: 000
Prior Symbol: 'p' Symbol: '-' Code: 1010010
Prior Symbol: 'p' Symbol: '.' Code: 101000
Prior Symbol: 'p' Symbol: 'a' Code: 001
Prior Symbol: 'p' Symbol: 'e' Code: 110
Prior Symbol: 'p' Symbol: 'h' Code: 1111
Prior Symbol: 'p' Symbol: 'i' Code: 1011
Prior Symbol: 'p' Symbol: 'l' Code: 010
Prior Symbol: 'p' Symbol: 'm' Code: 1010011
Prior Symbol: 'p' Symbol: 'o' Code: 0111
Prior Symbol: 'p' Symbol: 'p' Code: 11101
Prior Symbol: 'p' Symbol: 'r' Code: 100
Prior Symbol: 'p' Symbol: 's' Code: 01100
Prior Symbol: 'p' Symbol: 't' Code: 11100
Prior Symbol: 'p' Symbol: 'u' Code: 10101
Prior Symbol: 'p' Symbol: 'y' Code: 011010
Prior Symbol: 'q' Symbol: 27 Code: 0
Prior Symbol: 'q' Symbol: 'u' Code: 1
Prior Symbol: 'r' Symbol: 27 Code: 10011111
Prior Symbol: 'r' Symbol: ' ' Code: 111
Prior Symbol: 'r' Symbol: '"' Code: 1001110
Prior Symbol: 'r' Symbol: ')' Code: 100111100
Prior Symbol: 'r' Symbol: ',' Code: 100100

Prior Symbol: 'r' Symbol: '-' Code: 11001100
Prior Symbol: 'r' Symbol: '.' Code: 10001
Prior Symbol: 'r' Symbol: ';' Code: 100111101
Prior Symbol: 'r' Symbol: 'a' Code: 1101
Prior Symbol: 'r' Symbol: 'b' Code: 11001101
Prior Symbol: 'r' Symbol: 'c' Code: 100001
Prior Symbol: 'r' Symbol: 'd' Code: 11000
Prior Symbol: 'r' Symbol: 'e' Code: 101
Prior Symbol: 'r' Symbol: 'f' Code: 110011111
Prior Symbol: 'r' Symbol: 'g' Code: 100101
Prior Symbol: 'r' Symbol: 'i' Code: 010
Prior Symbol: 'r' Symbol: 'k' Code: 110010
Prior Symbol: 'r' Symbol: 'l' Code: 00100
Prior Symbol: 'r' Symbol: 'm' Code: 00101
Prior Symbol: 'r' Symbol: 'n' Code: 01100
Prior Symbol: 'r' Symbol: 'o' Code: 000
Prior Symbol: 'r' Symbol: 'p' Code: 11001110
Prior Symbol: 'r' Symbol: 'r' Code: 100110
Prior Symbol: 'r' Symbol: 's' Code: 0111
Prior Symbol: 'r' Symbol: 't' Code: 0011
Prior Symbol: 'r' Symbol: 'u' Code: 100000
Prior Symbol: 'r' Symbol: 'v' Code: 110011110
Prior Symbol: 'r' Symbol: 'y' Code: 01101
Prior Symbol: 's' Symbol: 27 Code: 10011100
Prior Symbol: 's' Symbol: ' ' Code: 0
Prior Symbol: 's' Symbol: '"' Code: 100111100
Prior Symbol: 's' Symbol: ''' Code: 100111101
Prior Symbol: 's' Symbol: ',' Code: 111011
Prior Symbol: 's' Symbol: '.' Code: 1000
Prior Symbol: 's' Symbol: ';' Code: 11101011
Prior Symbol: 's' Symbol: 'a' Code: 110011
Prior Symbol: 's' Symbol: 'b' Code: 100111110
Prior Symbol: 's' Symbol: 'c' Code: 10010
Prior Symbol: 's' Symbol: 'e' Code: 1101
Prior Symbol: 's' Symbol: 'h' Code: 11000
Prior Symbol: 's' Symbol: 'i' Code: 11100
Prior Symbol: 's' Symbol: 'k' Code: 100111111
Prior Symbol: 's' Symbol: 'l' Code: 1110100
Prior Symbol: 's' Symbol: 'm' Code: 111010100
Prior Symbol: 's' Symbol: 'n' Code: 111010101
Prior Symbol: 's' Symbol: 'o' Code: 11110
Prior Symbol: 's' Symbol: 'p' Code: 1001101
Prior Symbol: 's' Symbol: 's' Code: 11111
Prior Symbol: 's' Symbol: 't' Code: 101
Prior Symbol: 's' Symbol: 'u' Code: 110010
Prior Symbol: 's' Symbol: 'w' Code: 10011101
Prior Symbol: 's' Symbol: 'y' Code: 1001100
Prior Symbol: 't' Symbol: 27 Code: 11000011
Prior Symbol: 't' Symbol: ' ' Code: 111
Prior Symbol: 't' Symbol: ''' Code: 11000100
Prior Symbol: 't' Symbol: ',' Code: 0111100
Prior Symbol: 't' Symbol: '-' Code: 01111110
Prior Symbol: 't' Symbol: '.' Code: 01101
Prior Symbol: 't' Symbol: ';' Code: 110000100
Prior Symbol: 't' Symbol: 'a' Code: 0100
Prior Symbol: 't' Symbol: 'b' Code: 110000101
Prior Symbol: 't' Symbol: 'c' Code: 11000101
Prior Symbol: 't' Symbol: 'e' Code: 101
Prior Symbol: 't' Symbol: 'h' Code: 00
Prior Symbol: 't' Symbol: 'i' Code: 1101
Prior Symbol: 't' Symbol: 'l' Code: 0111101
Prior Symbol: 't' Symbol: 'm' Code: 01111111
Prior Symbol: 't' Symbol: 'n' Code: 0111110
Prior Symbol: 't' Symbol: 'o' Code: 100
Prior Symbol: 't' Symbol: 'r' Code: 11001
Prior Symbol: 't' Symbol: 's' Code: 0101
Prior Symbol: 't' Symbol: 't' Code: 01100
Prior Symbol: 't' Symbol: 'u' Code: 01110
Prior Symbol: 't' Symbol: 'w' Code: 1100000
Prior Symbol: 't' Symbol: 'y' Code: 1100011
Prior Symbol: 'u' Symbol: 27 Code: 1001100
Prior Symbol: 'u' Symbol: ' ' Code: 100000
Prior Symbol: 'u' Symbol: 'a' Code: 100111
Prior Symbol: 'u' Symbol: 'b' Code: 100001
Prior Symbol: 'u' Symbol: 'c' Code: 10001
Prior Symbol: 'u' Symbol: 'd' Code: 11100
Prior Symbol: 'u' Symbol: 'e' Code: 11101
Prior Symbol: 'u' Symbol: 'g' Code: 11110
Prior Symbol: 'u' Symbol: 'i' Code: 10010
Prior Symbol: 'u' Symbol: 'k' Code: 1001101
Prior Symbol: 'u' Symbol: 'l' Code: 0100
Prior Symbol: 'u' Symbol: 'm' Code: 111111
Prior Symbol: 'u' Symbol: 'n' Code: 110
Prior Symbol: 'u' Symbol: 'o' Code: 11111010
Prior Symbol: 'u' Symbol: 'p' Code: 0101
Prior Symbol: 'u' Symbol: 'r' Code: 00
Prior Symbol: 'u' Symbol: 's' Code: 011
Prior Symbol: 'u' Symbol: 't' Code: 101
Prior Symbol: 'u' Symbol: 'v' Code: 11111011
Prior Symbol: 'u' Symbol: 'y' Code: 1111100
Prior Symbol: 'v' Symbol: 27 Code: 00010
Prior Symbol: 'v' Symbol: 'a' Code: 001
Prior Symbol: 'v' Symbol: 'e' Code: 1
Prior Symbol: 'v' Symbol: 'i' Code: 01
Prior Symbol: 'v' Symbol: 'o' Code: 0000
Prior Symbol: 'v' Symbol: 's' Code: 000110
Prior Symbol: 'v' Symbol: 'y' Code: 000111
Prior Symbol: 'w' Symbol: 27 Code: 011101
Prior Symbol: 'w' Symbol: ' ' Code: 001
Prior Symbol: 'w' Symbol: '.' Code: 011100
Prior Symbol: 'w' Symbol: 'a' Code: 010
Prior Symbol: 'w' Symbol: 'e' Code: 1110
Prior Symbol: 'w' Symbol: 'h' Code: 000
Prior Symbol: 'w' Symbol: 'i' Code: 10
Prior Symbol: 'w' Symbol: 'l' Code: 011110
Prior Symbol: 'w' Symbol: 'm' Code: 011111
Prior Symbol: 'w' Symbol: 'n' Code: 11111
Prior Symbol: 'w' Symbol: 'o' Code: 110
Prior Symbol: 'w' Symbol: 'r' Code: 0110
Prior Symbol: 'w' Symbol: 's' Code: 11110
Prior Symbol: 'x' Symbol: 27 Code: 10
Prior Symbol: 'x' Symbol: ' ' Code: 0110
Prior Symbol: 'x' Symbol: ',' Code: 0111
Prior Symbol: 'x' Symbol: '-' Code: 1100
Prior Symbol: 'x' Symbol: 'a' Code: 111
Prior Symbol: 'x' Symbol: 'e' Code: 00
Prior Symbol: 'x' Symbol: 'i' Code: 010
Prior Symbol: 'x' Symbol: 't' Code: 1101
Prior Symbol: 'y' Symbol: 27 Code: 01010
Prior Symbol: 'y' Symbol: ' ' Code: 1
Prior Symbol: 'y' Symbol: ''' Code: 010010
Prior Symbol: 'y' Symbol: ',' Code: 0001
Prior Symbol: 'y' Symbol: '.' Code: 0111
Prior Symbol: 'y' Symbol: ';' Code: 011001
Prior Symbol: 'y' Symbol: '?' Code: 0100110
Prior Symbol: 'y' Symbol: 'a' Code: 0100111
Prior Symbol: 'y' Symbol: 'b' Code: 0110000
Prior Symbol: 'y' Symbol: 'd' Code: 000001
Prior Symbol: 'y' Symbol: 'e' Code: 0010
Prior Symbol: 'y' Symbol: 'f' Code: 0110001
Prior Symbol: 'y' Symbol: 'i' Code: 000010
Prior Symbol: 'y' Symbol: 'l' Code: 01000
Prior Symbol: 'y' Symbol: 'm' Code: 000000
Prior Symbol: 'y' Symbol: 'n' Code: 01011
Prior Symbol: 'y' Symbol: 'o' Code: 01101
Prior Symbol: 'y' Symbol: 's' Code: 0011
Prior Symbol: 'y' Symbol: 'w' Code: 000011
Prior Symbol: 'z' Symbol: 27 Code: 100
Prior Symbol: 'z' Symbol: ' ' Code: 1110
Prior Symbol: 'z' Symbol: '.' Code: 1111
Prior Symbol: 'z' Symbol: 'a' Code: 000
Prior Symbol: 'z' Symbol: 'e' Code: 001
Prior Symbol: 'z' Symbol: 'i' Code: 110
Prior Symbol: 'z' Symbol: 'l' Code: 010
Prior Symbol: 'z' Symbol: 'o' Code: 101
Prior Symbol: 'z' Symbol: 'z' Code: 011
Prior Symbol: '{' Symbol: 27 Code: 1
Prior Symbol: '|' Symbol: 27 Code: 1
Prior Symbol: '}' Symbol: 27 Code: 1
Prior Symbol: '~' Symbol: 27 Code: 1
Prior Symbol: 127 Symbol: 27 Code: 1

Table C7 English-language Program Description Decode Table

0	1	76	1	152	2	228	5	304	155	380	3	456	36
1	0	77	240	153	242	229	208	305	155	381	4	457	37
2	1	78	1	154	2	230	6	306	155	382	5	458	38
3	44	79	242	155	252	231	6	307	155	383	207	459	39
4	1	80	1	156	3	232	6	308	155	384	6	460	40
5	46	81	248	157	8	233	52	309	155	385	201	461	41
6	1	82	1	158	3	234	6	310	155	386	249	462	42
7	48	83	250	159	16	235	96	311	155	387	234	463	244
8	1	84	1	160	3	236	6	312	155	388	235	464	43
9	50	85	252	161	26	237	134	313	155	389	245	465	44
10	1	86	1	162	3	238	6	314	155	390	246	466	45
11	52	87	254	163	40	239	146	315	155	391	7	467	46
12	1	88	2	164	3	240	6	316	155	392	8	468	47
13	54	89	0	165	42	241	170	317	155	393	9	469	225
14	1	90	2	166	3	242	6	318	155	394	178	470	48
15	56	91	4	167	52	243	184	319	155	395	197	471	49
16	1	92	2	168	3	244	6	320	155	396	198	472	50
17	58	93	22	169	74	245	220	321	155	397	177	473	51
18	1	94	2	170	3	246	6	322	155	398	10	474	52
19	60	95	32	171	90	247	236	323	155	399	238	475	53
20	1	96	2	172	3	248	6	324	155	400	203	476	54
21	62	97	34	173	94	249	238	325	155	401	11	477	55
22	1	98	2	174	3	250	6	326	155	402	212	478	155
23	64	99	44	175	100	251	240	327	155	403	12	479	155
24	1	100	2	176	3	252	6	328	155	404	196	480	3
25	66	101	50	177	110	253	242	329	155	405	200	481	4
26	1	102	2	178	3	254	6	330	155	406	210	482	128
27	68	103	56	179	112	255	244	331	155	407	13	483	174
28	1	104	2	180	3	256	20	332	155	408	14	484	200
29	70	105	60	181	114	257	21	333	155	409	15	485	212
30	1	106	2	182	3	258	155	334	155	410	199	486	1
31	72	107	64	183	116	259	214	335	155	411	202	487	2
32	1	108	2	184	3	260	201	336	155	412	206	488	155
33	74	109	68	185	118	261	207	337	155	413	208	489	160
34	1	110	2	186	3	262	215	338	155	414	215	490	155
35	76	111	70	187	120	263	199	339	155	415	16	491	155
36	1	112	2	188	3	264	1	340	155	416	194	492	155
37	78	113	74	189	122	265	162	341	155	417	17	493	155
38	1	114	2	190	3	266	206	342	155	418	204	494	155
39	80	115	76	191	124	267	203	343	155	419	236	495	155
40	1	116	2	192	3	268	2	344	155	420	229	496	155
41	82	117	84	193	126	269	3	345	155	421	231	497	155
42	1	118	2	194	3	270	197	346	155	422	18	498	2
43	84	119	86	195	128	271	204	347	155	423	205	499	243
44	1	120	2	196	3	272	198	348	155	424	19	500	160
45	86	121	88	197	180	273	200	349	155	425	20	501	244
46	1	122	2	198	3	274	4	350	155	426	195	502	155
47	88	123	90	199	206	275	196	351	155	427	21	503	1
48	1	124	2	200	3	276	5	352	155	428	22	504	155
49	90	125	92	201	240	277	194	353	155	429	23	505	155
50	1	126	2	202	4	278	6	354	155	430	237	506	172
51	92	127	94	203	26	279	195	355	155	431	24	507	155
52	1	128	2	204	4	280	210	356	155	432	25	508	155
53	94	129	96	205	88	281	7	357	155	433	242	509	155
54	1	130	2	206	4	282	211	358	155	434	26	510	155
55	96	131	98	207	110	283	8	359	155	435	211	511	155
56	1	132	2	208	4	284	202	360	155	436	27	512	1
57	98	133	118	209	142	285	212	361	155	437	28	513	160
58	1	134	2	210	4	286	9	362	56	438	228	514	155
59	100	135	132	211	172	287	205	363	57	439	29	515	162
60	1	136	2	212	4	288	208	364	173	440	193	516	7
61	102	137	148	213	216	289	10	365	175	441	227	517	8
62	1	138	2	214	4	290	193	366	183	442	30	518	226
63	104	139	162	215	224	291	11	367	218	443	233	519	228
64	1	140	2	216	4	292	12	368	168	444	240	520	229
65	106	141	178	217	244	293	13	369	179	445	226	521	230
66	1	142	2	218	5	294	14	370	181	446	247	522	160
67	222	143	186	219	36	295	15	371	1	447	31	523	242
68	1	144	2	220	5	296	16	372	2	448	243	524	225
69	224	145	200	221	64	297	17	373	155	449	230	525	1
70	1	146	2	222	5	298	18	374	180	450	32	526	2
71	234	147	210	223	118	299	19	375	241	451	33	527	243
72	1	148	2	224	5	300	155	376	162	452	34	528	227
73	236	149	222	225	174	301	155	377	213	453	232	529	3
74	1	150	2	226	5	302	155	378	214	454	239	530	4
75	238	151	234	227	206	303	155	379	217	455	35	531	5

532	155	612	230	692	229	772	2	852	2	932	13	1012	230
533	6	613	245	693	236	773	3	853	3	933	14	1013	237
534	4	614	243	694	155	774	233	854	4	934	15	1014	247
535	128	615	244	695	239	775	4	855	239	935	236	1015	231
536	202	616	155	696	1	776	229	856	5	936	16	1016	246
537	211	617	228	697	242	777	3	857	6	937	244	1017	1
538	162	618	1	698	5	778	155	858	174	938	17	1018	2
539	1	619	237	699	6	779	233	859	1	939	18	1019	155
540	155	620	2	700	245	780	1	860	155	940	242	1020	238
541	2	621	3	701	239	781	225	861	238	941	160	1021	3
542	3	622	4	702	155	782	239	862	233	942	19	1022	4
543	160	623	242	703	236	783	2	863	2	943	20	1023	236
544	155	624	5	704	233	784	3	864	229	944	21	1024	5
545	160	625	6	705	1	785	4	865	155	945	238	1025	245
546	3	626	236	706	225	786	167	866	160	946	22	1026	6
547	4	627	238	707	242	787	238	867	1	947	23	1027	172
548	155	628	7	708	2	788	236	868	3	948	11	1028	228
549	183	629	160	709	229	789	242	869	4	949	12	1029	249
550	244	630	5	710	3	790	243	870	155	950	228	1030	242
551	160	631	6	711	4	791	1	871	232	951	243	1031	7
552	176	632	155	712	3	792	155	872	229	952	155	1032	8
553	243	633	236	713	4	793	2	873	225	953	174	1033	9
554	1	634	245	714	155	794	225	874	239	954	226	1034	174
555	2	635	1	715	229	795	6	875	1	955	1	1035	10
556	185	636	2	716	233	796	155	876	233	956	2	1036	239
557	2	637	225	717	245	797	232	877	2	957	3	1037	11
558	184	638	239	718	225	798	233	878	155	958	236	1038	225
559	155	639	229	719	1	799	1	879	155	959	160	1039	243
560	160	640	233	720	239	800	242	880	155	960	4	1040	12
561	1	641	242	721	2	801	236	881	239	961	233	1041	233
562	174	642	3	722	4	802	2	882	155	962	242	1042	13
563	2	643	4	723	5	803	239	883	155	963	245	1043	14
564	182	644	6	724	160	804	3	884	155	964	5	1044	15
565	155	645	7	725	201	805	229	885	155	965	249	1045	16
566	1	646	155	726	243	806	4	886	155	966	225	1046	229
567	160	647	233	727	155	807	5	887	155	967	6	1047	17
568	160	648	249	728	174	808	155	888	155	968	239	1048	18
569	1	649	242	729	242	809	155	889	155	969	7	1049	160
570	155	650	245	730	1	810	3	890	155	970	229	1050	29
571	176	651	1	731	2	811	4	891	155	971	8	1051	30
572	174	652	2	732	3	812	155	892	155	972	9	1052	169
573	1	653	3	733	238	813	174	893	155	973	10	1053	232
574	155	654	236	734	239	814	1	894	155	974	15	1054	245
575	160	655	239	735	5	815	233	895	155	975	16	1055	155
576	174	656	225	736	155	816	2	896	24	976	241	1056	1
577	1	657	4	737	174	817	225	897	25	977	174	1057	173
578	160	658	232	738	233	818	229	898	232	978	196	1058	187
579	155	659	5	739	229	819	239	899	239	979	249	1059	235
580	155	660	5	740	1	820	9	900	248	980	172	1060	250
581	155	661	6	741	245	821	10	901	155	981	1	1061	2
582	155	662	249	742	2	822	246	902	167	982	227	1062	167
583	1	663	242	743	225	823	249	903	247	983	2	1063	230
584	172	664	245	744	3	824	1	904	250	984	155	1064	226
585	174	665	155	745	4	825	174	905	1	985	242	1065	231
586	155	666	229	746	229	826	227	906	2	986	3	1066	3
587	155	667	239	747	3	827	233	907	3	987	4	1067	4
588	2	668	1	748	225	828	245	908	4	988	160	1068	5
589	3	669	2	749	233	829	155	909	229	989	236	1069	6
590	155	670	233	750	242	830	229	910	174	990	245	1070	233
591	160	671	225	751	155	831	239	911	5	991	5	1071	248
592	181	672	3	752	1	832	2	912	230	992	6	1072	7
593	182	673	4	753	2	833	3	913	226	993	233	1073	172
594	184	674	6	754	3	834	225	914	6	994	7	1074	239
595	1	675	7	755	4	835	4	915	246	995	235	1075	240
596	155	676	225	756	155	836	232	916	235	996	8	1076	8
597	160	677	233	757	233	837	5	917	245	997	244	1077	237
598	155	678	238	758	245	838	6	918	233	998	9	1078	246
599	160	679	246	759	1	839	244	919	7	999	229	1079	249
600	155	680	228	760	229	840	7	920	240	1000	10	1080	9
601	155	681	236	761	2	841	8	921	249	1001	239	1081	247
602	155	682	243	762	239	842	232	922	231	1002	225	1082	10
603	155	683	1	763	225	843	7	923	8	1003	232	1083	11
604	155	684	2	764	225	844	229	924	9	1004	11	1084	174
605	155	685	242	765	5	845	247	925	228	1005	12	1085	12
606	155	686	3	766	155	846	214	926	10	1006	13	1086	227
607	160	687	4	767	227	847	225	927	227	1007	14	1087	13
608	155	688	155	768	239	848	155	928	11	1008	19	1088	229
609	155	689	5	769	1	849	233	929	237	1009	20	1089	244
610	8	690	2	770	245	850	242	930	12	1010	167	1090	14
611	9	691	3	771	229	851	1	931	243	1011	187	1091	15

1092	228
1093	16
1094	236
1095	17
1096	225
1097	18
1098	19
1099	20
1100	21
1101	22
1102	238
1103	243
1104	23
1105	24
1106	242
1107	160
1108	25
1109	26
1110	27
1111	28
1112	9
1113	10
1114	174
1115	155
1116	236
1117	1
1118	245
1119	2
1120	244
1121	230
1122	3
1123	225
1124	229
1125	233
1126	4
1127	242
1128	239
1129	5
1130	6
1131	7
1132	160
1133	8
1134	14
1135	15
1136	173
1137	231
1138	155
1139	167
1140	249
1141	1
1142	236
1143	2
1144	172
1145	242
1146	3
1147	174
1148	243
1149	245
1150	4
1151	5
1152	239
1153	6
1154	7
1155	233
1156	225
1157	8
1158	9
1159	232
1160	10
1161	11
1162	229
1163	12
1164	160
1165	13
1166	13
1167	14
1168	167
1169	172
1170	243
1171	173
1172	1
1173	2
1174	155
1175	249
1176	245
1177	174
1178	3
1179	238
1180	4
1181	242
1182	5
1183	6
1184	244
1185	7
1186	8
1187	9
1188	239
1189	225
1190	160
1191	10
1192	233
1193	11
1194	12
1195	229
1196	20
1197	21
1198	172
1199	226
1200	248
1201	155
1202	174
1203	250
1204	1
1205	235
1206	2
1207	160
1208	3
1209	4
1210	240
1211	5
1212	6
1213	230
1214	246
1215	7
1216	228
1217	237
1218	231
1219	8
1220	225
1221	239
1222	242
1223	9
1224	10
1225	11
1226	236
1227	12
1228	229
1229	227
1230	13
1231	244
1232	14
1233	243
1234	15
1235	16
1236	17
1237	238
1238	18
1239	19
1240	3
1241	239
1242	155
1243	225
1244	229
1245	245
1246	1
1247	2
1248	8
1249	9
1250	236
1251	249
1252	167
1253	238
1254	1
1255	172
1256	155
1257	174
1258	2
1259	3
1260	4
1261	243
1262	5
1263	233
1264	6
1265	160
1266	7
1267	229
1268	22
1269	23
1270	167
1271	173
1272	238
1273	227
1274	235
1275	242
1276	155
1277	226
1278	1
1279	2
1280	245
1281	3
1282	244
1283	172
1284	4
1285	5
1286	230
1287	237
1288	246
1289	6
1290	174
1291	240
1292	7
1293	8
1294	243
1295	9
1296	10
1297	228
1298	11
1299	12
1300	249
1301	13
1302	239
1303	14
1304	225
1305	15
1306	16
1307	233
1308	236
1309	17
1310	160
1311	229
1312	18
1313	19
1314	20
1315	21
1316	12
1317	13
1318	167
1319	187
1320	155
1321	1
1322	249
1323	174
1324	226
1325	2
1326	237
1327	243
1328	3
1329	245
1330	239
1331	240
1332	4
1333	5
1334	233
1335	6
1336	7
1337	8
1338	9
1339	160
1340	225
1341	229
1342	10
1343	11
1344	25
1345	26
1346	173
1347	187
1348	226
1349	234
1350	237
1351	242
1352	250
1353	230
1354	236
1355	1
1356	2
1357	3
1358	155
1359	245
1360	4
1361	167
1362	246
1363	249
1364	5
1365	6
1366	235
1367	239
1368	7
1369	8
1370	9
1371	10
1372	172
1373	11
1374	12
1375	227
1376	174
1377	13
1378	238
1379	233
1380	14
1381	225
1382	15
1383	243
1384	16
1385	17
1386	244
1387	18
1388	231
1389	229
1390	19
1391	20
1392	228
1393	21
1394	22
1395	23
1396	160
1397	24
1398	26
1399	27
1400	194
1401	155
1402	173
1403	172
1404	248
1405	1
1406	174
1407	2
1408	3
1409	229
1410	231
1411	232
1412	249
1413	233
1414	235
1415	4
1416	227
1417	225
1418	5
1419	246
1420	6
1421	228
1422	7
1423	226
1424	240
1425	8
1426	9
1427	243
1428	244
1429	247
1430	239
1431	10
1432	11
1433	12
1434	13
1435	236
1436	14
1437	15
1438	16
1439	245
1440	237
1441	17
1442	230
1443	160
1444	18
1445	242
1446	19
1447	20
1448	21
1449	238
1450	22
1451	23
1452	24
1453	25
1454	14
1455	15
1456	173
1457	237
1458	249
1459	155
1460	174
1461	1
1462	243
1463	2
1464	3
1465	245
1466	244
1467	240
1468	4
1469	239
1470	5
1471	233
1472	6
1473	232
1474	160
1475	225
1476	236
1477	7
1478	242
1479	8
1480	229
1481	9
1482	10
1483	11
1484	12
1485	13
1486	155
1487	245
1488	25
1489	26
1490	169
1491	187
1492	246
1493	230
1494	1
1495	155
1496	173
1497	226
1498	240
1499	2
1500	167
1501	3
1502	4
1503	5
1504	245
1505	227
1506	172
1507	231
1508	242
1509	6
1510	235
1511	7
1512	236
1513	237
1514	238
1515	249
1516	8
1517	174
1518	9
1519	10
1520	228
1521	11
1522	12
1523	244
1524	13
1525	243
1526	14
1527	15
1528	16
1529	225
1530	239
1531	17
1532	233
1533	18
1534	19
1535	229
1536	20
1537	160
1538	21
1539	22
1540	23
1541	24
1542	160
1543	22
1544	162
1545	167
1546	226
1547	235
1548	237
1549	238
1550	155
1551	247
1552	1
1553	2
1554	3
1555	187
1556	249
1557	240
1558	4
1559	5
1560	236
1561	6
1562	7
1563	8
1564	245
1565	225
1566	9
1567	172
1568	227
1569	10
1570	232
1571	11
1572	233
1573	12
1574	239
1575	243
1576	174
1577	13
1578	14
1579	229
1580	15
1581	16
1582	17
1583	244
1584	18
1585	19
1586	20
1587	21
1588	20
1589	21
1590	187
1591	226
1592	173
1593	237
1594	1
1595	155
1596	167
1597	227
1598	172
1599	236
1600	238
1601	2
1602	247
1603	3
1604	4
1605	249
1606	5
1607	6
1608	7
1609	8
1610	244
1611	174
1612	245
1613	9
1614	10
1615	242
1616	225
1617	243
1618	11
1619	12
1620	13
1621	233
1622	14
1623	15
1624	239
1625	229
1626	16
1627	160
1628	232
1629	17
1630	18
1631	19
1632	17
1633	18
1634	239
1635	246
1636	155
1637	235
1638	249
1639	1
1640	160
1641	226
1642	2
1643	225
1644	3
1645	237
1646	4
1647	227
1648	233
1649	5
1650	228
1651	229

1652	231	1671	229	1690	243	1709	172	1728	233	1747	9	1766	233
1653	6	1672	243	1691	238	1710	173	1729	247	1748	10	1767	1
1654	236	1673	249	1692	242	1711	244	1730	167	1749	174	1768	2
1655	240	1674	155	1693	3	1712	233	1731	1	1750	11	1769	3
1656	7	1675	1	1694	229	1713	1	1732	2	1751	12	1770	4
1657	8	1676	239	1695	4	1714	2	1733	187	1752	13	1771	5
1658	9	1677	2	1696	232	1715	225	1734	3	1753	14	1772	155
1659	10	1678	3	1697	160	1716	229	1735	4	1754	15	1773	155
1660	11	1679	225	1698	225	1717	3	1736	236	1755	16	1774	155
1661	243	1680	4	1699	5	1718	155	1737	5	1756	6	1775	155
1662	12	1681	233	1700	239	1719	4	1738	155	1757	7	1776	155
1663	244	1682	10	1701	6	1720	17	1739	238	1758	160	1777	155
1664	238	1683	11	1702	7	1721	160	1740	6	1759	174	1778	155
1665	13	1684	174	1703	8	1722	191	1741	239	1760	225	1779	155
1666	242	1685	155	1704	233	1723	225	1742	7	1761	229	1780	155
1667	14	1686	236	1705	9	1724	226	1743	172	1762	236	1781	155
1668	15	1687	237	1706	5	1725	230	1744	229	1763	250		
1669	16	1688	1	1707	6	1726	237	1745	243	1764	155		
1670	5	1689	2	1708	160	1727	228	1746	8	1765	239		

Annex D

An Overview of PSIP for Terrestrial Broadcast with Application Examples

(Informative)

1. INTRODUCTION

The Program and System Information Protocol (PSIP) is a small collection of tables designed to operate within every Transport Stream for terrestrial broadcast of digital TV. Its purpose is to describe the information at the system and event levels for all virtual channels carried in a particular Transport Stream. Additionally, information for analog channels as well as digital channels from other Transport Streams may be incorporated. The relational hierarchy for the component tables is explained through typical application examples in this document.

PSIP is the result of combining and compacting two existing optional ATSC protocols: A/55 and A/56. Although these protocols were individually efficient and accomplished their purpose, their mutual implementation was difficult due to their structural differences and their overlapping definitions. PSIP solves this problem. The tables defined in PSIP use packet identifiers (PIDs) that are different from those specified by the optional A/55 and A/56 standards. This provision has been included to enable the operation of existing equipment designed or manufactured to support A/55 and/or A/56.

2. OVERVIEW

Under the adopted ATSC standard for digital TV, the typical 6 MHz channel used for analog TV broadcast supports about 19 Mbps of throughput for terrestrial broadcast. Since audiovisual signals with standard resolution can be compressed using MPEG-2 to sustainable rates of around 6 Mbps, then around 3 or 4 digital TV channels can be safely supported in a single physical channel without congestion. Moreover, enough bandwidth remains within the same Transport Stream to provide several additional low-bandwidth non-conventional services such as: weather reports, stock indices, headline news, software download (for games or enhanced applications), image-driven classified ads, home shopping, pay-per-view information, and others.

It is therefore practical to anticipate that in the future, the list of services (virtual channels) carried in a physical transmission channel (6 MHz of bandwidth for the U.S.) may easily reach ten or more. What is even more important is that the number and type of services may also change continuously, thus becoming a more dynamic medium than what we have today.

An important feature of terrestrial broadcasting is that sources follow a distributed information model rather than a centralized one. Unlike cable or satellite, service providers are geographically distributed and have no interaction with respect to data unification or even synchronization. It is therefore necessary to develop a protocol for describing system information and event descriptions which is followed by every organization in charge of a physical transmission channel. System information allows navigation and access to each of the channels within the Transport Stream, whereas event descriptions give the user content information for browsing and selection.

In this document we describe the development of a transport-based implementation of the PSIP protocol using examples. Our hope is to introduce the reader to the most important concepts and components that constitute the protocol.

3. ELEMENTS OF PSIP

PSIP is a collection of hierarchically-associated tables each of which describes particular elements of typical digital TV services. Figures D1 and D2 show the different components and the notation used to describe them. The packets of the base tables are all labeled with the base PID (base_PID) which has been chosen as 0x1FFB. The base tables are: the System Time Table (STT), the Rating Region Table (RRT), the Master Guide Table (MGT), and the Virtual Channel Table (VCT).

A second set of tables are the Event Information Tables (EIT) whose packet identifiers (PIDs) are defined in the MGT. A third set of tables are the Extended Text Tables (ETT), and similarly, their packet identifiers (PIDs) are defined in the MGT.

The System Time Table (STT) is a small data structure that fits in one Transport Stream packet and serves as a reference for time of day. Receivers can use this table as a reference for timing start times of advertised events.

It should be noted that, except for the MGT, PSIP table sections may start in any byte position within an MPEG-2 transport stream packet. The Master Guide Table is special in that the first byte always is aligned with the first byte of the packet payload. The A/65 standard states this restriction as the pointer_field of the Transport Stream packet carrying the table_id field of the MGT section shall have the value 0x00 (section starts immediately after the pointer_field).

In general, table sections may span packet boundaries. Also, if the table sections are small enough, more than one PSIP table section may be present within a single transport stream packet. The MPEG-2 pointer_field mechanism is used to indicate the first byte of a table section within a packet payload. The starting byte of subsequent table sections that might be in the same payload is determined by processing successive section_length fields. The location of the section_length field is guaranteed to be consistent for any type of PSIP table section, as the format conforms to MPEG-2 defined Program Specific Information (PSI) tables.

If a packet payload does not include the start of a table section, the payload_unit_start_indicator bit in the packet header is set to '0' and the pointer_field is not present.

Transmission syntax for the United States' voluntary program rating system is included in this standard. The Rating Region Table (RRT) has been designed to transmit the rating standard in use for each country using the standard. Provisions were made for different rating systems for different countries and multi-country regions as well.

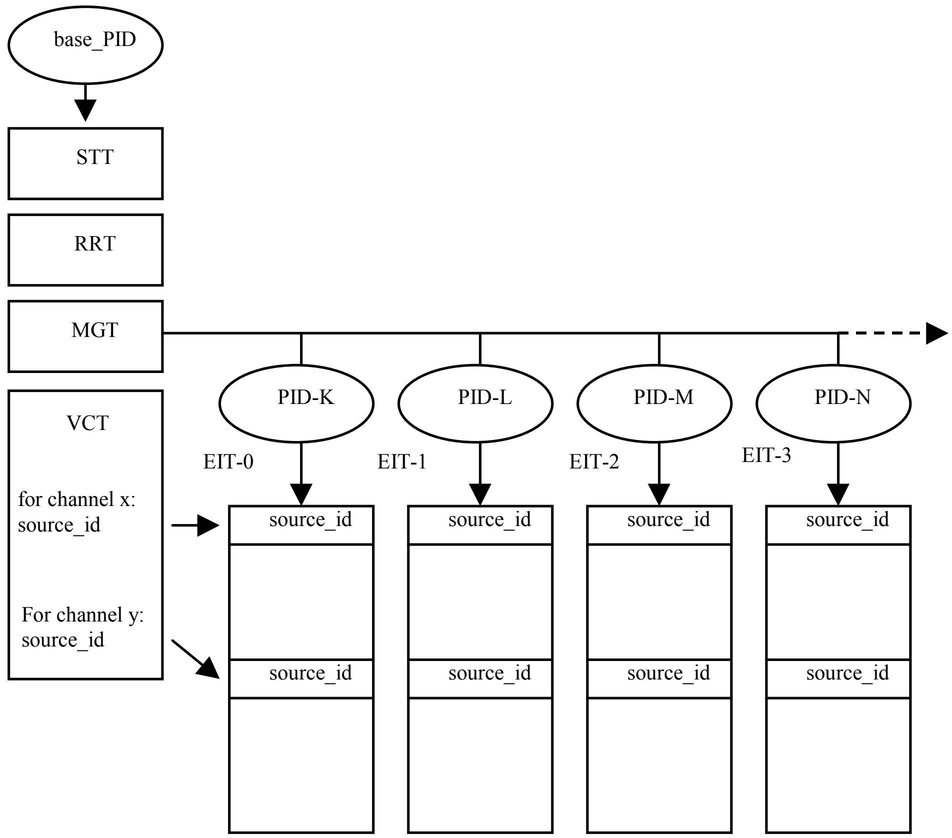


Figure D1 Main Structure for the PSIP tables.

The Master Guide Table (MGT) provides general information about all of the other tables that comprise the PSIP standard. It defines table sizes necessary for memory allocation during decoding; it defines version numbers to identify those tables that need to be updated; it has a constrained header location to facilitate receiver acquisition; and it gives the packet identifiers (PIDs) that label the tables.

The Virtual Channel Table (VCT), also referred to as the Terrestrial VCT (TVCT), contains a list of all the channels that are or will be on-line plus their attributes. Among the attributes we have the channel name, navigation identifiers, stream components and types, etc.

As part of PSIP there are several Event Information Tables, each of which describes the events or TV programs associated with each of the virtual channels listed in the VCT. Each EIT is valid for a time interval of 3 hours. Since the total number of EITs is 128, up to 16 days of programming may be advertised in advance. EIT-0 always denotes the current 3 hours of programming, EIT-1 the next 3 hours, and so on. As a minimum, the first four EITs must always be present in every Transport Stream.

Start times for EITs are constrained to be one of the following UTC times: 0:00 (midnight), 3:00, 6:00, 9:00, 12:00 (noon), 15:00, 18:00, and 21:00. Imposing constraints on the start times as well as the interval duration is necessary for the purpose of re-multiplexing. During re-multiplexing, EIT tables coming from several distinct Transport Streams may end up grouped together or *vice versa*. If no constraints were imposed, re-multiplexing equipment would have to parse EITs by content in real time, which is a difficult task.

For example, consider a broadcast corporation operating in the Eastern time zone of the U.S. This corporation decides to carry 6 EITs (18 hours of TV program information). If at present, the Eastern time is 15:30 EDT (19:30 UTC), then the coverage times for the EIT tables are:

Table D1 An Example of EIT Coverage Times

EIT number	Version Num.	Assigned PID	Coverage (UTC)	Coverage (EDT)
0	6	123	18:00 - 21:00	14:00 - 17:00
1	4	190	21:00 - 24:00	17:00 - 20:00
2	2	237	0:00 - 3:00	20:00 - 23:00
3	7	177	3:00 - 6:00	23:00 - 2:00 (nd)
4	8	295	6:00 - 9:00	2:00 (nd) - 5:00 (nd)
5	15	221	9:00 - 12:00	5:00 (nd) - 8:00 (nd)

The abbreviation “nd” denotes next day. Before 17:00 EDT, the MGT will list the currently valid PIDs as: 123, 190, 237, 177, 295, and 221. At 17:00 EDT, table EIT-0 will become obsolete while the other ones will remain valid. At that time, the PID list can be changed to 190, 237, 177, 295, 221, maintaining the version number list as 4, 2, 7, 8, 15. Therefore, by simply shifting the listed PID values in the MGT, table EIT-1 can become EIT-0, table EIT-2 can become EIT-1, and so on.

However, it is also possible to regenerate one or several EITs at any time for correcting and/or updating the content (e.g., in cases where “to be assigned” events become known). Regeneration of EITs is flagged by updating version fields in the MGT. For example, if table EIT-2 needs to be updated at 16:17 EDT, then the new table must be transmitted with a version number equal to 3. Whenever the decoder monitoring the MGT detects a change in the version number of a table, it assumes that the table has changed and needs to be reloaded.

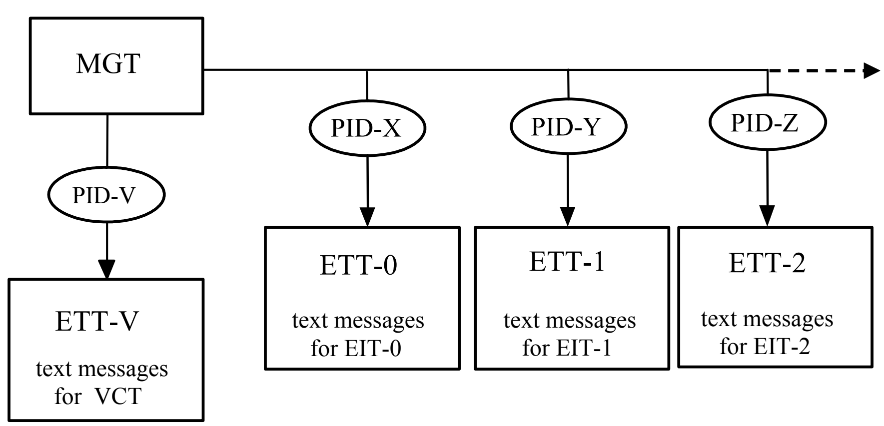


Figure D2 Extended Text Tables in the PSIP hierarchy.

As illustrated in Figure D2, there can be several Extended Text Tables (ETTs), each of them having its PID defined in the MGT. Each Event Information Table (EIT) can have one ETT. Similarly, the Virtual Channel Table can have one ETT. As its name indicates, the purpose of an Extended Text Table (ETT) is to carry text messages. For example, for channels in the VCT, the messages can describe channel information, cost, coming attractions, etc. Similarly, for an event such as a movie listed in the EIT, the typical message is a short paragraph that describes the movie itself. Extended Text Tables are optional.

In this final section paragraph we review once more the requirement list. The minimum amount of information required in an ATSC terrestrial digital Transport Stream is the VCT, the MGT, the RRT, the STT, and the first four EITs. All of the other elements are optional.

4. APPLICATION EXAMPLE

For the purpose of this example, we assume that a broadcast group, here denominated NBZ, manages the frequency bands for RF channels 12 and 39. The first one is its analog channel whereas the second one will be used for digital broadcast. According to the premises established in this document, NBZ must carry the PSIP tables in the digital Transport Stream of RF channel 39. The tables must describe TV programs and other services provided on RF channel 39 but can also describe information for the analog RF channel 12.

Assume that NBZ operates in the Eastern time zone of the U.S., and that the current time is 15:30 EDT (19:30 UTC). NBZ decides to operate in minimal configuration, therefore only the first four EITs need to be transmitted. As explained previously, EIT-0 must carry event information for the time window between 14:00 and 17:00 EDT, whereas EIT-1 to EIT-3 will cover the subsequent 9 hours. For the first 6 hours, the following scenario applies:

Table D2 The First 3-Hour Segment to be Described in VCT and EIT-0

		14:00-14:30	**14:30 -15:00**	**15:00 - 15:30**	**15:30 - 16:00**	**16:00 - 16:30**	**16:30-17:00**
PTC 12 (12-0)	NBZ	City Life	City Life	Travel Show	Travel Show	News	News
PTC 39 (12-1)	NBZ	City Life	City Life	Travel Show	Travel Show	News	News
PTC 39 (12-2)	NBZ	Soccer	Golf Report	Golf Report	Car Racing	Car Racing	Car Racing
PTC 39 (12-3)	NBZ	Secret Agent	Secret Agent	Lost Worlds	Lost Worlds	Lost Worlds	Lost Worlds
PTC 39 (12-4)	NBZ	Headlines	Headlines	Headlines	Headlines	Headlines	Headlines

Table D3 The Second 3-Hour Segment to be Described in VCT and EIT-1

		17:00-17:30	17:30-18:00	18:00 - 18:30	18:30 - 19:00	19:00-19:30	19:30 - 20:00
PTC 12 (12-0)	NBZ	Music Today	NY Comedy	World View	World View	News	News
PTC 39 (12-1)	NBZ	Music Today	NY Comedy	World View	World View	News	News
PTC 39 (12-2)	NBZ	Car Racing	Car Racing	Sports News	Tennis Playoffs	Tennis Playoffs	Tennis Playoffs
PTC 39 (12-3)	NBZ	Preview	The Bandit	The Bandit	The Bandit	The Bandit	Preview
PTC 39 (12-4)	NBZ	Headlines	Headlines	Headlines	Headlines	Headlines	Headlines

Similar tables can be built for the next 6 hours (for EIT-2 and EIT-3). According to this scenario, NBZ broadcasts four regular digital channels (also called virtual channels and denoted by their major and minor channel numbers), one with the same program as the analog transmission, another for sports, and a third one for movies. The fourth one supports a service displaying headlines with text and images.

4.1 The Master Guide Table (MGT)

The purpose of the MGT is to describe everything about the other tables, listing features such as version numbers, table sizes, and packet identifiers (PIDs). Figure D3 shows a typical Master Guide Table indicating, in this case, the existence in the Transport Stream of a Virtual Channel Table, the Rating Region Table, four EITs, one Extended Text Table for channels, and two Extended Text Tables for events.

The first entry of the MGT describes the version number and size of the Virtual Channel Table. The second entry corresponds to an instance of the Rating Region Table. If some region's policy makers decided to use more than one instance of an RRT, the MGT would list each PID, version number, and size. Notice that the base PID (0x1FFB) must be used for the VCT and the RRT instances as specified in PSIP.

The next entries in the MGT correspond to the first four EITs that must be supplied in the Transport Stream. The user is free to choose their PIDs as long as they are unique in the MGT list of PIDs. After the EITs, the MGT indicates the existence of an Extended Text Table for channels carried using PID 0x1AA0. Similarly, the last two entries in the MGT signal the existence of two Extended Text Tables, one for EIT-0 and the other for EIT-1.

MGT

table_type	PID	version_num.	table size
VCT	0x1FFB (base_PID)	4	485 bytes
RRT	0x1FFB (base_PID)	1	560 bytes
EIT-0	0x1FD0	6	2730 bytes
EIT-1	0x1FD1	4	1342 bytes
EIT-2	0x1DD1	2	1224 bytes
EIT-3	0x1DB3	7	1382 bytes
ETT for VCT	0x1AA0	21	4232 bytes
ETT-0	0x1BA0	10	32420 bytes
ETT-1	0x1BA1	2	42734 bytes

Figure D3 Content of the Master Guide Table.

Descriptors can be added for each entry as well as for the entire MGT. By using descriptors, future improvements can be incorporated without modifying the basic structure of the MGT. The MGT is like a flag table that continuously informs the decoder about the status of all the other tables (except the STT which has an independent function). The MGT is continuously monitored at the receiver to prepare and anticipate changes in the channel/event structure. When tables are changed at the broadcast side, their version numbers are incremented and the new numbers are listed in the MGT. Based on the version updates and on the memory requirements, the decoder can reload the newly defined tables for proper operation.

4.2 The Virtual Channel Table (VCT)

Figure D4 shows the structure of the VCT which essentially contains the list of channels available in the Transport Stream. For convenience, it is possible to include analog channels and even other digital channels found in different Transport Streams.

The field number_of_channels_in_section indicates the number of channels described in one section of the VCT. In normal applications, as in the example being considered here, all channel information will fit into one section. However, there may be rare times when most of the physical channel is used to convey dozens of low-bandwidth services such as audio-only and data channels in addition to one video program. In those cases, the channel information may be larger than the VCT section limit of 1 Kbyte and therefore VCT segmentation will be required.

For example, assuming that a physical channel conveys 20 low-bandwidth services in addition to a TV program, and assuming that their VCT information exceeds 1 Kbyte, then two or more sections may be defined. The first section may describe 12 virtual channels and the second 9 if such a partition leads to VCT sections with less than 1 Kbyte.

A new VCT containing updated information can be transmitted at any time with the version_number increased by one. However, since a VCT describes only those channels from a particular Transport Stream, virtual channels added to the VCT at arbitrary times will not be detected by the receiver until it is tuned to that particular Transport Stream. For this reason, it is highly recommended that channel addition be made in advance to give the receivers the opportunity to scan the frequencies and detect the channel presence.

The fields major_channel_number and minor_channel_number are used for identification. The first one, the major channel number, is used to group all channels that are to be identified as belonging to a particular broadcast corporation (or particular identifying number such as 12 in this case). The minor channel number specifies a particular channel within the group.

The field short_name is a seven-character name for the channel and may allow text-based access and navigation. The fields transport_stream_id and program_number are included to link the VCT with the PAT and sections of the PMT. A sequence of flags follows these fields. The flags indicate: 1) if the channel is hidden (e.g., for NVOD applications), 2) if the channel has a long text message in the VCT-ETT, and 3) if the channel is visible in general or has some conditional access constraints.

After the flags, a description of the type of service offered is included, followed by the source_id. The source_id is simply an internal index for representing the particular logical channel. Event Information Tables and Extended Text Tables use this number to provide a list of associated events or text messages respectively.

VCT

current_next_indicator = 1
number_channels_in_section = 5

major num.	minor num.	short name	channel TSID	progr. num.	flags	service type	source id	descriptors
12	0	NBZ	0x0AA0	0xFFFF	--	analog	20	ch_name
12	1	NBZD	0x0AA1	0x00F1	--	digital	21	ch_name serv_locat.
12	5	NBZ-S	0x0AA1	0x00F2	--	digital	2	ch_name serv_locat.
12	12	NBZ-M	0x0AA1	0x00F3	--	digital	23	ch_name serv_locat.
12	31	NBZ-H	0x0AA1	0x00F8	--	digital	24	ch_name serv_locat.

Figure D4 Content of the Virtual Channel Table.

Two descriptors are associated with the logical channels in the example. The first one is extended_channel_name and, as its name indicates, it gives the full name of the channel. An example for channel NBZ-S could be: "NBZ Sports and Fitness". The other one, the service_location descriptor, is used to list the available bit streams and their PIDs necessary to decode packets at the receiver. Assuming that NBZ-M offers bilingual transmission, then the following attributes are tabulated within its service_location descriptor:

PID_audio_1	AC-3 audio	English
PID_audio_2	AC-3 audio	Spanish
PID_video	MPEG-2 video	No lang.

Two VCTs may exist simultaneously in a Transport Stream: the current and the next VCT. The current VCT is recognized by having the flag current_next_indicator set to 1, while the next one has this flag set to 0. The "next" VCT should not be transmitted when it fits into a single table section, since delivery of the new "current" table will take effect as soon as the one (and only) section arrives. Multi-sectioned next VCTs may be sent, but should not be delivered until immediately before the point at which they are to become current. This recommendation arises because no mechanism is available to update the "next" tables without affecting the current table definition.

For multi-sectioned VCTs, delivery of the “next” table is helpful. Consider a Transport Stream containing the following table sections:

- Current VCT, version_number=5, section 1 of 2
- Current VCT, version_number=5, section 2 of 2
- Next VCT, version_number=6, section 1 of 2
- Next VCT, version_number=6, section 2 of 2

At the point in time when the “next” tables are to become current, the following table section may be placed into the Transport Stream:

- Current VCT, version_number=6, section 1 of 2

At the moment this table section is processed, both sections of the version 6 VCT are understood to be the new “current” VCT, even before section 2 of 2 of VCT version 6 labeled “current” is received. As long as the “next” table sections have been cached, they can be taken as “current” as soon as the version number is seen to increment.

When the VCT refers to an analog service type, the channel_TSID cannot refer to the identifier of a "Transport Stream" in the MPEG-2 sense. Analog NTSC broadcast signals can, however, carry a 16-bit unique identifier called a "Transmission Signal Identifier."[18] For the example VCT in Figure D4, the Transmission Signal Identifier for channel 12.0 is 0x0AA0. Subsequently, receivers are expected to associate the NTSC broadcast identified by the Transmission Signal ID with the frequency tuned to acquire it. Given this association, a receiver can use the Transmission Signal ID to determine how to tune to the NTSC channel it identifies.

It is recommended that the broadcaster insert into the VCT any major-minor channel that would be used to carry any program announced in the EIT. This means if no current program was using 7-7, and if a program 16 days from now was going to use 7-7, that 7-7 would be in the VCT. This would enable receivers to include the channel number in a program guide presented to the consumer. If a program is announced in the EIT and the source ID for that program is not found in the VCT, the receiver cannot determine which "channel" to display for that program.

Any channels in the VCT which are not currently active shall have the hidden attribute set to 1 and the hide_guide attribute set to 0.

The following table shows DTV behavior for the various combinations of the hidden and hide_guide attributes. In the table the “x” entry indicates “don’t care.” A check in the “surf” column indicates the channel is available by channel surfing and via direct channel number entry. A check in the “guide” column indicates that the channel may appear in the program guide listing.

Table D4 Receiver Behavior with Hidden and Hide Guide Attributes

hidden	hide_guide	Receiver Behavior		Description
		Surf	Guide	
0	x	✓	✓	Normal channel
1	1			Special access only
1	0		✓	Inactive channel

[18] The 16-bit “Transmission Signal ID” for the NTSC VBI is specified in EIA/CEA-608-B [13] Section 9.5.2.4.

4.3 The Event Information Tables (EITs)

The purpose of an EIT is to list all events for those channels that appear in the VCT for a given time window. As mentioned before, EIT-0 describes the events for the first 3 hours, EIT-1 for the next 3 hours, and so on. EIT-i and EIT-j have different PIDs as defined in the MGT. In PSIP, tables can have a multitude of instances. The different instances of a table share the same table_id value and PID but use different table_id_extension values.

In PSIP, an instance of EIT-k contains the list of events for a single virtual channel with a unique source_id. For this reason, the table_id_extension has been renamed as source_id in the EIT syntax. Figure D5 shows, for example, the NBZ-S instance for EIT-0. Following similar procedures, the NBZD, NBZ-M, and NBZ-H instances of EIT-0 can be constructed. The process can be extended and repeated to obtain all of the instances for the other tables in the time sequence: EIT-1, EIT-2, etc.

The three events programmed for the 3-hour period for NBZ-S are listed in Figure D5. The field event_id is a number used to identify each event. If an event time period extends over more than one EIT, the same event_id has to be used. The event_id is used to link events with their messages defined in the ETT, and therefore it has to be unique only within a virtual channel and a 3-hour interval defined by EITs. The event_id is followed by the start_time and then the length_in_seconds. Notice that events can have start times before the activation time (14:00 EDT in this example) of the table.. The ETM_location specifies the existence and the location of an Extended Text Message (ETM) for this event. ETMs are simply long textual descriptions. The collection of ETMs constitutes an Extended Text Table (ETT).

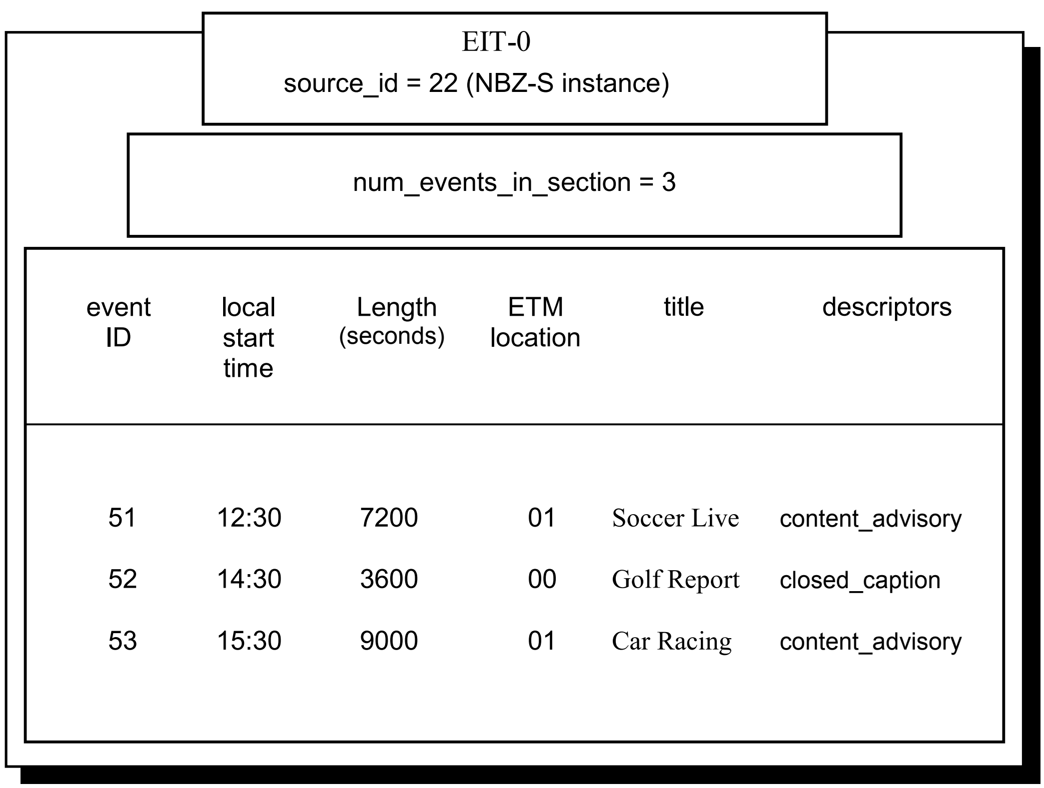


Figure D5 Content of EIT-0 for NBZ-S

An example of an ETM for the Car Racing event may be:

> “Live coverage from Indianapolis. This car race has become the largest single-day sporting event in the world. Two hundred laps of full action and speed.”

Several descriptors can be associated with each event. One is the content advisory descriptor which assigns a rating value according to one or more systems. Recall that the actual rating system definitions are tabulated within the RRT. Another is a closed caption descriptor which signals the existence of closed captioning and lists the necessary parameters for decoding.

4.4 The Rating Region Table (RRT)

The Rating Region Table is a fixed data structure in the sense that its content remains mostly unchanged. It defines the rating standard that is applicable for each region and/or country. The concept of table instance introduced in the previous Section is also used for the RRT. Several instances of the RRT can be constructed and carried in the Transport Stream simultaneously. Each instance is identified by a different table_id_extension value (which becomes the rating_region in the RRT syntax) and corresponds to one and only one particular region. Each instance has a different version number which is also carried in the MGT. This feature allows updating each instance separately.

Figure D6 shows an example of one instance of an RRT, for a region called “Tumbolia,” assigned by the ATSC to rating_region 20. Each event listed in any of the EITs may carry a content advisory descriptor. This descriptor is an index or pointer to one or more instances of the RRT.

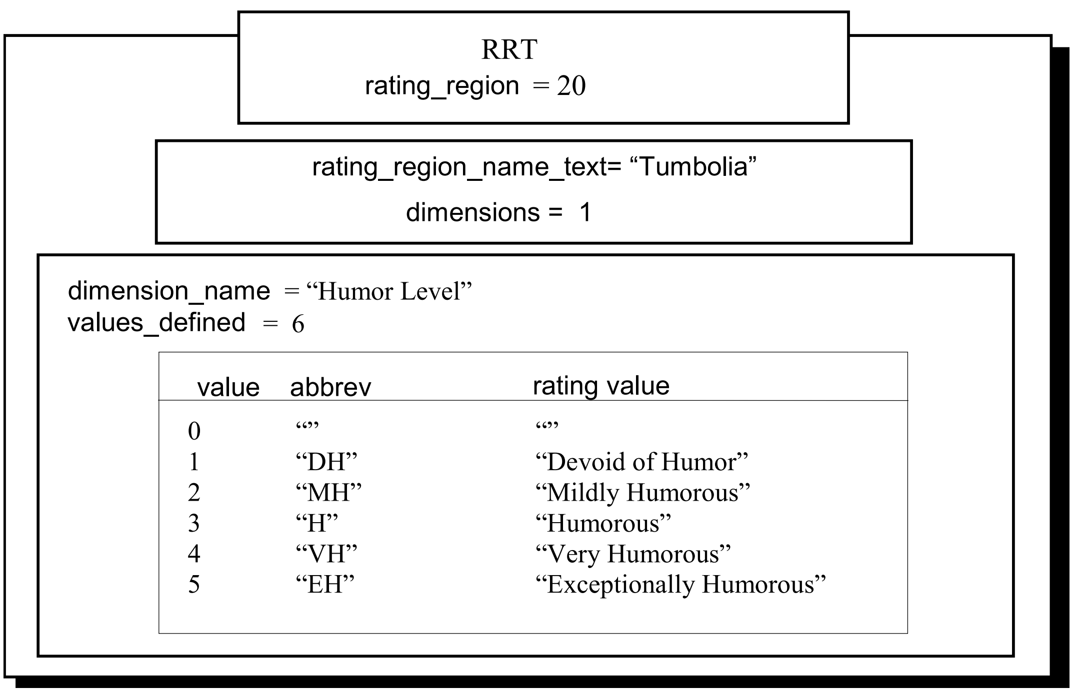


Figure D6 An Instance of a Rating Region Table (RRT).

5. PACKETIZATION AND TRANSPORT

In the previous sections, we have described how to construct the MGT, VCT, RRT, and EITs based on the typical scenario described in Tables D1 and D2. The number of virtual channels described in the VCT is 5 and therefore, each EIT will have 5 instances.

For the example, the size of the MGT is less than a hundred bytes and the VCT ranges between 300 to around 1500 bytes depending on the length of the text strings. Similarly, each EIT instance can have from 1 to about 3 Kbytes depending again on the text length.

Typically, the MGT, STT, VCT, and each instance of the RRT and EIT will have one or at most a few sections. For each table, the sections are appended one after the other, and then segmented into 184-byte packets. After adding the 4-byte MPEG-2 TS header, the packets are multiplexed with the others carrying audio, video, data, and any other components of the service. Figure D7 illustrates this process.

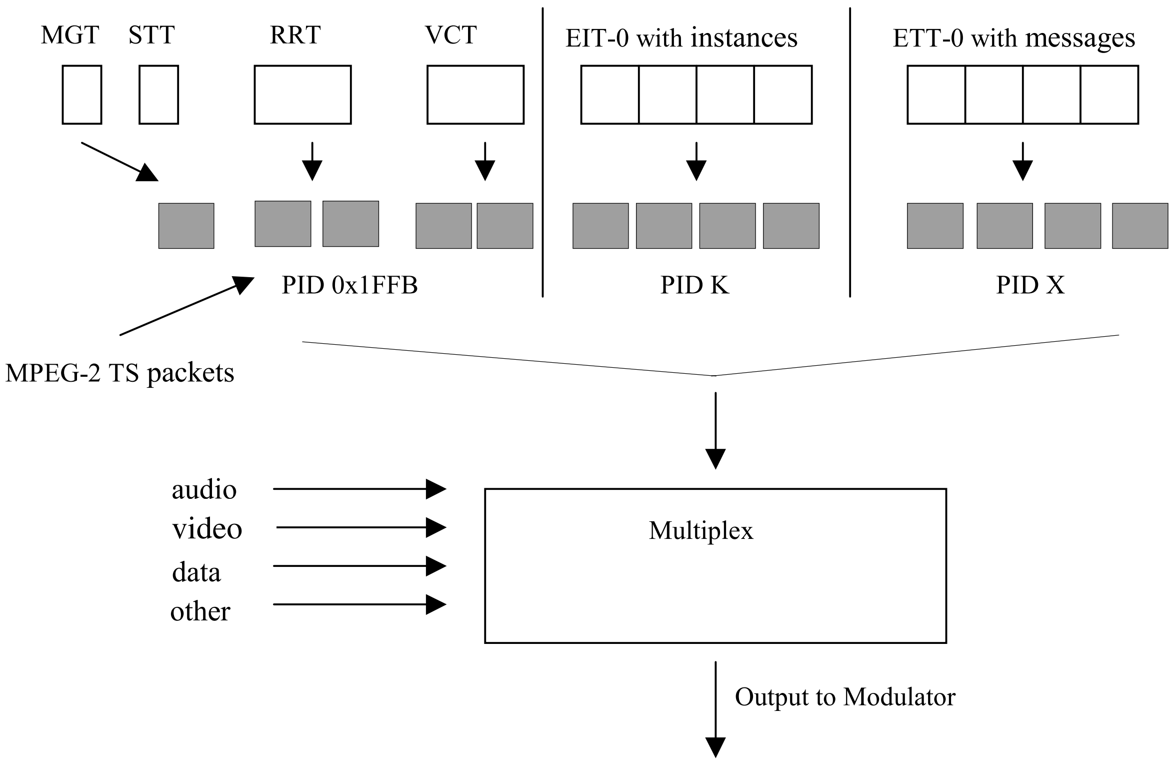


Figure D7 Packetization and Transport of the PSIP tables.

6. TUNING OPERATIONS AND TABLE ACCESS

As described by the PSIP protocol, each Transport Stream will carry a set of tables describing system information and event description. For channel tuning, the first step is to collect the VCT from the Transport Stream which contains the current list of services available. Figure D8 shows this process.

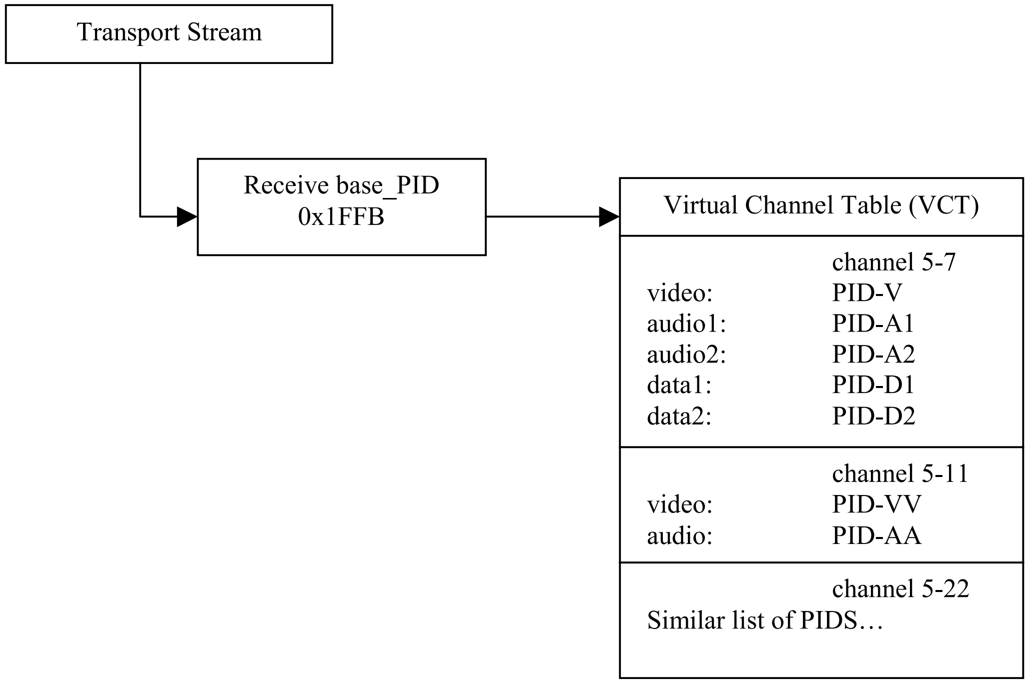


Figure D8 Extraction of the VCT from the Transport Stream.

Once the VCT has been collected, a user can tune to any virtual channel present in the Transport Stream by referring to the major and minor channel numbers. Assuming that in this case, the user selects channel 5 - 11, then the process for decoding the audio and video components is shown in Figure D9.

For terrestrial broadcast, the existence of a service location descriptor in the TVCT is mandatory. The PID values needed for acquisition of audio and video elementary streams may be found in either a service_location_descriptor() within a TVCT, or in a TS_program_map_section(). The service_location_descriptor() has been included in PSIP to minimize the time required for changing and tuning to channels. However, PAT and PMT information is required to be present in the Transport Stream to provide MPEG-2 compliance. Access to data or other supplemental services may require access to the PAT or TS_program_map_section(). Cable systems may or may not carry the Service Location Descriptor, and the information contained therein will be found in the TS_program_map_section().

The PMT should always be processed and monitored for changes because, in some instances, the structure of a service may exceed the capability of the service_location_descriptor() to describe it. One example is a service that includes multiple audio tracks for the same language (see Section 6.9.8). In this case, the TS_program_map_section()carries the textual name for each of the tracks to help in user selection.

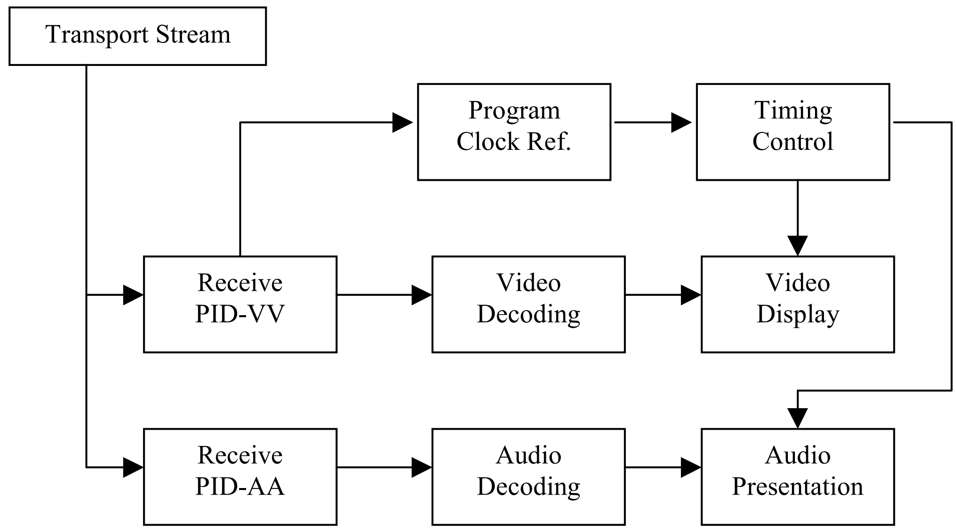


Figure D9 Acquisition of audiovisual components.

7. GPS TIME

The System Time Table provides time of day information to receivers. In PSIP, time of day is represented as the number of seconds that have elapsed since the beginning of "GPS time," 00:00:00 UTC January 6th, 1980. GPS time is referenced to the Master Clock at the US Naval Observatory and steered to Coordinated Universal Time (UTC). UTC is the time source we use to set our clocks.

UTC is occasionally adjusted by one-second increments to ensure that the difference between a uniform time scale defined by atomic clocks does not differ from the Earth's rotational time by more than 0.9 seconds. The timing of occurrence of these "leap seconds" is determined by careful observations of the Earth's rotation; each is announced months in advance. On the days it is scheduled to occur, the leap second is inserted just following 12:59:59 PM UTC.

UTC can be directly computed from the count of GPS seconds since January 6, 1980, by subtracting from it the count of leap seconds that have occurred since the beginning of GPS time. In the months just following January 1, 1999, this offset was 13 seconds.

In the A/65 protocol, times of future events (such as event start times in the EIT) are specified the same as time of day, as the count of seconds since January 6, 1980. Converting an event start time to UTC and local time involves the same calculation as the conversion of system time to local time. In both cases, the leap seconds count is subtracted from the count of GPS seconds to derive UTC.

GPS time is used to represent future times because it allows the receiver to compute the time interval to the future event without regard for the possible leap second that may occur in the meantime. Also, if UTC were to be used instead, it wouldn't be possible to specify an event time that occurred right at the point in time where a leap second was added. UTC is discontinuous at those points.

Around the time a leap second event occurs, program start times represented in local time (UTC adjusted by local time zone and [as needed] daylight savings time) may appear to be off by plus or minus one second. PSIP generating equipment may use one of two methods to handle leap seconds.

In method A, PSIP generating equipment does not anticipate the future occurrence of a leap second. In this case, prior to the leap second, program start times will appear correct. An event starting at exactly 10 AM will be computed as starting at 10:00:00. But just following the leap second, that same event time will be computed as 9:59:59. The PSIP generating equipment should re-compute the start times in all the EITs and introduce the leap second correction. Once that happens, and receivers have updated their EIT data, the computed time will again show as 10:00:00. In this way the disruption can be limited to a matter of seconds.

In method B, PSIP generating equipment does anticipate the occurrence of a leap second, and adjusts program start times for events happening after the new leap second is added. If the leap second event is to occur at midnight tonight, an event starting at 10 AM tomorrow will be computed by receiving equipment as starting at 10:00:01.

For certain types of events, the precision of method B is necessary. By specifying events using a time system that involves no discontinuities, difficulties involving leap seconds are avoided. Events such as program start times do not require that level of precision. Therefore, method A works well.

Consider the following example. Times are given relative to UTC, and would be corrected to local time zone and daylight savings time as necessary.

- Time of day (UTC): 1:00 p.m., December 30, 1998
- Event start time (UTC): 2:00 p.m., January 2, 1999
- A leap second event will occur just after 12:59:59 p.m. on December 31 , 1998
- Leap seconds count on December 30 is 12

The data in the System Time message is:

- GPS seconds = 599,058,012 = 0x23B4E65C
- GPS to UTC offset = 12

Using method A (upcoming leap second event is not accounted for):

- Event start time in EIT: 599,320,812 = 0x23B8E8EC
- Converted to UTC: 2:00:00 p.m., January 2, 1999
- Number of seconds to event: 262,800 = 73 hours, 0 minutes, 0 seconds

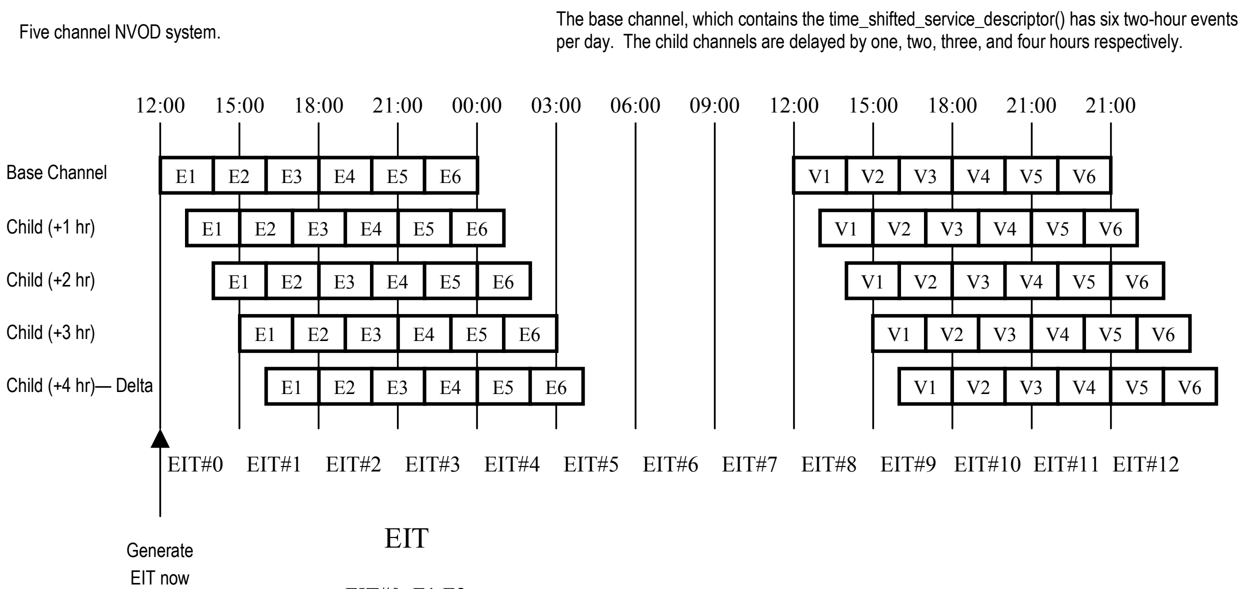
Using method B (upcoming leap second event is anticipated):

- Event start time in EIT: 599,320,813 = 0x23B8E8ED
- Converted to UTC: 2:00:01 p.m., January 2, 1999
- Number of seconds to event: 262,801 = 73 hours, 0 minutes, 1 second

Note that using method B, the number of seconds to event is correct, and does not need to be recomputed when the leap seconds count moves from 12 to 13 at year-end.

8. NVOD EXAMPLES

The examples within this annex describe an NVOD Base channel with four Child channels. (See Figures D10 and D11.) The most delayed Child channel runs four hours after its Base channel.



EIT

EIT#0 E1 E2
EIT#1 E2 E3
EIT#2 E4 E5
EIT#3 E5 E6
EIT#4
EIT#5
EIT#6
EIT#7
EIT#8 V1 V2
EIT#9 V2 V3
EIT#10 V4 V5
EIT#11 V5 V6
EIT#12 V6

Figure D10 NVOD Example no. one.

Notes:

As there are no events that have expired in the base channel, all EIT and ETT entries are the same as those for an ordinary channel.

To find what event starts on channel Delta at 18:00 the steps are:

1) Subtract the channel's time offset (4 hours) from 18:00, giving 14:00.
2) Calculate which EIT window covers 14:00, giving EIT#0.
3) Look in EIT#0 of the base channel for the event at 14:00, giving event E2.

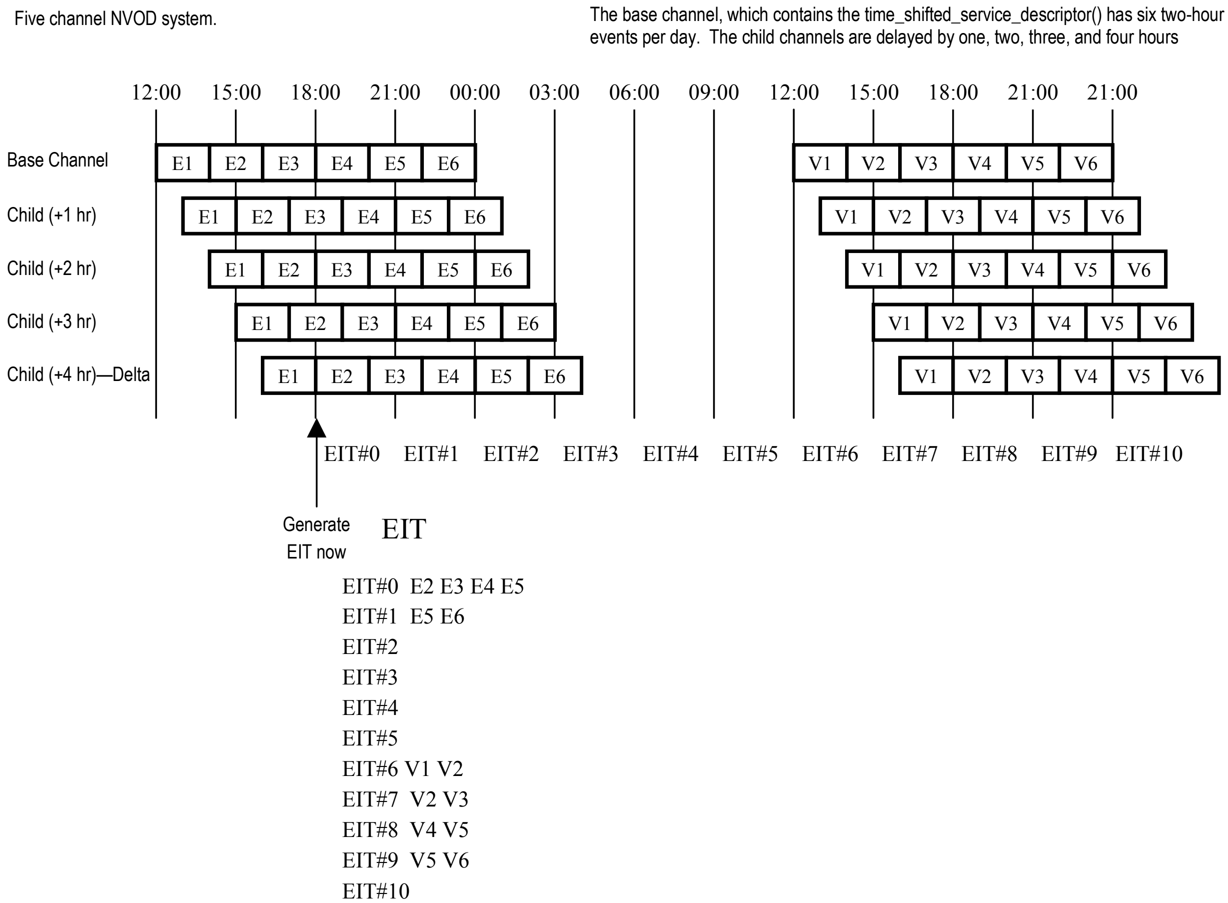


Figure D11 NVOD Example #2

Notes:

E1 has expired on channel Delta (the most delayed child); it is no longer listed in the EIT.

E2 & E3 have not expired on channel Delta (although they have expired on the base channel); they have to be listed in EIT#0.

EIT#1 and above are still the same as for a normal channel.

To find what event starts on channel Delta at 18:00 the steps are:

1) Subtract the channel's time offset (4 hours) from 18:00, giving 14:00.
2) Calculate which EIT window covers 14:00, giving EIT#-2 (i.e., minus two). A negative number is not legal for a window => We must use EIT#0.
3) Look in EIT#0 of the base channel for the event at 14:00, giving event E2.

9. INTERPRETATION OF MGT TABLE VERSION NUMBERS

On first glance, it may appear that the MGT simply provides the version numbers for table sections that make up the EIT/ETT tables for each timeslot. For example, the MGT may indicate a table_type_version_number of 5 for a table_type value of 0x0100 (EIT-0), which could lead one to say "EIT-0 is at version 5." In fact, the MGT does give table version information for all transmitted tables, but a careful and correct interpretation of the data provided, including table_type_PID, must be made to avoid errors in processing.

The proper interpretation of table_type_version_number is to consider it to reflect the version_number field in the referenced table. In accordance with MPEG-2 Systems, the scope of table version_number is limited to table sections delivered in transport packets with a common PID value. For example, for table sections with a given value of table_ID, a table section delivered in transport packets with PID value 0x1E00 and version_number 6 must be interpreted as a separate and distinct table from a table section delivered in transport packets with PID value 0x1E01 and version_number 6.

The following example is designed to illustrate the distinction between the simple (incorrect) interpretation and the correct one. In the illustration, the incorrect interpretation leads to processing errors, which involve re-loading tables that have not in fact changed, or (more seriously) not updating tables that *have* changed.

For the following example, the time zone offset is 0. Each EIT table instance is associated with a separate PID (as per A/65 rules).

1) Say it's noon. From noon to 3 the following is true:
    a) The EIT describing noon to 3 p.m. is in PID[19] 0x1000; version number is 0
    b) The EIT describing 3 p.m. to 6 p.m. is in PID 0x1001; version number is 1
    c) The EIT describing 6 p.m. to 9 p.m, is in PID 0x1002; version number is 0
    d) The EIT describing 9 p.m. to midnight is in PID 0x1003; version number is 0
    e) The MGT is at version 7 and indicates:
        1) EIT-0, PID 0x1000, version number 0
        2) EIT-1, PID 0x1001, version number 1
        3) EIT-2, PID 0x1002, version number 0
        4) EIT-3, PID 0x1003, version number 0
2) The time moves to 3pm, crossing a timeslot boundary. Let's say the EIT describing 6pm to 9 p.m. is changed now too.
    a) The EIT for noon to 3 p.m. is no longer sent, since its time has passed
    b) The EIT for 3 p.m. to 6 p.m. is still in PID 0x1001; version number is still 1
    c) The EIT for 6 p.m. to 9 p.m. is still in PID 0x1002; but its content changed at the same time, so its version number is moved to 1
    d) The EIT for 9 p.m. to midnight is still in PID 0x1003; version number is still 0
    e) MGT moves to version 8 and indicates:
        1) EIT-0, PID 0x1001, version number 1
        2) EIT-1, PID 0x1002, version number 1
        3) EIT-2, PID 0x1003, version number 0

What is now EIT-0 did not change. What is now EIT-1 *did* change.

[19] The expression "in PID" as used here is a shorthand way of saying that the indicated table section is "carried in transport packets with a PID value equal to" the indicated value.

For this case, if the MGT is interpreted to give the version numbers of EIT-n for each value of n, the receiver will see the version of EIT-0 change from 0 to one and refresh it. It will decide the version of EIT-1 has not changed, and not refresh it. But both inferences are incorrect: in this example, EIT-0 has not changed, and EIT-1 *has* changed.

The correct interpretation involves processing version numbers with respect to the associated PID values. Looking at the same example, the MGT indicates that the table associated with PID 0x1001 did not change versions. Likewise, the table associated with PID value 0x1002 changed from version 0 to 1 and should be refreshed.

10. USE OF ANALOG TRANSMISSION SIGNAL ID

The Virtual Channel Table in PSIP associates a user-friendly definition of a service (a channel name and number) with the physical location of that service. Both digital and analog services are accommodated. For digital services, the Transport Stream ID (TSID) parameter defined in ISO/IEC 13818-1 (MPEG-2 Systems) is used as a unique identifier at the TS level. For analog services, an identifier called the Transmission Signal ID (the acronym is also TSID) may be used.

The analog TSID, like its digital counterpart, is a 16-bit number that uniquely identifies the NTSC signal within which it is carried. EIA/CEA-608-B [13] Section 9.5.2.4 defines the data format for carriage of the Transmission Signal ID within eXtended Data Service (XDS) packets in the NTSC Vertical Blanking Interval.

In the US, the DTV system is designed with the expectation that the analog TSID will be included in any NTSC broadcast signal referenced by PSIP data. Whenever PSIP data provides a reference to an analog service, the receiver is expected to use that service's analog TSID to make a positive identification. The receiver is expected to not associate any channel or program information data with an NTSC service that does not broadcast its analog TSID.

11. USE OF COMPONENT NAME DESCRIPTOR

The component_name_descriptor() provides a mechanism to associate a multilingual textual label with an Elementary Stream component of any MPEG-2 program. If the program consists of one video stream and one audio track, such a label would not give much value. A program may be offered multilingually, for example with separate French and English tracks. In that case, a receiving device may choose, without need for user intervention, the track corresponding to the language set up as the user's preferred language.

It may be, however, that the service happens to have two English-language audio tracks of the same audio type (for example both may be Complete Main audio tracks). In another case, one or more of the audio tracks may not be associated with a spoken language. An example of such a track, sometimes called "clean effects," is ambient sound such as crowd noise from a sporting event. In both of these cases, use of the component_name_descriptor() is mandatory by the rules established in this Standard. The net result is that a display device will always have sufficient information to either choose an audio track by its language, by its type, or will have text describing each track that can be used to create an on-screen user dialog to facilitate the user's choice.

Annex E

Typical Size of PSIP Tables

(Informative)

1. OVERVIEW

The typical sizes for the PSIP tables (STT, MGT, VCT, RRT, EIT and ETT) are calculated in this Section. The notation used here for the different equations is listed in the Table E1.

Table E1 Symbols

Symbol	Description
P	number of EITs (4 to 128)
C	number of virtual channels (analog and digital) per EIT
Cd	number of digital channels per EIT
E	number of events per EIT
R	number of rating regions
D	average number of rating dimensions defined per rating region
Dr	average number of rated dimensions per rating region
L	average number of rating values per rating dimension

2. SYSTEM TIME TABLE (STT)

The typical size for the STT is 20 bytes, with the assumption of having no descriptors.

3. MASTER GUIDE TABLE (MGT)

The typical size for the MGT (in bytes), based on the assumptions listed in the column "Assumption", is shown in Table E2

Table E2 Typical Size (bytes) of MGT

Part	Size (bytes)	Assumption
PSI header and trailer	13	
message body	26+22*P	1. With one Terrestrial VCT, one RRT instance, P EITs and P event ETTs 2. No descriptors
Total	39+22*P	

4. TERRESTRIAL VIRTUAL CHANNEL TABLE (TVCT)

The typical size of the TVCT (bytes), based on the assumptions listed in the column labeled "Assumption" is shown in Table E3.

Table E3 Typical TVCT Size (bytes)

Part	Size (bytes)	Assumption
PSI header and trailer	13	1. All TVCT messages are carried in one section.
message body	3+32*C	
extended channel name descriptor	20*C	2. One string and one segment per string for long channel name text. 3. Long channel name text is compressed by Huffman coding with a standard table, and the text length after compression is 10 bytes
service location descriptor	23*Cd	4. Three elementary streams per virtual channel for digital channels.
Total	16+52*C+23*Cd	

5. RATING REGION TABLE (RRT)

The typical size (in bytes per rating region) of the RRT, based on the assumptions listed in the column "Assumption", is shown in Table E4.

Table E4 Typical Size (in bytes per rating region) of RRT

Part	Size (bytes per rating region)	Assumption
PSI header and trailer	13	1. One section only.
message body	24+D*(14+ 26*L)	2. One string and one segment per string for all text. 3. Rating region name text is compressed by Huffman coding with a standard table, and the size after compression is 12 bytes. 4. Dimension name text is compressed by Huffman coding with a standard table, and the size after compression is 4 bytes. 5. Abbreviated rating value text is compressed by Huffman coding with a standard table, and the size after compression is 2 bytes. 6. Rating value text is compressed by Huffman coding with a standard table, and the size after compression is 6 bytes. 7. No descriptors.
Total	37+D*(14+26*L)	

6. EVENT INFORMATION TABLE (EIT)

The typical size of the EIT (in bytes per virtual channel per EIT), based on the assumptions listed in the column "Assumption", is shown in Table E5.

Table E5 Typical Size (bytes per virtual channel per EIT) of EIT

Part	Size (bytes per virtual channel per EIT)	Assumption
PSI header and trailer	13	1. One section only
message body	1+40*E	2. One string and one segment per string for title text. 3. Title text is compressed by Huffman coding with a standard table, and the size after compression is 10 bytes. 4. Two AC-3 descriptors @ 5 bytes each..
closed captioning service descriptor	9*E	5. number_of_services = 1.
content advisory descriptor	(3+R*(3+2*D))*E	6. No rating_description_text.
Total	14+(52+R*(3+2*D))*E	

7. EXTENDED TEXT TABLE (ETT)

The typical size for the ETT (in bytes per event per EIT), based on the assumptions listed in the column labeled "Assumptions," is shown in Table E6.

Table E6 Typical Size (bytes per virtual channel or bytes per event) of ETT

Part	Size (bytes per event per EIT)	Assumptions
PSI header and trailer	13	
message body	92	1. A virtual channel or an event can have one text string and one segment per string for the extended text message. 2. Extended text message is compressed by Huffman coding with a standard table, and the size after compression is 88 bytes. 3. Channel ETTs are typically not needed.
Total	105	

8. DIRECTED CHANNEL CHANGE TABLE (DCCT)

The typical size for the DCCT is 44 bytes, with the assumption of having a single from/to channel, a single selection criterion, and no additional descriptors. The typical size for the DCCT (in bytes) based on the assumptions listed in the column "Assumption" is shown in Table E7.

Table E7 Typical Size (bytes) of DCCT

Part	Size (bytes)	Assumption
PSI header and trailer	13	
message body	3+(17*D)+(11*S)	1. No descriptors. 2. D = number of DCC opportunities defined. 3. S = number of selection criteria.
Total	16+(17*D)+(11*S)	

9. DIRECTED CHANNEL CHANGE SELECTION CODE TABLE (DCCSCT)

The typical size for the DCCSCT is 72 bytes, with the assumption of having four extra genre code categories and no additional descriptors. The typical size for the DCCSCT (in bytes) based on the assumptions listed in the column "Assumptions" is shown in Table E8.

Table E8 Typical Size (bytes) of DCCSCT

Part	Size (bytes)	Assumptions
PSI header and trailer	13	
message body	3+(Sg*(5+9))	1. No descriptors. 2. Sg = number of genre category updates 3. Genre category name is compressed by Huffman coding with a standard table, and the length of the MSS after coding is 9 bytes.
Total	16+(Sg*14)	

10. AN EXAMPLE FOR TERRESTRIAL BROADCAST

Suppose that a TV provider is in charge of two physical transmission channels, one for analog and the other for digital services. Assume that the digital Transport Stream carries five virtual channels (NTSC plus four digital), each with an average of 4.5 events in each of EIT-0, EIT-1, EIT-2 and EIT-3. For each virtual channel an extended channel name descriptor is present. For each event an extended text message is available. Regarding the Rating Region Table, suppose that a different rating region than that given in the example of Annex D Section 4.4 is used, and it is defined with six dimensions and five values per dimension. The example assumes this system and that a typical content advisory uses two of the six dimensions. No Channel ETTs are used.

Based on these assumptions, typical sizes for every PSIP table can be calculated. The results are listed in Table E9 and Table E10.

Table E9 Typical Sizes of PSIP tables (except ETT) for the Example

Part	Size in bytes (excluding Transport Stream packet header)	Size in Transport Stream packets
STT	20	1
MGT	127	1
TVCT	368	2
RRT	901	5
Subtotal for tables identified by the base_PID	1,416	8
EIT-0	1,398	8
EIT-1	1,398	8
EIT-2	1,398	8
EIT-3	1,398	8
Total	7,008	39

Table E10 Typical Sizes of ETTs for the Example

Part	Size in bytes (excluding Transport Stream packet header)	Size in Transport Stream packets
Channel ETT	0	0
Event ETT-0	2,835	16
Event ETT-1	2,835	16
Event ETT-2	2,835	16
Event ETT-3	2,835	16
Total	11,340	62

Annex F

An Overview of Huffman-based Text Compression

(Informative)

1. INTRODUCTION

This section describes the Huffman-based text compression and coding methods supported in the Program and System Information Protocol. In particular, this section:

- Describes the partial first-order Huffman coding used to compress PSIP text data.
- Provides background description of finite-context Huffman coding. The mechanisms for generating and parsing Huffman codes are described.
- Describes the decode tree data structure.
- Defines the character set supported by this Standard.

2. DATA COMPRESSION OVERVIEW

Program and System Information data may use partial first-order Huffman encoding to compress English-language text. The Huffman-table based approach has the following features:

- A typical firmware-resident Huffman decode table requires less than 2K of storage.
- The encode and decode algorithms are relatively simple and fast.
- Since first-order Huffman codes are significantly influenced by language phonetics, codes produced from a sample of current program titles produce reasonable compression ratios for future program titles, even though the future program titles may be significantly different from current titles. Therefore, hard-coded tables stored in receiver non-volatile memory are helpful.

The data compression approach has the following implementation characteristics:

- Program descriptions and program titles may use different Huffman codes. Titles and descriptions have significantly different text characteristics; for example, program titles usually have an upper-case character following a space character, whereas program descriptions usually have a lower-case character following a space-character.
- Hard-coded decode tables, one optimized for titles and one for descriptions, must reside in the receiver's non-volatile memory.

3. OVERVIEW OF CONTEXT-SENSITIVE HUFFMAN CODING

Each and every character does not occur with the same frequency in program titles and program descriptions. For example, the character "e" occurs more often than the character "x." With Huffman coding, the number of bits used to represent a character is inversely proportional to the character's usage frequency.

The Huffman coding compression ratio depends upon the statistical distribution of the characters being compressed. When character usage is uniformly distributed, no compression is achieved with Huffman coding. To achieve satisfactory compression, the Huffman codes are generated using statistics that match the data being compressed. For example, Huffman codes generated from Pascal computer programs would be less than ideal for compressing C programs. For text strings in the PSIP, program descriptions and program titles may be compressed with different sets of Huffman codes.

Context-sensitive Huffman coding recognizes that a character's usage statistics are context dependent. For example, the character "u" has a high probability of occurrence after the character "q". The "order" of the Huffman code defines the "look-back" context by which a character is coded. With order-0, each character is coded independently of the previous character. With order-1, the Huffman code used to represent a given character depends upon the previous character. In zero-order Huffman compression, the occurrence probability of the alphabet elements is used to develop an optimal encoding tree. In first-order Huffman, the conditional probability of a character, given that the previous character is known, is used as the basis of a decoding tree. For this reason, while zero-order Huffman has typically a single tree, first-order Huffman has many, one for each character.

Huffman compression involves the following steps:

- Determine the statistical distribution of the characters or symbols in the source data.
- Create Huffman codes from this statistical information.
- Encode the source data: Translate each character into its corresponding Huffman code.

To decompress the coded data, the data string is parsed bit-by-bit and translated to the original characters. To do this, the decompressor must have the correct decode table, which maps the Huffman codes to their corresponding characters. The following example illustrates the generation and decoding of Huffman codes.

3.1 Example

Huffman codes are mapped to their corresponding characters using a binary tree structure. The leaves of this tree are the alphabet elements to be coded. The tree is produced by recursively summing the two nodes in the tree with the lowest usage frequency. For the following example (Table F1), assume that an alphabet contains the following twelve characters which occur a certain number of times in the sample database:

Table F1 Example Character Set and Frequency of Character Occurrence

Character	Occurrence Number
'a'	144
'b'	66
'c'	30
'd'	30
'e'	18
'f'	12
'g'	6
'h'	1
'i'	1
'j'	1
ESC	arbitrary

The "escape" character is inserted into the table to handle input characters which rarely occur, and have no corresponding Huffman codes. In this example, no Huffman codes will be generated for the characters 'h', 'i', and 'j'. Instead, their frequencies will be summed into the ESC character. Whenever one of these characters occur in the input stream, the encoder inserts the ESC Huffman code, then inserts the original ASCII value for that character.

Figure F1 shows the construction of the Huffman tree from the character frequencies. The two nodes with the lowest frequencies, ('ESC' and 'g'), are joined together, with a resulting node weight of (9). The next two lowest nodes, ('f' and the intermediate node), are then joined together, with the combined weight of (21). This process continues until the tree's root node is formed. Once the tree is completed, the bit (1) is assigned to all right-hand branches, and the bit (0) is assigned to all left-hand branches.

Decoding a Huffman string is straight-forward. Starting at the Huffman tree root, the decoder parses the string, bit by bit, until it reaches a leaf node. The leaf node is the decoded character. The decoder then moves back to the root of the Huffman tree to continue decoding the bit string. For example, the input string 10111011100010 would be decoded into 'beeaab'.

This example uses order-0 Huffman codes. With order-1, each character in the alphabet has an associated tree of Huffman codes for possible succeeding characters. The ESC character would be inserted into each of these order-1 tables to handle statistically unlikely character pairs.

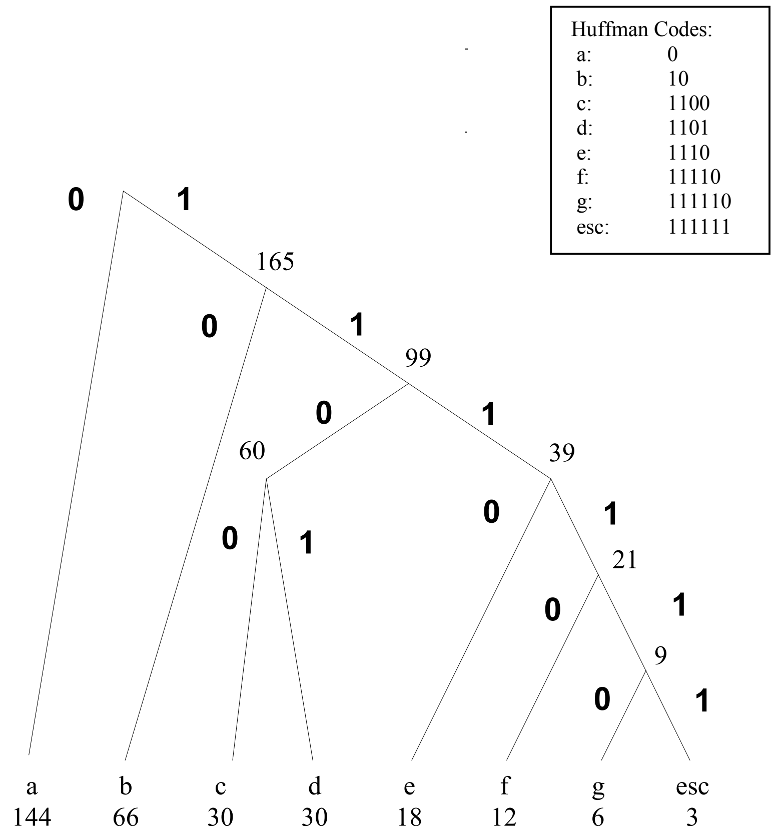


Figure F1 Example Huffman tree.

3.2 Decode Tree Example

Actual implementations of Huffman decoders need to map the trees into a suitable data structure that can be used by a computer or processor to traverse the tree top-down. In Annex C, a possible method for representing the trees was described and explicitly defined. Such a method is used here to build the decoding tree data for the example given in Figure F1. Although an order-0 tree, this table is representative of order-1 decode trees, except that the bytes of each order-1 tree start at a byte location specified by the corresponding tree root offset (rather than starting at location 0), shown in Table F2.

Table F2 Decode Tree Example

Byte #	Left/Right Child Word Offset or Character Leaf	
0 (tree root)	225	(ASCII "a" + 128)
1	1	(word offset of right child)
2 (tree node)	226	(ASCII "b" + 128)
3	2	(word offset of right child)
4 (tree node)	3	(word offset of left child)
5	4	(word offset of right child)
6 (tree node)	227	(ASCII "c" + 128)
7	228	(ASCII "d" + 128)
8 (tree node)	229	(ASCII "e" + 128)
9	5	(word offset of right child)
10 (tree node)	230	(ASCII "f" + 128)
11	6	(word offset of right child)
12 (tree node)	231	(ASCII "g" + 128)
13	155	(ASCII "ESC" + 128)

3.3 Encoding/Character Decoding Examples with 1st-order Huffman tables

As an example of using the Huffman table defined in Table C4 in Annex C, here we show the procedure to encode and decode the string “The next” using the tables optimized for titles. The coding sequence that generates the bit stream for “The next” is described in Figure F2.

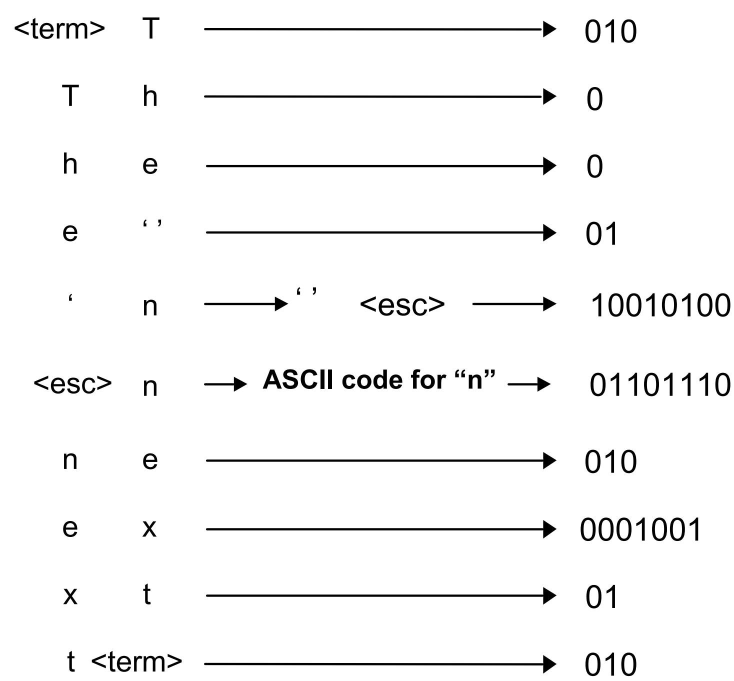


Figure F2 Coding example for the string “The next”.

The first character ‘T’ is encoded assuming that the previous one was a *terminate* character. The second letter ‘h’ is encoded based on the Huffman tree corresponding to the prior symbol ‘T.’ The sequence proceeds as shown in the Figure. The combination blank-space followed by an ‘n’ is not listed in the tree, thus the escape character is used to switch the coding process to uncompressed mode. Once in this mode, the ‘n’ is encoded using its standard 8-bit ISO Latin-1 value. After the ‘n’, an ‘e’ is encoded using the appropriate n-tree and the algorithm continues until reaching the final letter followed by a string-terminate character. Uncompressed transmission of this string requires 9 bytes, while after compression, only 39 bits, equivalent to 5 bytes, are needed.

Decoding requires traversing the different trees top-down. As an example, Figure F3 shows the tree when the prior character is ‘x’. From our example, after decoding the letter ‘x’, the remaining bit sequence is ‘01010’. Traversing the x-tree top-down using this sequence shows that ‘01’ corresponds to ‘t’, a newly decoded character. The process now jumps to the t-tree and so on, to decode the remaining bits until the terminate code results. Notice that the trees can be obtained by examining the encoding tables or by following the semantics of the provided decoding tables.

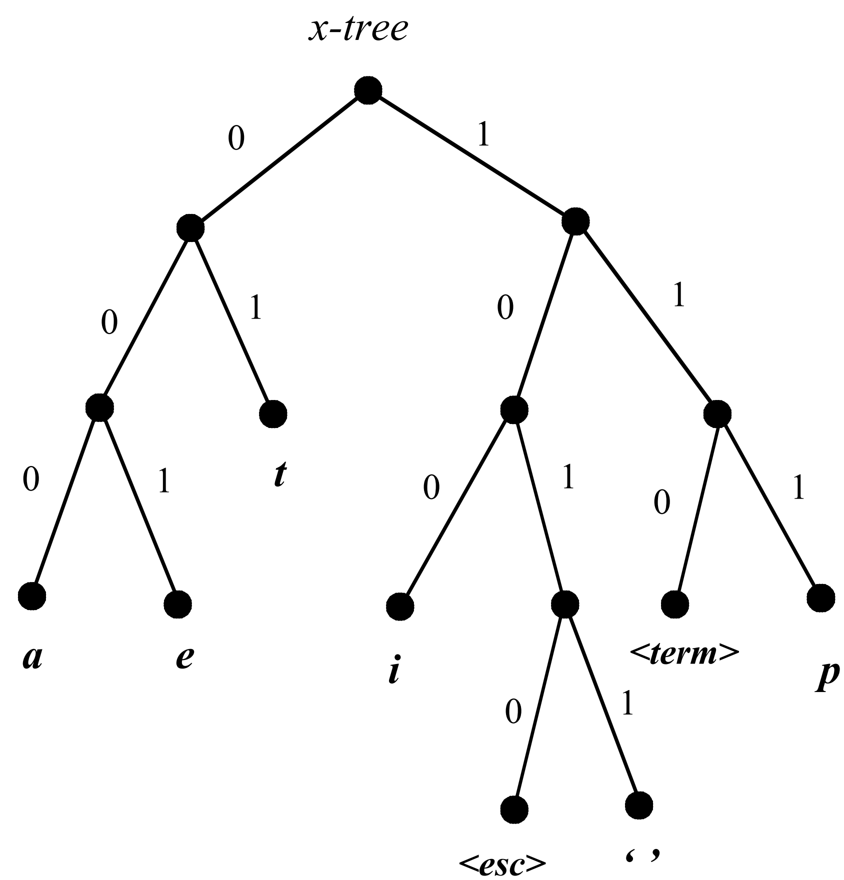


Figure F3 Huffman tree for prior symbol “x”.

Annex G

An Overview of PSIP for Cable

(Informative)

1. INTRODUCTION

As described in this standard, certain data specified in the Program and System Information Protocol (PSIP) forms a mandatory part of every ATSC-compliant digital multiplex signal delivered via terrestrial broadcast. Annex D provides an overview of the use of PSIP for the terrestrial broadcast application. This Annex supplements that discussion, focusing on the use of PSIP for digital cable.

2. OVERVIEW

PSIP was designed, as much as possible, to be independent of the physical system used to deliver the MPEG-2 multiplex. Therefore, the System Time Table, Master Guide Table, Virtual Channel Table (VCT), and Event Information Tables and Extended Text Tables are generally applicable equally as well to cable as to terrestrial broadcast delivery methods. The differences can be summarized as follows:

- For cable, the Cable Virtual Channel Table (CVCT) provides the VCT function, while the Terrestrial Virtual Channel Table (TVCT) applies for terrestrial broadcast. The cable VCT includes two parameters not applicable to the terrestrial broadcast case, and the syntax of several parameters in the table is slightly different for cable as compared to the terrestrial broadcast case. The specifics are discussed in Section 3 of this Annex.
- Use of the program guide portion of PSIP (EIT and ETT) for cable is considered optional, while it is mandatory when PSIP is used for terrestrial broadcasting. Cable operators are free to *not* provide any program guide data at all if they so choose, or provide the data in a format other than PSIP if they do support an EPG.

3. COMPARING CVCT AND TVCT

While the syntax of the Cable and Terrestrial VCTs are nearly identical, the Cable VCT has two parameters not present in the Terrestrial VCT: a "path select" bit, and a bit that can indicate that a given virtual channel is transported out-of-band (OOB). Also, the semantics of the major and minor channel number fields and the source_id differ for the Cable VCT as compared with its terrestrial broadcast counterpart.

3.1 Path Select

Use of the path select is required when PSIP is used in a cable network in which two separate physical cables are present. In such a case, the value of the path_select bit indicates whether the receiver should select the cable connected to its primary port ("path 1") or the secondary cable ("path 2").

3.2 Out of Band

When a cable virtual channel is flagged as being "out of band," it is carried on the out-of-band channel. If a receiver is implemented with a dedicated OOB tuner, it can select and process the OOB multiplex if a user chooses a virtual channel flagged as out_of_band. Receivers not capable of receiving or processing data on out-of-band carriers may use the out_of_band flag as a way to skip or ignore them.

3.3 Major and Minor Channel Numbers

When PSIP is used for terrestrial broadcast, care must be taken in the assignment of major and minor channel numbers to avoid conflicts. For example, the PSIP standard indicates that for the US and its possessions, a terrestrial broadcaster with an existing NTSC license shall use a major channel number for digital services that corresponds to the NTSC RF channel number in present use for the analog signal. For cable, such restrictions are technically unnecessary. The use or potential re-assignment of a broadcaster's major channel number is beyond the scope of this standard. For terrestrial broadcast, the major channel number is limited to the range 1 to 99 for ATSC digital television or audio services. For cable, major channel numbers may range from 1 to 999.

For minor channel numbers, this Standard specifies that zero shall be used for NTSC analog television services, 1 to 99 for ATSC digital television or audio only services, or 1 to 999 for data services. Minor channel numbers for cable, on the other hand, have no restrictions on use: they can range from 0 to 999 for any type of service.

3.4 Source Ids

The source_id parameter is defined identically between cable and terrestrial broadcast VCTs, except that for the cable case, value 0x0000 indicates that the programming source is not identified. Value zero is therefore valid for cable but is reserved (not presently defined) for terrestrial broadcast.

A source ID with value zero is useful for cases where a cable operator wishes to define a channel for which no EPG data is currently available. It would also be useful to an operator who wishes not to supply EPG data at all.

Assignment of a value for source_id for any given service should remain constant over time because changes may disrupt user applications. For example, programs targeted for future storage may be identified by source_id, and any change would result in failure to achieve the desired recording.

4. IN-BAND VERSUS OUT-OF-BAND SYSTEM INFORMATION

Cable operators often make use of one or more out-of-band (OOB) control channels. OOB control gives the operator nearly guaranteed access to each set-top box deployed in a cable network, because a dedicated tuner in each set-top remains tuned to the OOB channel independent of where the user might choose to tune the frequency-agile tuner while accessing various services.

Without an OOB channel, the cable operator either wouldn't be able to supply a uninterrupted stream of control messages to each set-top, or would be forced to carry (redundantly) the same control stream on each analog and digital signal. Duplicating the control stream this way is costly and wasteful of bandwidth. Analog channels in the network pose a problem because there isn't a convenient way to add a channel for control data to each NTSC signal.

PSIP data on cable is provided in-band so that cable-ready consumer electronic equipment can receive navigation data without having to process an OOB channel. Some custom, cable system-specific receiving devices may supplement the PSIP data by making use of other data, provided that the delivery of such data does not conflict with any requirements of the PSIP specification.

5. USING PSIP ON CABLE

PSIP data carried on cable in-band is analogous to PSIP included in the terrestrial digital broadcast multiplex: a receiver can discover the structure of digital services carried on that multiplex by collecting the current VCT from it. A cable-ready digital TV can visit each digital signal on the cable, in sequence, and record from each a portion of the full cable VCT. This is exactly the same process a terrestrial digital broadcast receiver performs to build the terrestrial channel map.

5.1 Terrestrial Virtual Channel Maps on Cable

If a cable operator chooses to deploy digital cable boxes in a cable network, to properly support the cable terminals, that network will need to conform to the transmission and transport standards defined through the Society of Cable Telecommunications Engineers (SCTE). In some instances, however, a small cable operator may offer a cable service in which no cable boxes are required. That operator may wish to implement a low-cost headend where off-air terrestrial broadcasts are simply received and placed onto the cable, as is done with a community antenna scheme such as SMATV. In some cases, signals may be shifted in frequency before being placed on the cable (such as to move a UHF frequency down to the VHF range).

In cases such as these, a receiver may encounter a Terrestrial Virtual Channel Table when acquiring a Transport Stream from the 75 Ω cable port on the receiver. Although that TS does not comply with SCTE standards for digital cable, cable-ready receivers should nonetheless be designed to handle the case where a Terrestrial VCT is found where a Cable VCT is expected.

5.2 Use of the Cable VCT

The carrier_frequency field in the Cable VCT is not used except to identify the location of out-of-band services. Receiving devices must be designed to find and navigate among analog and digital services without reliance on downloaded frequency references.

Cable signals are transmitted in accordance with established frequency plans, so initially discovering the location of each digital or analog carrier is straightforward. PSIP data may describe services carried on the same Transport Stream as the PSIP data itself. The channel_TSID value for these services will match the TSID value found in the PAT of the same Transport Stream.

Whenever PSIP data references a service carried on a different digital Transport Stream or references an NTSC analog service, the channel_TSID should be used to positively identify the target TS or analog service.The recommended approach involves use of a digital signal's Transport Stream ID (TSID) and an analog NTSC signal's Transmission Signal ID (we call this the analog TSID). The FCC is expected to assign each broadcast station operator in the US two unique TSID values, one for analog and one for digital transmission. The digital TSID is defined by the MPEG-2 *Systems* specification, ISO/IEC 13818-1. The analog TSID is defined in EIA/CEA-608-B, and is simply a 16-bit signal identifier that is carried in an Extended Data Service packet.

Upon initial setup by an installer or consumer, a receiver should perform an automatic scan of all frequencies where analog or digital signals may be found.[20] The frequencies used for the scan correspond to standard frequency plans for off-air broadcast or cable, as appropriate. When a signal is found at a given frequency, the receiver should take note of the analog or digital TSID. Although not all analog signals are required to include TSIDs, all digital transport streams are required to carry the unique TSID.

Now, when asked to acquire a specific service, the receiver should use the frequency upon which it was last found.

The data in the modulation field may be in error unless the cable system modifies it. The SCTE has standardized two modulation modes for cable television transmission of digital television. The terrestrial broadcast PSIP shall indicate ATSC 8-VSB modulation for over-the-air transmission of digital television. Any receiver that does not have access to an out-of-band data stream indicating the modulation modes of the various carriers on the network will need to be designed to acquire any of the modes that may be present. In the US, 64-QAM, 256-QAM, 8-VSB or 16-VSB modulation may be encountered.

5.3 Service Location on Cable

The service_location_descriptor() indicates the stream types, PID and language code for each elementary stream that comprises a virtual channel. As mentioned, one of the differences between the terrestrial and cable is that the service_location_descriptor() is not required in the Cable VCT, even though its use is mandatory for the Terrestrial VCT. The difference arises from the fact that cable operators may re-multiplex digital Transport Steams that are available to them, adding or deleting services as necessary to create cable Transport Streams. A motivation for re-multiplexing is that the data rate for information on cable is typically higher than that available from terrestrial broadcast transmissions, and a cable operator may wish to construct multiplexes that make full use of the channel capacity.

For cable, the receiver or set-top box needs to learn the structure of each service via the TS_program_map_section() which contains the same information as the service_location_descriptor() [except for the language code]. ATSC multiplexes are MPEG-2 compliant, and the presence of the TS_program_map_section() is mandatory.

A typical cable receiver or set-top box may implement a scheme where the last-used PID values for audio and video streams are stored with each VCT record. Initial acquisition of a virtual channel may be slower by as much as 400 milliseconds (the maximum interval between repetitions of the TS_program_map_section()) since the TS_program_map_section() will need to be processed to learn the PID values, but subsequent acquisitions can avoid this delay. However, one step in the acquisition process should always be to check the current TS_program_map_section()

[20] It is strongly recommended that such a scan is done also when the receiver is in the "off" state to refresh VCT and program guide data.

to verify that the PID values have not changed since the last acquisition of the service. If they have changed, the new values replace the old.

5.4 Analog Channel Sharing

Some cable operators time-share certain 6-MHz slots between two analog television services, switching from one to the other on a daily schedule. If PSIP were to be used (out of band) to describe such a shared analog channel, two approaches are possible:

Define the channel as a single entity, using one source ID. The channel name may be a combination of the two service names, such as “WXYZ/USTV” for example. Or it could be a neutral name such as “Combo.” Since the channel is defined as a single entity in PSIP, it appears as one horizontal grid line on the EPG display.

Define the channel using two source IDs, one for the first source and another for the second. Using PSIP it would be possible to assign each source a separate channel name. Both would be assigned the same channel number and frequency, corresponding to the channel’s EIA RF 6-MHz band on the cable. Use of the RF channel number is necessary for consistency between DTV receivers using PSIP and analog receivers that tune and number using the conventional analog method. On the EPG grid, each of the services are expected to show “Off the air” (or equivalent) during the part of the broadcast day when the transmission channel is being used for the other source.

The second case represents an unusual situation for the DTV, in that two services share the exact same channel number. If the user selects such a doubly-defined channel by direct entry of the number, the frequency is unambiguous so the DTV can tune straightforwardly. If the DTV would wish to display the proper channel name or program name, it must rely on the analog TSID to properly identify the received signal.

In both of these cases, it is the responsibility of the cable headend to perform source switching as necessary to create the composite channel.

6. RE-MULTIPLEXING ISSUES

As mentioned, a cable operator may wish to take incoming digital Transport Streams from various sources (terrestrial broadcast, satellite, or locally generated), add or delete services or elementary streams, and then re-combine them into output Transport Streams. If the incoming Transport Streams carry PSIP data, care must be taken to properly process this data in the re-multiplexer.

Specifically, the re-multiplexer needs to account for any MPEG or PSIP fields or variables that are scoped to be unique within the Transport Stream. Such fields include PID values, MPEG program_numbers, source_id tags that are in the range 0x0001 through 0x0FFF and event_id fields.

Other PSI and PSIP-related tasks that need to be performed include:

- Construct an output Virtual Channel Table represents the virtual channels that will be included in the resulting Transport Stream.
- Combine EIT and ETT data from the various sources and remove data for any deleted services. (Rules for deleting services are beyond the scope of this standard.)
- The output Rating Region Table includes all regions that the cable operator is either required to support or chooses to support.

- Rebuild the Master Guide Table to represent the resulting PSIP tables.
- The service_location_descriptors present in incoming Terrestrial Virtual Channel Tables may be deleted, and if so should be reconstructed to identify all the services in the Cable Virtual Channel Table for a new transport stream.
- Edit the MPEG-2 Program Map Table to accurately reflect the Transport Stream PID values for all elementary streams in each service.

The special case of remultiplexing without adding or dropping content in the transport stream does not require PSIP modification, although some modification could reduce frequency information inconsistencies.

7. THE TRANSITION TO PSIP ON CABLE

The first digital cable boxes to employ MPEG-2 transport and video coding were deployed in North America beginning in 1996. This PSIP standard was developed and approved by the ATSC in 1997. In 1998, the use of PSIP on cable was balloted and approved by the Digital Video Subcommittee of SCTE.

Cable systems supporting the first digital cable terminals provide an out-of-band control channel for system control and addressing of these boxes. System Information in accordance with ATSC A/56 *System Information for Digital Television*, as extended by the SCTE DVS-011 *Cable and Satellite Extensions to ATSC System Information Standard* provides navigational information such as the cable frequency plan in use, the channel line-up, and channel names and numbers. The A/56 standard used the same virtual channel map approach that PSIP uses.

Cable operators wishing to support cable-ready CE equipment in their network would need to begin sending PSIP data in-band for each digital transport multiplex. PSIP support involves supplying transport stream packets with PID value 0x1FFB. Legacy digital terminals are unaffected by the presence of these new packets, because they have no reason to process data from PID 0x1FFB. Both the PSIP and the A/56 SI data can co-exist in the same system with neither affecting the other.

8. DATA RATES FOR PSIP ON CABLE

The typical sizes of PSIP data in the cable application are computed here. Since the structure of the PSIP tables is unchanged from the terrestrial application, the analysis of table sizes found in Annex E of the PSIP Standard applies equally well to cable. On cable, the service_location_descriptor() is optional, however, so the CVCT data size may be reduced by (23 * Cd) where Cd represents the number of digital services in the multiplex.

If the CVCT is repeated at a rate of 2.5 repetitions per second, and we say that there are 10 digital channels and one reference to an analog channel, the total data rate for each instance of the CVCT is

$$R_{CVCT} = (\text{size of CVCT in bytes}) * (8 \text{ bits/byte}) * (\text{table repetition rate})$$

$$= (16+52*11) * 8 * 2.5 = 11{,}760 \text{ bps}$$

If the MGT is repeated at a rate of one repetition each 150 milliseconds, and it includes references to EIT-0 through –3, the data rate for the MGT is

$$R_{MGT} = (\text{size of MGT in bytes}) * (8 \text{ bits/byte}) * (\text{table repetition rate})$$

$$= 138 * 8 * 1 / .15 = 7360 \text{ bps}$$

An adjustment needs to be made to account for the fact that the MGT must be placed into the transport multiplex such that the first byte of the table (the table_id) aligns with the first byte of the packet payload. If we assume that, on average, half of the prior packet's payload (for the base_PID) will be padded to create this alignment, the data rate for the padding is

$$R_{PAD} = (\text{number of pad bytes}) * (8 \text{ bits/byte}) * (\text{MGT repetition rate})$$

$$= 92 * 8 * 1 / .15 = 4907 \text{ bps}$$

If the RRT is repeated at a rate of one repetition per minute, assuming one region with nine dimensions and an average of four levels per dimension, the data rate is

$$R_{RRT} = (\text{size of RRT in bytes}) * (8 \text{ bits/byte}) * (\text{table repetition rate})$$

$$= (37+9*(14+26*4)) * 8 * 1/60 = 1099 * 8 / 60 = 147 \text{ bps}$$

If the STT is repeated at a rate of once per second the data rate is

$$R_{STT} = 20 * 8 = 160 \text{ bps}$$

So, the total data rate for tables required for the cable application is

$$R_{TOTAL} = R_{CVCT} + R_{MGT} + R_{PAD} + R_{RRT} + R_{STT}$$

$$= 11{,}760 + 7360 + 4907 + 147 + 160 = 19.427 \text{ kbps} \cong 25 \text{ kbps}$$

The analysis can be extended to include the case that EIT/ETT is present in the multiplex as well.

Annex H

Location Codes for Counties and Equivalent Entities of the United States, Its Possessions, and Associated Areas

(Normative)

The state and county codes in this Annex are based upon the codes contained within FIPS Pub 6-4 dated August 31, 1990, and amended by Change Notices 1 through 7, the last dated July 7, 2001. The values within the table in this section take precedence over those documents because of spelling and code errors discovered and fixed while editing and assembling the table. (See next page.)

--- ----
01 ALABAMA (AL)
---- ----
001 Autauga
003 Baldwin
005 Barbour
007 Bibb
009 Blount
011 Bullock
013 Butler
015 Calhoun
017 Chambers
019 Cherokee
021 Chilton
023 Choctaw
025 Clarke
027 Clay
029 Cleburne
031 Coffee
033 Colbert
035 Conecuh
037 Coosa
039 Covington
041 Crenshaw
043 Cullman
045 Dale
047 Dallas
049 DeKalb
051 Elmore
053 Escambia
055 Etowah
057 Fayette
059 Franklin
061 Geneva
063 Greene
065 Hale
067 Henry
069 Houston
071 Jackson
073 Jefferson
075 Lamar
077 Lauderdale
079 Lawrence
081 Lee
083 Limestone
085 Lowndes
087 Macon
089 Madison
091 Marengo
093 Marion
095 Marshall
097 Mobile
099 Monroe
101 Montgomery
103 Morgan
105 Perry
107 Pickens
109 Pike
111 Randolph
113 Russell
115 St. Clair
117 Shelby
119 Sumter
121 Talladega
123 Tallapoosa
125 Tuscaloosa
127 Walker
129 Washington
131 Wilcox
133 Winston

---- ----
02 ALASKA (AK)
---- ----
013 Aleutians East
016 Aleutians West
020 Anchorage
050 Bethel
060 Bristol Bay
068 Denali
070 Dillingham
090 Fairbanks-North Star
100 Haines
110 Juneau
122 Kenai Peninsula
130 Ketchikan Gateway
150 Kodiak Island
164 Lake and Peninsula
170 Matanuska-Susitna
180 Nome
290 Yukon-Koyukuk
185 North Slope
188 Northwest Arctic
201 Prince of Wales-Outer Ketchikan
220 Sitka
232 Skagway-Hoonah-Angoon
240 Southeast Fairbanks
261 Valdez-Gordova
270 Wade Hampton
280 Wrangell-Petersburg
282 Yakutat
290 Yukon-Koyukuk

---- ----
04 ARIZONA (AZ)
---- ----
001 Apache
003 Cochise
005 Coconino
007 Gila
009 Graham
011 Greenlee
012 La Paz
013 Maricopa
015 Mohave
017 Navajo
019 Pima
021 Pinal
023 Santa Cruz
025 Yavapai
027 Yuma

---- ----
05 ARKANSAS (AR)
---- ----
001 Arkansas
003 Ashley
005 Baxter
007 Benton
009 Boone
011 Bradley
013 Calhoun
015 Carroll
017 Chicot
019 Clark
021 Clay
023 Cleburne
025 Cleveland
027 Columbia
029 Conway
031 Craighead
033 Crawford
035 Crittenden
037 Cross
039 Dallas
041 Desha
043 Drew
045 Faulkner
047 Franklin
049 Fulton
051 Garland
053 Grant
055 Greene
057 Hempstead
059 Hot Spring
061 Howard
063 Independence
065 Izard
067 Jackson
069 Jefferson
071 Johnson
073 Lafayette
075 Lawrence
077 Lee
079 Lincoln
081 Little River
083 Logan
085 Lonoke
087 Madison
089 Marion
091 Miller
093 Mississippi
095 Monroe
097 Montgomery
099 Nevada
101 Newton
103 Ouachita
105 Perry
107 Phillips
109 Pike
111 Poinsett
113 Polk
115 Pope
117 Prairie
119 Pulaski
121 Randolph
123 St. Francis
125 Saline
127 Scott
129 Searcy
131 Sebastian
133 Sevier
135 Sharp
137 Stone
139 Union
141 Van Buren
143 Washington
145 White
147 Woodruff
149 Yell

---- ----
06 CALIFORNIA (CA)
---- ----
001 Alameda
003 Alpine
005 Amador
007 Butte
009 Calaveras
011 Colusa
013 Contra Costa
015 Del Norte
017 El Dorado
019 Fresno
021 Glenn
023 Humboldt
025 Imperial
027 Inyo
029 Kern
031 Kings
033 Lake
035 Lassen
037 Los Angeles
039 Madera
041 Marin
043 Mariposa
045 Mendocino
047 Merced
049 Modoc
051 Mono
053 Monterey
055 Napa
057 Nevada
059 Orange
061 Placer
063 Plumas
065 Riverside
067 Sacramento
069 San Benito
071 San Bernardino
073 San Diego
075 San Francisco
077 San Joaquin
079 San Luis Obispo
081 San Mateo
083 Santa Barbara
085 Santa Clara
087 Santa Cruz
089 Shasta
091 Sierra
093 Siskiyou
095 Solano
097 Sonoma
099 Stanislaus
101 Sutter
103 Tehama
105 Trinity
107 Tulare
109 Tuolumne
111 Ventura
113 Yolo
115 Yuba

---- ----
08 COLORADO (CO)
---- ----
001 Adams
003 Alamosa
005 Arapahoe
007 Archuleta
009 Baca
011 Bent
013 Boulder
014 Broomfield
015 Chaffee
017 Cheyenne
019 Clear Creek
021 Conejos
023 Costilla
025 Crowley
027 Custer
029 Delta
031 Denver
033 Dolores
035 Douglas
037 Eagle
039 Elbert
041 El Paso
043 Fremont
045 Garfield
047 Gilpin
049 Grand
051 Gunnison
053 Hinsdale
055 Huerfano
057 Jackson
059 Jefferson
061 Kiowa
063 Kit Carson
065 Lake
067 La Plata
069 Larimer
071 Las Animas
073 Lincoln
075 Logan
077 Mesa
079 Mineral
081 Moffat
083 Montezuma
085 Montrose
087 Morgan
089 Otero
091 Ouray
093 Park
095 Phillips
097 Pitkin
099 Prowers
101 Pueblo
103 Rio Blanco
105 Rio Grande
107 Routt
109 Saguache
111 San Juan
113 San Miguel
115 Sedgwick
117 Summit
119 Teller
121 Washington
123 Weld
125 Yuma

---- ----
09 CONNECTICUT (CT)
---- ----
001 Fairfield
003 Hartford
005 Litchfield

007 Middlesex
009 New Haven
011 New London
013 Tolland
015 Windham

---- ----
10 DELAWARE (DE)
---- ----
001 Kent
003 New Castle
005 Sussex

---- ----
11 DISTRICT OF COLUMBIA (DC)
---- ----
001 District of Columbia

---- ----
12 FLORIDA (FL)
---- ----
001 Alachua
003 Baker
005 Bay
007 Bradford
009 Brevard
011 Broward
013 Calhoun
015 Charlotte
017 Citrus
019 Clay
021 Collier
023 Columbia
025 Dade
027 DeSoto
029 Dixie
031 Duval
033 Escambia
035 Flagler
037 Franklin
039 Gadsden
041 Gilchrist
043 Glades
045 Gulf
047 Hamilton
049 Hardee
051 Hendry
053 Hernando
055 Highlands
057 Hillsborough
059 Holmes
061 Indian River
063 Jackson
065 Jefferson
067 Lafayette
069 Lake
071 Lee
073 Leon
075 Levy
077 Liberty
079 Madison
081 Manatee
083 Marion
085 Martin
086 Miami-Dade
087 Monroe
089 Nassau
091 Okaloosa
093 Okeechobee
095 Orange
097 Osceola
099 Palm Beach
101 Pasco
103 Pinellas
105 Polk
107 Putnam
109 St. Johns
111 St. Lucie
113 Santa Rosa
115 Sarasota
117 Seminole
119 Sumter
121 Suwannee
123 Taylor
125 Union
127 Volusia
129 Wakulla
131 Walton
133 Washington

---- ----
13 GEORGIA (GA)
---- ----
001 Appling
003 Atkinson
005 Bacon
007 Baker
009 Baldwin
011 Banks
013 Barrow
015 Bartow
017 Ben Hill
019 Berrien
021 Bibb
023 Bleckley
025 Brantley
027 Brooks
029 Bryan
031 Bulloch
033 Burke
035 Butts
037 Calhoun
039 Camden
043 Candler
045 Carroll
047 Catoosa
049 Charlton
051 Chatham
053 Chattahoochee
055 Chattooga
057 Cherokee
059 Clarke
061 Clay
063 Clayton
065 Clinch
067 Cobb
069 Coffee
071 Colquitt
073 Columbia
075 Cook
077 Coweta
079 Crawford
081 Crisp
083 Dade
085 Dawson
087 Decatur
089 DeKalb
091 Dodge
093 Dooly
095 Dougherty
097 Douglas
099 Early
101 Echols
103 Effingham
105 Elbert
107 Emanuel
109 Evans
111 Fannin
113 Fayette
115 Floyd
117 Forsyth
119 Franklin
121 Fulton
123 Gilmer
125 Glascock
127 Glynn
129 Gordon
131 Grady
133 Greene
135 Gwinnett
137 Habersham
139 Hall
141 Hancock
143 Haralson
145 Harris
147 Hart
149 Heard
151 Henry
153 Houston
155 Irwin
157 Jackson
159 Jasper
161 Jeff Davis
163 Jefferson
165 Jenkins
167 Johnson
169 Jones
171 Lamar
173 Lanier
175 Laurens
177 Lee
179 Liberty
181 Lincoln
183 Long
185 Lowndes
187 Lumpkin
189 McDuffie
191 McIntosh
193 Macon
195 Madison
197 Marion
199 Meriwether
201 Miller
205 Mitchell
207 Monroe
209 Montgomery
211 Morgan
213 Murray
215 Muscogee
217 Newton
219 Oconee
221 Oglethorpe
223 Paulding
225 Peach
227 Pickens
229 Pierce
231 Pike
233 Polk
235 Pulaski
237 Putnam
239 Quitman
241 Rabun
243 Randolph
245 Richmond
247 Rockdale
249 Schley
251 Screven
253 Seminole
255 Spalding
257 Stephens
259 Stewart
261 Sumter
263 Talbot
265 Taliaferro
267 Tattnall
269 Taylor
271 Telfair
273 Terrell
275 Thomas
277 Tift
279 Toombs
281 Towns
283 Treutlen
285 Troup
287 Turner
289 Twiggs
291 Union
293 Upson
295 Walker
297 Walton
299 Ware
301 Warren
303 Washington
305 Wayne
307 Webster
309 Wheeler
311 White
313 Whitfield
315 Wilcox
317 Wilkes
319 Wilkinson
321 Worth

---- ----
15 HAWAII (HI)
---- ----
001 Hawaii
003 Honolulu
005 Kalawao
007 Kauai
009 Maui

---- ----
16 IDAHO (ID)
---- ----
001 Ada
003 Adams
005 Bannock
007 Bear Lake
009 Benewah
011 Bingham
013 Bane
015 Boise
017 Bonner
019 Bonneville
021 Boundary
023 Butte
025 Camas
027 Canyon
029 Caribou
031 Cassia
033 Clark
035 Clearwater
037 Custer
039 Elmore
041 Franklin
043 Fremont
045 Gem
047 Gooding
049 Idaho
051 Jefferson
053 Jerome
055 Kootenai
057 Latah
059 Lemhi
061 Lewis
063 Lincoln
065 Madison
067 Minidoka
069 Nez Perce
071 Oneida
073 Owyhee
075 Payette
077 Power
079 Shoshone
081 Teton
083 Twin Falls
085 Valley
087 Washington

---- ----
17 ILLINOIS (IL)
---- ----
001 Adams
003 Alexander
005 Bond
007 Boone
009 Brown
011 Bureau
013 Calhoun
015 Carroll
017 Cass
019 Champaign
021 Christian
023 Clark
025 Clay
027 Clinton
029 Coles
031 Cook
033 Crawford
035 Cumberland
037 DeKalb
039 De Witt
041 Douglas
043 DuPage
045 Edgar
047 Edwards
049 Effingham
051 Fayette
053 Ford
055 Franklin
057 Fulton
059 Gallatin
061 Greene
063 Grundy
065 Hamilton
067 Hancock
069 Hardin
071 Henderson
073 Henry
075 Iroquois
077 Jackson
079 Jasper
081 Jefferson

083 Jersey
085 Jo Daviess
087 Johnson
089 Kane
091 Kankakee
093 Kendall
095 Knox
097 Lake
099 La Salle
101 Lawrence
103 Lee
105 Livingston
107 Logan
109 McDonough
111 McHenry
113 McLean
115 Macon
117 Macoupin
119 Madison
121 Marion
123 Marshall
125 Mason
127 Massac
129 Menard
131 Mercer
133 Monroe
135 Montgomery
137 Morgan
139 Moultrie
141 Ogle
143 Peoria
145 Perry
147 Piatt
149 Pike
151 Pope
153 Pulaski
155 Putnam
157 Randolph
159 Richland
161 Rock Island
163 St. Clair
165 Saline
167 Sangamon
169 Schuyler
171 Scott
173 Shelby
175 Stark
177 Stephenson
179 Tazewell
181 Union
183 Vermilion
185 Wabash
187 Warren
189 Washington
191 Wayne
193 White
195 Whiteside
197 Will
199 Williamson
201 Winnebago
203 Woodford

---- ----
18 INDIANA (IN)
---- ----
001 Adams
003 Allen
005 Bartholomew
007 Benton
009 Blackford
011 Boone
013 Brown
015 Carroll
017 Cass
019 Clark
021 Clay
023 Clinton
025 Crawford
027 Daviess
029 Dearborn
031 Decatur
033 De Kalb
035 Delaware
037 Dubois
039 Elkhart
041 Fayette
043 Floyd
045 Fountain
047 Franklin
049 Fulton
051 Gibson
053 Grant
055 Greene
057 Hamilton
059 Hancock
061 Harrison
063 Hendricks
065 Henry
067 Howard
069 Huntington
071 Jackson
073 Jasper
075 Jay
077 Jefferson
079 Jennings
081 Johnson
083 Knox
085 Kosciusko
087 Lagrange
089 Lake
091 La Porte
093 Lawrence
095 Madison
097 Marion
099 Marshall
101 Martin
103 Miami
105 Monroe
107 Montgomery
109 Morgan
111 Newton
113 Noble
115 Ohio
117 Orange
119 Owen
121 Parke
123 Perry
125 Pike
127 Porter
129 Posey
131 Pulaski
133 Putnam
135 Randolph
137 Ripley
139 Rush
141 St. Joseph
143 Scott
145 Shelby
147 Spencer
149 Starke
151 Steuben
153 Sullivan
155 Switzerland
157 Tippecanoe
159 Tipton
161 Union
163 Vanderburgh
165 Vermillon
167 Vigo
169 Wabash
171 Warren
173 Warrick
175 Washington
177 Wayne
179 Wells
181 White
183 Whitley

---- ----
19 IOWA (IA)
---- ----
001 Adair
003 Adams
005 Allamakee
007 Appanoose
009 Audubon
011 Benton
013 Black Hawk
015 Boone
017 Bremer
019 Buchanan
021 Buena Vista
023 Butler
025 Calhoun
027 Carroll
029 Cass
031 Cedar
033 Cerro Gordo
035 Cherokee
037 Chickasaw
039 Clarke
041 Clay
043 Clayton
045 Clinton
047 Crawford
049 Dallas
051 Davis
053 Decatur
055 Delaware
057 Des Moines
059 Dickinson
061 Dubuque
063 Emmet
065 Fayette
067 Floyd
069 Franklin
071 Fremont
073 Greene
075 Grundy
077 Guthrie
079 Hamilton
081 Hancock
083 Hardin
085 Harrison
087 Henry
089 Howard
091 Humboldt
093 Ida
095 Iowa
097 Jackson
099 Jasper
101 Jefferson
103 Johnson
105 Jones
107 Keokuk
109 Kossuth
111 Lee
113 Linn
115 Louisa
117 Lucas
119 Lyon
121 Madison
123 Mahaska
125 Marion
127 Marshall
129 Mills
131 Mitchell
133 Monona
135 Monroe
137 Montgomery
139 Muscatine
141 O'Brien
143 Osceola
145 Page
147 Palo Alto
149 Plymouth
151 Pocahontas
153 Polk
155 Pottawattamie
157 Poweshiek
159 Ringgold
161 Sac
163 Scott
165 Shelby
167 Sioux
169 Story
171 Tama
173 Taylor
175 Union
177 Van Buren
179 Wapello
181 Warren
183 Washington
185 Wayne
187 Webster
189 Winnebago
191 Winneshiek
193 Woodbury
195 Worth
197 Wright

---- ----
20 KANSAS (KS)
---- ----
001 Allen
003 Anderson
005 Atchison
007 Barber
009 Barton
011 Bourbon
013 Brown
015 Butler
017 Chase
019 Chautauqua
021 Cherokee
023 Cheyenne
025 Clark
027 Clay
029 Cloud
031 Coffey
033 Comanche
035 Cowley
037 Crawford
039 Decatur
041 Dickinson
043 Doniphan
045 Douglas
047 Edwards
049 Elk
051 Ellis
053 Ellsworth
055 Finney
057 Ford
059 Franklin
061 Geary
063 Gove
065 Graham
067 Grant
069 Gray
071 Greeley
073 Greenwood
075 Hamilton
077 Harper
079 Harvey
081 Haskell
083 Hodgeman
085 Jackson
087 Jefferson
089 Jewell
091 Johnson
093 Kearny
095 Kingman
097 Kiowa
099 Labette
101 Lane
103 Leavenworth
105 Lincoln
107 Linn
109 Logan
111 Lyon
113 McPherson
115 Marion
117 Marshall
119 Meade
121 Miami
123 Mitchell
125 Montgomery
127 Morris
129 Morton
131 Nemaha
133 Neosho
135 Ness
137 Norton
139 Osage
141 Osborne
143 Ottawa
145 Pawnee
147 Phillips
149 Pottawatomie
151 Pratt
153 Rawlins
155 Reno
157 Republic
159 Rice
161 Riley
163 Rooks
165 Rush
167 Russell
169 Saline
171 Scott
173 Sedgwick
175 Seward
177 Shawnee
179 Sheridan
181 Sherman
183 Smith
185 Stafford
187 Stanton
189 Stevens
191 Sumner

193 Thomas
195 Trego
197 Wabaunsee
199 Wallace
201 Washington
203 Wichita
205 Wilson
207 Woodson
209 Wyandotte

---- ----
21 KENTUCKY (KY)
---- ----
001 Adair
003 Allen
005 Anderson
007 Ballard
009 Barren
011 Bath
013 Bell
015 Boone
017 Bourbon
019 Boyd
021 Boyle
023 Bracken
025 Breathitt
027 Breckinridge
029 Bullitt
031 Butler
033 Caldwell
035 Calloway
037 Campoell
039 Carlisle
041 Carroll
043 Carter
045 Casey
047 Christian
049 Clark
051 Clay
053 Clinton
055 Crittenden
057 Cumberland
059 Daviess
061 Edmonson
063 Elliott
065 Estill
067 Fayette
069 Fleming
071 Floyd
073 Franklin
075 Fulton
077 Gallatin
079 Garrard
081 Grant
083 Graves
085 Grayson
087 Green
089 Greenup
091 Hancock
093 Hardin
095 Harlan
097 Harrison
099 Hart
101 Henderson
103 Henry
105 Hickman
107 Hopkins
109 Jackson
111 Jefferson
113 Jessamine
115 Johnson
117 Kenton
119 Knott
121 Knox
123 Larue
125 Laurel
127 Lawrence
129 Lee
131 Leslie
133 Letcher
135 Lewis
137 Lincoln
139 Livingston
141 Logan
143 Lyon
145 McCracken
147 McCreary
149 McLean
151 Madison
153 Magoffin
155 Marion
157 Marshall
159 Martin
161 Mason
163 Meade
165 Menifee
167 Mercer
169 Metcalfe
171 Monroe
173 Montgomery
175 Morgan
177 Muhlenberg
179 Nelson
181 Nicholas
183 Ohio
185 Oldham
187 Owen
189 Owsley
191 Pendleton
193 Perry
195 Pike
197 Powell
199 Pulaski
201 Robertson
203 Rockcastle
205 Rowan
207 Russell
209 Scott
211 Shelby
213 Simpson
215 Spencer
217 Taylor
219 Todd
221 Trigg
223 Trimble
225 Union
227 Warren
229 Washington
231 Wayne
233 Webster
235 Whitley
237 Wolfe
239 Woodford

---- ----
22 LOUISIANA (LA)
---- ----
001 Acadia
003 Allen
005 Ascension
007 Assumption
009 Avoyelles
011 Beauregard
013 Bienville
015 Bossier
017 Caddo
019 Calcasieu
021 Caldwell
023 Cameron
025 Catahoula
025 Catahoula
029 Concordia
031 De Soto
033 E. Baton Rouge
035 E. Carroll
037 E. Feliciana
039 Evangeline
041 Franklin
043 Grant
045 Iberia
047 Iberville
049 Jackson
051 Jefferson
053 Jefferson Davis
055 Lafayette
057 Lafourche
059 La Salle
061 Lincoln
063 Livingston
065 Madison
067 Wilkin
069 Winona
071 Orleans
073 Ouachita
075 Plaquemines
077 Pointe Coupee
079 Rapides
081 Red River
083 Richland
085 Sabine
087 St. Bernard
089 St. Charles
091 St. Helena
093 St. James
095 St. John the Baptist
097 St. Landry
099 St. Martin
101 St. Mary
103 St. Tammany
105 Tangipahoa
107 Tensas
109 Terrebonne
111 Union
113 Vermilion
115 Vernon
117 Washington
119 Webster
121 W. Baton Rouge
123 W. Carroll
125 W. Feliciana
127 Winn

---- ----
23 MAINE (ME)
---- ----
001 Androscoggin
003 Aroostook
005 Cumberland
007 Franklin
009 Hancock
011 Kennebec
013 Knox
015 Lincoln
017 Oxford
019 Penobscot
021 Piscataquis
023 Sagadahoc
025 Somerset
027 Waldo
029 Washington
031 York

---- ----
24 MARYLAND (MD)
---- ----
001 Allegany
003 Anne Arundel
005 Baltimore
009 Calvert
011 Caroline
013 Carroll
015 Cecil
017 Charles
019 Dorchester
021 Frederick
023 Garrett
025 Harford
027 Howard
029 Kent
031 Montgomery
033 Prince George's
035 Queen Anne's
037 St. Mary's
039 Somerset
041 Talbot
043 Washington
045 Wicomico
047 Worcester
510 Baltimore (city)

---- ----
25 MASSACHUSETTS (MA)
---- ----
001 Barnstable
003 Berkshire
005 Bristo
007 Dukes
009 Essex
011 Franklin
013 Hampden
015 Hampshire
017 Middlesex
019 Nantucket
021 Norfolk
023 Plymouth
025 Suffolk
027 Worcester

---- ----
26 MICHIGAN (MI)
---- ----
001 Alcona
003 Alger
005 Allegan
007 Alpena
009 Antrim
011 Arenac
013 Baraga
015 Barry
017 Bay
019 Benzie
021 Berrien
023 Branch
025 Calhoun
027 Cass
029 Charlevoix
031 Cheboygan
033 Chippewa
035 Clare
037 Clinton
039 Crawford
041 Delta
043 Dickinson
045 Eaton
047 Emmet
049 Genesee
051 Gladwin
053 Gogebic
055 Grand Traverse
057 Gratiot
059 Hillsdale
061 Houghton
063 Huron
065 Ingham
067 Ionia
069 Iosco
071 Iron
073 Isabella
075 Jackson
077 Kalamazoo
079 Kalkaska
081 Kent
083 Keweenaw
085 Lake
087 Lapeer
089 Leelanau
091 Lenawee
093 Livingston
095 Luce
097 Mackinac
099 Macomb
101 Manistee
103 Marquette
105 Mason
107 Mecosta
109 Menominee
111 Midland
113 Missaukee
115 Monroe
117 Montcalm
119 Montmorency
121 Muskegon
123 Newaygo
125 Oakland
127 Oceana
129 Ogemaw
131 Ontonagon
133 Osceola
135 Oscoda
137 Otsego
139 Ottawa
141 Presque Isle
143 Roscommon
145 Saginaw
147 St. Clair
149 St. Joseph
151 Sanilac
153 Schoolcraft

155 Shiawassee
157 Tuscola
159 Van Buren
161 Washtenaw
163 Wayne
165 Wexford

---- ----
27 MINNESOTA (MN)
---- ----
001 Aitkin
003 Anoka
005 Becker
007 Beltrami
009 Benton
011 Big Stone
013 Blue Earth
015 Brown
017 Carlton
019 Carver
021 Cass
023 Chippewa
025 Chisago
027 Clay
029 Clearwater
031 Cook
033 Cottonwood
035 Crow Wing
037 Dakota
039 Dodge
041 Douglas
043 Faribault
045 Fillmore
047 Freeborn
049 Goodhue
051 Grant
053 Hennepin
055 Houston
057 Hubbard
059 Isanti
061 Itasca
063 Jackson
065 Kanabec
067 Kandiyohi
069 Kittson
071 Koochiching
073 Lac qui Pare
075 Lake
077 Lake of the Woods
079 Le Sueur
081 Lincoln
083 Lyon
085 McLeod
087 Mahnomen
089 Marshall
091 Martin
093 Meeker
095 Mille Lacs
097 Morrison
099 Mower
101 Murray
103 Nicollet
105 Nobles
107 Norman
109 Olmsted
111 Otter Tail
113 Pennington
115 Pine
117 Pipestone
119 Polk
121 Pope
123 Ramsey
125 Red Lake
127 Redwood
129 Renville
131 Rice
133 Rock
135 Roseau
137 St. Louis
139 Scott
141 Sherburne
143 Sibley
145 Stearns
147 Steele
149 Stevens
151 Swift
153 Todd
155 Traverse
157 Wabasha
159 Wadena
161 Waseca
163 Washington
165 Watonwan
167 Wilkin
169 Winona
171 Wright
173 Yellow Medicine

---- ----
28 MISSISSIPPI (MS)
---- ----
001 Adams
003 Alcorn
005 Amite
007 Attala
009 Benton
011 Bolivar
013 Calhoun
015 Carroll
017 Chickasaw
019 Choctaw
021 Claiborne
023 Clarke
025 Clay
027 Coahoma
029 Copiah
031 Covington
033 DeSoto
035 Forrest
037 Franklin
039 George
041 Greene
043 Grenada
045 Hancock
047 Harrison
049 Hinds
051 Holmes
053 Humphreys
055 Issaquena
057 Itawamba
059 Jackson
061 Jasper
063 Jefferson
065 Jefferson Davis
067 Jones
069 Kemper
071 Lafayette
073 Lamar
075 Lauderdale
077 Lawrence
079 Leake
081 Lee
083 Leflore
085 Lincoln
087 Lowndes
089 Madison
091 Marion
093 Marshall
095 Monroe
097 Montgomery
101 Newton
103 Noxubee
105 Oktibbeha
107 Panola
109 Pearl River
111 Perry
113 Pike
115 Pontotoc
117 Prentiss
119 Quitman
121 Rankin
123 Scott
125 Sharkey
127 Simpson
129 Smith
131 Stone
133 Sunflower
135 Tallahatchie
137 Tate
139 Tippah
141 Tishomingo
143 Tunica
145 Union
147 Walthall
149 Warren
151 Washington
153 Wayne
155 Webster
157 Wilkinson
159 Winston
161 Yalobusha
163 Yazoo

---- ----
29 MISSOURI (MO)
---- ----
001 Adair
003 Andrew
005 Atchison
007 Audrain
009 Barry
011 Barton
013 Bates
015 Benton
017 Bollinger
019 Boone
021 Buchanan
023 Butler
025 Caldwell
027 Callaway
029 Camden
031 Cape Girardeau
033 Carroll
035 Carter
037 Cass
039 Cedar
041 Chariton
043 Christian
045 Clark
047 Clay
049 Clinton
051 Cole
053 Cooper
055 Crawford
057 Dade
059 Dallas
061 Daviess
063 DeKalb
065 Dent
067 Douglas
069 Dunklin
071 Franklin
073 Gasconade
075 Gentry
077 Greene
079 Grundy
081 Harrison
083 Henry
085 Hickory
087 Holt
089 Howard
091 Howell
093 Iron
095 Jackson
097 Jasper
099 Jefferson
101 Johnson
103 Knox
105 Laclede
107 Lafayette
109 Lawrence
111 Lewis
113 Lincoln
115 Linn
117 Livingston
119 McDonald
121 Macon
123 Madison
125 Maries
127 Marion
129 Mercer
131 Miller
133 Mississippi
135 Moniteau
137 Monroe
139 Montgomery
141 Morgan
143 New Madrid
145 Newton
147 Nodaway
149 Oregon
151 Osage
153 Ozark
155 Pemiscot
157 Perry
159 Pettis
161 Phelps
163 Pike
165 Platte
167 Polk
169 Pulaski
171 Putnam
173 Ralls
175 Randolph
177 Ray
179 Reynolds
181 Ripley
183 St. Charles
185 St. Clair
186 Ste. Genevieve
187 St. Francois
189 St. Louis
195 Saline
197 Schuyler
199 Scotland
201 Scott
203 Shannon
205 Shelby
207 Stoddard
209 Stone
211 Sullivan
213 Taney
215 Texas
217 Vernon
219 Warren
221 Washington
223 Wayne
225 Webster
227 Worth
229 Wright
510 St. Louis (city)

---- ----
30 MONTANA (MT)
---- ----
001 Beaverhead
003 Big Horn
005 Blaine
007 Broadwater
009 Carbon
011 Carter
013 Cascade
015 Chouteau
017 Custer
019 Daniels
021 Dawson
023 Deer Lodge
025 Fallon
027 Fergus
029 Flathead
031 Gallatin
033 Garfield Park
035 Glacier
037 Golden Valley
039 Granite
041 Hill
043 Jefferson
045 Judith Basin
047 Lake
049 Lewis & Clark
051 Liberty
053 Lincoln
055 McCone
057 Madison
059 Meagher
061 Mineral
063 Missoula
065 Musselshell
067 Park
069 Petroleum
071 Phillips
073 Pondera
075 Powder River
077 Powell
079 Prairie
081 Ravalli
083 Richland
085 Roosevelt
087 Rosebud
089 Sanders
091 Sheridan

093 Silver Bow
095 Stillwater
097 Sweet Grass
099 Teton
101 Toole
103 Treasure
105 Valley
107 Wheatland
109 Wibaux
111 Yellowstone

---- ----
31 NEBRASKA (NE)
---- ----
001 Adams
003 Antelope
005 Arthur
007 Banner
009 Blaine
011 Boone
013 Box Butte
015 Boyd
017 Brown
019 Buffalo
021 Burt
023 Butler
025 Cass
027 Cedar
029 Chase
031 Cherry
033 Cheyenne
035 Clay
037 Colfax
039 Cuming
041 Custer
043 Dakota
045 Dawes
047 Dawson
049 Deuel
051 Dixon
053 Dodge
055 Douglas
057 Dundy
059 Fillmore
061 Franklin
063 Frontier
065 Furnas
067 Gage
069 Garden
071 Garfield
073 Gosper
075 Grant
077 Greeley
079 Hall
081 Hamilton
083 Harlan
085 Hayes
087 Hitchcock
089 Holt
091 Hooker
093 Howard
095 Jefferson
097 Johnson
099 Kearney
101 Keith
103 Keya Paha
105 Kimball
107 Knox
109 Lancaster
111 Lincoln
113 Logan
115 Loup
117 McPherson
119 Madison
121 Merrick
123 Morrill
125 Nance
127 Nemaha
129 Nuckolls
131 Otoe
133 Pawnee
135 Perkins
137 Phelps
139 Pierce
141 Platte
143 Polk
145 Red Willow
147 Richardson
149 Rock
151 Saline
153 Sarpy
155 Saunders
157 Scotts Bluff
159 Seward
161 Sheridan
163 Sherman
165 Sioux
167 Stanton
169 Thayer
171 Thomas
173 Thurston
175 Valley
177 Washington
179 Wayne
181 Webster
183 Wheeler
185 York

---- ----
32 NEVADA (NV)
---- ----
001 Churchill
003 Clark
005 Douglas
007 Elko
009 Esmeralda
011 Eureka
013 Humboldt
015 Lander
017 Lincoln
019 Lyon
021 Mineral
023 Nye
027 Pershing
029 Storey
031 Washoe
033 White Pine
510 Carson City

---- ----
33 NEW HAMPSHIRE (NH)
---- ----
001 Belknap
003 Carroll
005 Cheshire
007 Coos
009 Grafton
011 Hillsborough
013 Merrimack
015 Rockingham
017 Strafford
019 Sullivan

---- ----
34 NEW JERSEY (NJ)
---- ----
001 Atlantic
003 Bergen
005 Burlington
007 Camden
009 Cape May
011 Cumberland
013 Essex
015 Gloucester
017 Hudson
019 Hunterdon
021 Mercer
023 Middlesex
025 Monmouth
027 Morris
029 Ocean
031 Passaic
033 Salem
035 Somerset
037 Sussex
039 Union
041 Warren

---- ----
35 NEW MEXICO (NM)
---- ----
001 Bernalillo
003 Catron
005 Chaves
006 Cibola
007 Colfax
009 Curry
011 DeBaca
013 Dona Ana
015 Eddy
017 Grant
019 Guadalupe
021 Harding
023 Hidalgo
025 Lea
027 Lincoln
028 Los Alamos
029 Luna
031 McKinley
033 Mora
035 Otero
037 Quay
039 Rio Arriba
041 Roosevelt
043 Sandoval
045 San Juan
047 San Miguel
049 Santa Fe
051 Sierra
053 Socorro
055 Taos
057 Torrance
059 Union
061 Valencia

---- ----
36 NEW YORK (NY)
---- ----
001 Albany
003 Allegany
005 Bronx
007 Broome
009 Cattarnugus
011 Cayuga
013 Chautauqua
015 Chemung
017 Chenango
019 Clinton
021 Columbia
023 Cortland
025 Delaware
027 Dutchess
029 Erie
031 Essex
033 Franklin
035 Fulton
037 Genesee
039 Greene
041 Hamilton
043 Herkimer
045 Jefferson
047 Kings
049 Lewis
051 Livingston
053 Madison
055 Monroe
057 Montgomery
059 Nassau
061 New York
063 Niagara
065 Oneida
067 Onondaga
069 Ontario
071 Orange
073 Orleans
075 Oswego
077 Otsego
079 Putnam
081 Queens
083 Renselaer
085 Richmond
087 Rockland
089 St. Lawrence
091 Saratoga
093 Schenectady
095 Schoharie
097 Schuyler
099 Seneca
101 Steuben
103 Suffolk
105 Sullivan
107 Tioga
109 Tompkins
111 Ulster
113 Warren
115 Washington
117 Wayne
119 Westchester
121 Wyoming
123 Yates

---- ----
37 NORTH CAROLINA (NC)
---- ----
001 Alamance
003 Alexander
005 Alleghany
007 Anson
009 Ashe
011 Avery
013 Beaufort
015 Bertie
017 Bladen
019 Brunswick
021 Buncombe
023 Burke
025 Cabarrus
027 Caldwell
029 Camden
031 Carteret
033 Caswell
035 Catawba
037 Chatham
039 Cherokee
041 Chowan
043 Clay
045 Cleveland
047 Columbus
049 Craven
051 Cumberland
053 Currituck
055 Dare
057 Davidson
059 Davie
061 Duplin
063 Durham
065 Edgecombe
067 Forsyth
069 Franklin
071 Gaston
073 Gates
075 Graham
077 Granville
079 Greene
081 Guilford
083 Halifax
085 Harnett
087 Haywood
089 Henderson
091 Hertford
093 Hoke
095 Hyde
097 Iredell
099 Jackson
101 Johnston
103 Jones
105 Lee
107 Lenoir
109 Lincoln
111 McDowell
113 Macon
115 Madison
117 Martin
119 Mecklenburg
121 Mitchell
123 Montgomery
125 Moore
127 Nash
129 New Hanover
131 Northampton
133 Onslow
135 Orange
137 Pamlico
139 Pasquotank
141 Pender
143 Perquimans
145 Person
147 Pitt
149 Polk
151 Randolph
153 Richmond
155 Robeson
157 Rockingham
159 Rowan

161 Rutherford
163 Sampson
165 Scotland
167 Staniy
169 Stokes
171 Surry
173 Swain
175 Transylvania
177 Tyrrell
179 Union
181 Vance
183 Wake
185 Warren
187 Washington
189 Watauga
191 Wayne
193 Wilkes
195 Wilson
197 Yadkin
199 Yancey

---- ----
38 NORTH DAKOTA (ND)
---- ----
001 Adams
003 Barnes
005 Benson
007 Billings
009 Bottineau
011 Bowman
013 Burke
015 Burleigh
017 Cass
019 Cavalier
021 Dickey
023 Divide
025 Dunn
027 Eddy
029 Emmons
031 Foster
033 Golden Valley
035 Grand Forks
037 Grant
039 Griggs
041 Hettinger
043 Kidder
045 LaMoure
047 Logan
049 McHenry
051 Mclntosh
053 McKenzie
055 McLean
057 Mercer
059 Morton
061 Mountrail
063 Nelson
065 Oliver
067 Pembina
069 Pierce
071 Ramsey
073 Ransom
075 Renville
077 Richland
079 Rolette
081 Sargent
083 Sheridan
085 Sioux
087 Slope
089 Stark
091 Steele
093 Stutsman
095 Towner
097 Traill
099 Walsh
101 Ward
103 Wells
105 Williams

---- ----
39 OHIO (OH)
---- ----
001 Adams
003 Allen
005 Ashland
007 Ashtabula
009 Athens
011 Auglaize
013 Belmont
015 Brown
017 Butler
019 Carroll
021 Champaign
023 Clark
025 Clermont
027 Clinton
029 Columbiana
031 Coshocton
033 Crawford
035 Cuyahoga
037 Darke
039 Defiance
041 Delaware
043 Erie
045 Fairfield
O47 Fayette
049 Franklin
051 Fulton
053 Gallia
055 Geauga
057 Greene
059 Guernsey
061 Hamilton
063 Hancock
065 Hardin
067 Harrison
069 Henry
071 Highland
073 Hocking
075 Holmes
077 Huron
079 Jackson
081 Jefferson
083 Knox
085 Lake
087 Lawrence
089 Licking
091 Logan
093 Lorain
095 Lucas
097 Madison
099 Mahoning
101 Marion
103 Medina
105 Meigs
107 Mercer
109 Miami
111 Monroe
113 Montgomery
115 Morgan
117 Morrow
119 Muskingum
121 Noble
123 Ottawa
125 Paulding
127 Perry
129 Pickaway
131 Pike
133 Portage
135 Preble
137 Putnam
139 Richland
141 Ross
143 Sandusky
145 Scioto
147 Seneca
149 Shelby
151 Stark
153 Summit
155 Trumbull
157 Tuscarawas
159 Union
161 Van Wert
163 Vinton
165 Warren
167 Washington
169 Wayne
171 Williams
173 Wood
175 Wyandot

---- ----
40 OKLAHOMA (OK)
---- ----
001 Adair
003 Alfalfa
005 Atoka
007 Beaver
009 Beckham
011 Blaine
013 Bryan
015 Caddo
017 Canadian
019 Carter
021 Cherokee
023 Choctaw
025 Cimarron
027 Cleveland
029 Coal
031 Comanche
033 Cotton
035 Craig
037 Creek
039 Custer
041 Delaware
043 Dewey
045 Ellis
047 Garfield
049 Garvin
051 Grady
053 Grant
055 Greer
057 Harmon
059 Harper
061 Haskell
063 Hughes
065 Jackson
067 Jefferson
069 Johnston
071 Kay
073 Kingfisher
075 Kiowa
077 Latimer
079 Le Flore
081 Lincoln
083 Logan
085 Love
087 McClain
089 McCurtain
091 McIntosh
093 Major
095 Marshall
097 Mayes
099 Murray
101 Muskogee
103 Noble
105 Nowata
107 Okfuskee
109 Oklahoma
111 Okmulgee
113 Osage
115 Ottawa
117 Pawnee
119 Payne
121 Pittsburg
123 Pontotoc
125 Pottawatomie
127 Pushmataha
129 Roger Mills
131 Rogers
133 Seminole
135 Sequoyah
137 Stephens
139 Texas
141 Tillman
143 Tulsa
145 Wagoner
147 Washington
149 Washita
151 Woods
153 Woodward

---- ----
41 OREGON (OR)
---- ----
001 Baker
003 Benton
005 Clackamas
007 Clatsop
009 Columbia
011 Coos
013 Crook
015 Curry
017 Deschutes
019 Douglas
021 Gilliam
023 Grant
025 Harney
027 Hood River
029 Jackson
031 Jefferson
033 Josephine
035 Klamath
037 Lake
039 Lane
041 Lincoln
043 Linn
045 Malheur
047 Marion
049 Morrow
051 Multnomah
053 Polk
055 Sherman
057 Tillamook
059 Umatilla
061 Union
063 Wallowa
065 Wasco
067 Washington
069 Wheeler
071 Yamhill

---- ----
42 PENNSYLVANIA (PA)
---- ----
001 Adams
003 Allegheny
005 Armstrong
007 Beaver
009 Bedford
011 Berks
013 Blair
015 Bradford
017 Bucks
019 Butler
021 Cambria
023 Cameron
025 Carbon
027 Centre
029 Chester
031 Clarion
033 Clearfield
035 Clinton
037 Columbia
039 Crawford
041 Cumberland
043 Dauphin
045 Delaware
047 Elk
049 Erie
051 Fayette
053 Forest
055 Franklin
057 Fulton
059 Greene
061 Huntingdon
063 Indiana
065 Jefferson
067 Juniata
069 Lackawanna
071 Lancaster
073 Lawrence
075 Lebanon
077 Lehigh
079 Luzerne
081 Lycoming
083 McKean
085 Mercer
087 Mifflin
089 Monroe
091 Montgomery
093 Montour
095 Northampton
097 Northumberland
099 Perry
101 Philadelphia
103 Pike
105 Potter
107 Schuylkill
109 Snyder
111 Somerset
113 Sullivan
115 Susquehanna
117 Tioga

119 Union
121 Venango
123 Warren
125 Washington
127 Wayne
129 Westmornland
131 Wyoming
133 York

---- ----
44 RHODE ISLAND (RI)
---- ----
001 Bristol
003 Kent
005 Newport
007 Providence
009 Washington

---- ----
45 SOUTH CAROLINA (SC)
---- ----
001 Abbeville
003 Aiken
005 Allendale
007 Anderson
009 Bamberg
011 Barnwell
013 Beaufort
015 Berkeley
017 Calhoun
019 Charleston
021 Cherokee
023 Chester
025 Chesterfield
027 Clarendon
029 Colleton
031 Darlington
033 Dillon
035 Dorchester
037 Edgefield
039 Fairfield
041 Florence
043 Georgetown
045 Greenville
047 Greenwood
049 Hampton
051 Horry
053 Jasper
055 Kershaw
057 Lancaster
059 Laurens
061 Lee
063 Lexington
065 McCormick
067 Marion
069 Marlboro
071 Newberry
073 Oconee
075 Orangeburg
077 Pickens
079 Richland
081 Saluda
083 Spartanburg
085 Sumter
087 Union
089 Williamsburg
091 York

---- ----
46 SOUTH DAKOTA (SD)
---- ----
003 Aurora
005 Beadle
007 Bennett
009 Bon Homme
011 Brookings
013 Brown
015 Brule
017 Buffalo
019 Butte
021 Campbell
023 Charles Mix
025 Clark
027 Clay
029 Codington
031 Corson
033 Custer
035 Davison
037 Day
039 Deuel
041 Dewey
043 Douglas
045 Edmunds
047 Fall River
049 Faulk
051 Grant
053 Gregory
055 Haakon
057 Hamlin
059 Hand
061 Hanson
063 Harding
065 Hughes
067 Hutchinson
069 Hyde
071 Jackson
073 Jerauld
075 Jones
077 Kingsbury
079 Lake
081 Lawrence
083 Lincoln
085 Lyman
087 McCook
089 McPherson
091 Marshall
093 Meade
095 Mellette
097 Miner
099 Minnehaha
101 Moody
103 Pennington
105 Perkins
107 Potter
109 Roberts
111 Sanborn
113 Shannon
115 Spink
117 Stanley
119 Sully
121 Todd
123 Tripp
125 Turner
127 Union
129 Walworth
135 Yankton
137 Ziebach

---- ----
47 TENNESSEE (TN)
---- ----
001 Anderson
003 Bedford
005 Benton
007 Bledsoe
009 Blount
011 Bradley
013 Campbell
015 Cannon
017 Carroll
019 Carter
021 Cheatham
023 Chester
025 Claiborne
027 Clay
029 Cocke
031 Coffee
033 Crockett
035 Cumberland
037 Davidson
039 Decatur
041 DeKalb
043 Dickson
045 Dyer
047 Fayette
049 Fentress
051 Franklin
053 Gibson
055 Giles
057 Grainger
059 Greene
061 Grundy
063 Hamblen
065 Hamilton
067 Hancock
069 Hardeman
071 Hardin
073 Hawkins
075 Haywood
077 Henderson
079 Henry
081 Hickman
083 Houston
085 Humphreys
087 Jackson
089 Jefferson
091 Johnson
093 Knox
095 Lake
097 Lauderdale
099 Lawrence
101 Lewis
103 Lincoln
105 Loudon
107 McMinn
109 McNairy
111 Macon
113 Madison
115 Marion
117 Marshall
119 Maury
121 Meigs
123 Monroe
125 Montgomery
127 Moore
129 Morgan
131 Obion
133 Overton
135 Perry
137 Pickett
139 Polk
141 Putnam
143 Rhea
145 Roane
147 Robertson
149 Rutherford
151 Scott
153 Sequatchie
155 Sevier
157 Shelby
159 Smith
161 Stewart
163 Sullivan
165 Sumner
167 Tipton
169 Trousdale
171 Unicoi
173 Union
175 Van Buren
177 Warren
179 Washington
181 Wayne
183 Weakley
185 White
187 Williamson
189 Wilson

---- ----
48 TEXAS (TX)
---- ----
001 Anderson
003 Andrews
005 Angelina
007 Aransas
009 Archer
011 Armstrong
013 Atascosa
015 Austin
017 Bailey
019 Bandera
021 Bastrop
023 Baylor
025 Bee
027 Bell
029 Bexar
031 Blanco
033 Borden
035 Bosque
037 Bowie
039 Brazoria
041 Brazos
043 Brewster
045 Briscoe
047 Brooks
049 Brown
051 Burleson
053 Burnet
055 Caldwell
057 Calhoun
059 Callahan
061 Cameron
063 Camp
065 Carson
067 Cass
069 Castro
071 Chambers
073 Cherokee
075 Childress
077 Clay
079 Cochran
081 Coke
083 Coleman
085 Collin
087 Collingsworth
089 Colorado
091 Comal
093 Comanche
095 Concho
097 Cooke
099 Coryell
101 Cottle
103 Crane
105 Crockett
107 Crosby
109 Culberson
111 Dallam
113 Dallas
115 Dawson
117 Deaf Smith
119 Delta
121 Denton
123 DeWitt
125 Dickens
127 Dimmit
129 Donley
131 Duval
133 Eastland
135 Ector
137 Edwards
139 Ellis
141 El Paso
143 Erath
145 Falls
147 Fannin
149 Fayette
151 Fisher
153 Floyd
155 Foard
157 Fort Bend
159 Franklin
161 Freestone
163 Frio
165 Gaines
167 Galveston
169 Garza
171 Gillespie
173 Glasscock
175 Goliad
177 Gonzales
179 Gray
181 Grayson
183 Gregg
185 Grimes
187 Guadalupe
189 Hale
191 Hall
193 Hamilton
195 Hansford
197 Hardeman
199 Hardin
201 Harris
203 Harrison
205 Hartley
207 Haskell
209 Hays
211 Hemphill
213 Henderson
215 Hidalgo
217 Hill
219 Hockley
221 Hood
223 Hopkins
225 Houston
227 Howard
229 Hudspeth
231 Hunt

233 Hutchinson
235 Irion
237 Jack
239 Jackson
241 Jasper
243 Jeff Davis
245 Jefferson
247 Jim Hogg
249 Jim Wells
251 Johnson
253 Jones
255 Karnes
257 Kaufman
259 Kendall
261 Kenedy
263 Kent
265 Kerr
267 Kimble
269 King
271 Kinney
273 Kleberg
275 Knox
277 Lamar
279 Lamb
281 Lampasas
283 La Salle
285 Lavaca
287 Lee
289 Leon
291 Liberty
293 Limestone
295 Lipscomb
297 Live Oak
299 Llano
301 Loving
303 Lubbock
305 Lynn
307 McCulloch
309 McLennan
311 McMullen
313 Madison
315 Marion
317 Martin
319 Mason
321 Matagorda
323 Maverick
325 Medina
327 Menard
329 Midland
331 Milam
333 Mills
335 Mitchell
337 Montague
339 Montgomery
341 Moore
343 Morris
345 Motley
347 Nacogdoches
349 Navarro
351 Newton
353 Nolan
355 Nueces
357 Ochiltree
359 Oldham
361 Orange
363 Palo Pinto
365 Panola
367 Parker
369 Parmer
371 Pecos
373 Polk
375 Potter
377 Presidio
379 Rains
381 Randall
383 Reagan
385 Real
387 Red River
389 Reeves
391 Refugio
393 Roberts
395 Robertson
397 Rockwall
399 Runnels
401 Rusk
403 Sabine
405 San Augustine
407 San Jacinto
409 San Patricio
411 San Saba
413 Schleicher
415 Scurry
417 Shackelford
419 Shelby
421 Sherman
423 Smith
425 Somervell
427 Starr
429 Stephens
431 Sterling
433 Stonewall
435 Sutton
437 Swisher
439 Tarrant
441 Taylor
443 Terrell
445 Terry
447 Throckmorton
449 Titus
451 Tom Green
453 Travis
455 Trinity
457 Tyler
459 Upshur
461 Upton
463 Uvalde
465 Val Verde
467 Van Zandt
469 Victoria
471 Walker
473 Waller
475 Ward
477 Washington
479 Webb
481 Wharton
483 Wheeler
485 Wichita
487 Wilbarger
489 Willacy
491 Williamson
493 Wilson
495 Winkler
497 Wise
499 Wood
501 Yoakum
503 Young
505 Zapata
507 Zavala

---- ----
49 UTAH (UT)
---- ----
001 Beaver
003 Box Elder
005 Cache
007 Carbon
009 Daggett
011 Davis
013 Duchesne
015 Emery
017 Garfield
019 Grand
021 Iron
023 Juab
025 Kane
027 Millard
029 Morgan
031 Piute
035 Salt Lake
037 San Juan
039 Sanpete
041 Sevier
043 Summit
045 Tooele
047 Uintah
049 Utah
051 Wasatch
053 Washington
055 Wayne
057 Weber

---- ----
50 VERMONT (VT)
---- ----
001 Addison
003 Bennington
005 Caledonia
007 Chittenden
009 Essex
011 Franklin
013 Grand Isle
015 Lamoille
017 Orange
019 Orleans
021 Rutland
023 Washington
025 Windham
027 Windsor

---- ----
51 VIRGINIA (VA)
---- ----
001 Accomack
003 Albemarle
005 Alleghany
007 Amelia
009 Amherst
011 Appomattox
013 Arlington
015 Augusta
017 Bath
019 Bedford
021 Bland
023 Botetourt
025 Brunswick
027 Buchanan
029 Buckingham
031 Campbell
033 Caroline
035 Carroll
036 Charles City
037 Charlotte
041 Chesterfield
043 Clarke
045 Craig
047 Culpeper
049 Cumberland
051 Dickenson
053 Dinwiddie
057 Essex
059 Fairfax
061 Fauquier
063 Floyd
065 Fluvanna
067 Franklin
069 Frederick
071 Giles
073 Gloucester
075 Goochland
077 Grayson
079 Greene
081 Greensville
083 Halifax
085 Hanover
087 Henrico
089 Henry
091 Highland
093 Isle of Wight
095 James City
097 King and Queen
099 King George
101 King William
103 Lancaster
105 Lee
107 Loudoun
109 Louisa
111 Lunenburg
113 Madison
115 Mathews
117 Mecklenburg
119 Middlesex
121 Montgomery
125 Nelson
127 New Kent
131 Northampton
133 Northumberland
135 Nottoway
137 Orange
139 Page
141 Patrick
143 Pittsylvania
145 Powhatan
147 Prince Edward
149 Prince George
153 Prince William
155 Pulaski
157 Rappahannock
159 Richmond
161 Roanoke
163 Rockbridge
165 Rockingham
167 Russell
169 Scott
171 Shenandoah
173 Smyth
175 Southampton
177 Spotsylvania
179 Stafford
181 Surry
183 Sussex
185 Tazewell
187 Warren
191 Washington
193 Westmoreland
195 Wise
197 Wythe
199 York
510 Alexandria (city)
515 Bedford (city)
520 Bristol (city)
530 Buena Vista (city)
540 Charlottesville (city)
550 Chesapeake (city)
570 Colonial Heights (city)
580 Covington (city)
590 Danville (city)
595 Emporia (city)
600 Fairfax (city)
610 Falls Church (city)
620 Franklin (city)
630 Fredericksburg (city)
640 Galax (city)
650 Hampton (city)
660 Harrisonburg (city)
670 Hopewell (city)
678 Lexington (city)
680 Lynchburg (city)
683 Manassas (city)
685 Manassas Park (city)
690 Martinsville (city)
700 Newport News (city)
710 Norfolk (city)
720 Norton (city)
730 Petersburg (city)
735 Poquoson (city)
740 Portsmouth (city)
750 Radford (city)
760 Richmond (city)
770 Roanoke (city)
775 Salem (city)
790 Staunton (city)
800 Suffolk (city)
810 Virginia Beach (city)
820 Waynesboro (city)
830 Williamsburg (city)
840 Winchester (city)

---- ----
53 WASHINGTO

N (WA)
---- ----
001 Adams
003 Asotin
005 Benton
007 Chelan
009 Clallam
011 Clark
013 Columbia
015 Cowlitz
017 Douglas
019 Ferry
021 Franklin
023 Garfield
025 Grant
027 Grays Harbor
029 Island
031 Jefferson
033 King
035 Kitsap
037 Kittitas
039 Klickitat
041 Lewis
043 Lincoln
045 Mason
047 Okanogan
049 Pacific
051 Pend Oreille
053 Pierce
055 San Juan
057 Skagit
059 Skamania
061 Snohomish
063 Spokane
065 Stevens
067 Thurston
069 Wahkiakum
071 Walla Walla
073 Whatcom
075 Whitman
077 Yakima

---- ----
54 WEST VIRGINIA (WV)
---- ----
001 Barbour
003 Berkeley
005 Boone
007 Braxton
009 Brooke
011 Cabell
013 Calhoun
015 Clay
017 Doddridge
019 Fayette
021 Gilmer
023 Grant
025 Greenbrier
027 Hampshire
029 Hancock
031 Hardy
033 Harrison
035 Jackson
037 Jefferson
039 Kanawha
041 Lewis
043 Lincoln
045 Logan
047 McDowell
049 Marion
051 Marshall
053 Mason
055 Mercer
057 Mineral
059 Mingo
061 Monongalia
063 Monroe
065 Morgan
067 Nicholas
069 Ohio
071 Pendleton
073 Pleasants
075 Pocahontas
077 Preston
079 Putnam
081 Raleigh
083 Randolph
085 Ritchie
087 Roane
089 Summers
091 Taylor
093 Tucker
095 Tyler
097 Upshur
099 Wayne
101 Webster
103 Wetzel
105 Wirt
107 Wood
109 Wyoming

---- ----
55 WISCONSIN (WI)
---- ----
001 Adams
003 Ashland
005 Barron
007 Bayfield
009 Brown
011 Buffalo
013 Burnett
015 Calumet
017 Chippewa
019 Clark
021 Columbia
023 Crawford
025 Dane
027 Dodge
029 Door
031 Douglas
033 Dunn
035 Eau Claire
037 Florence
039 Fond du Lac
041 Forest
043 Grant
045 Green
047 Green Lake
049 Iowa
051 Iron
053 Jackson
055 Jefferson
057 Juneau
059 Kenosha
061 Kewaunee
063 La Crosse
065 Lafayette
067 Langlade
069 Lincoln
071 Manitowoc
073 Marathon
075 Marinette
077 Marquette
078 Menominee
079 Milwaukee
081 Monroe
083 Oconto
085 Oneida
087 Outagamie
089 Ozaukee
091 Pepin
093 Pierce
095 Polk
097 Portage
099 Price
101 Racine
103 Richland
105 Rock
107 Rusk
109 St. Croix
111 Sauk
113 Sawyer
115 Shawano
117 Sheboygan
119 Taylor
121 Trempealeau
123 Vernon
125 Vilas
127 Walworth
129 Washburn
131 Washington
133 Waukesha
135 Waupaca
137 Waushara
139 Winnebago
141 Wood

---- ----
56 WYOMING (WY)
---- ----
001 Albany
003 Big Horn
005 Campbell
007 Carbon
009 Converse
011 Crook
013 Fremont
015 Goshen
017 Hot Springs
019 Johnson
021 Laramie
023 Lincoln
025 Natrona
027 Niobrara
029 Park
031 Platte
033 Sheridan
035 Sublette
037 Sweetwater
039 Teton
041 Uinta
043 Washakie
045 Weston

---- ----
60 AMERICA SAMOA (AS)
---- ----
010 Eastern (District)
020 Manu'a (District)
030 Rose Island
040 Swains Island
050 Western (District)

---- ----
64 FEDERATED STATES OF MICRONESIA (FM)
---- ----
002 Chuuk
005 Kosrae
040 Pohnpeit
060 Yap

---- ----
66 GUAM (GU)
---- ----
010 Guam

---- ----
68 MARSHALL ISLANDS (MH)
---- ----
007 Ailinginae
010 Ailinglaplap
030 Ailuk
040 Arno
050 Aur
060 Bikar
070 Bikini
073 Bokak
080 Ebon
090 Enewetak
100 Erikub
110 Jabat
120 Jaluit
130 Jemo
140 Kili
150 Kwajalein
160 Lae
170 Lib
180 Likiep
190 Majuro
300 Maloelap
310 Mejit
320 Mili
330 Namorik
340 Namu
350 Rongelap
360 Rongrik
385 Toke
390 Ujae
400 Ujelang
410 Utrik
420 Wotho
430 Wotle

---- ----
69 NORTHERN MARIANA ISLANDS (MP)
---- ----
085 Northern Islands
100 Rota
110 Saipan
120 Tinian

---- ----
70 PALAU (PW)
---- ----
002 Aimeliik
004 Airai
010 Angaur
050 Hatobohei
100 Kayangel
150 Koror
212 Melekeok
214 Ngaraard
218 Ngarchelong
222 Ngardmau
224 Ngatpang
226 Ngchesar
227 Ngeremlengui
228 Ngiwal
350 Peleliu
370 Sonsorol

---- ----
72 PUERTO RICO (PR)
---- ----
001 Adjuntas
003 Aguada
005 Aguadilla
007 Aguas Buenas
009 Aibonito
011 Anasco
013 Arecibo
015 Arroyo
017 Barceloneta
019 Barranquitas
021 Bayamo'n
023 Cabo Rojo
025 Caguas
027 Camuy
029 Canovanas
031 Carolina
033 Catano
035 Cayey
037 Ceiba
039 Ciales
041 Cidra
043 Coamo
045 Comerio
047 Corozal
049 Culebra
051 Dorado
053 Fajardo
054 Florida
055 Guanica
057 Guayama
059 Guayanilla
061 Guaynabo
063 Gurabo
065 Hatillo
067 Hormigueros
069 Humacao
071 Isabela
073 Jayuya
075 Juana Diaz
077 Juncos
079 Lajas
081 Lares
083 Las Marias
085 Las Piedras
087 Loiza
089 Luquillo
091 Manati
093 Maricao
095 Maunabo
097 Mayaguez
099 Moca
101 Morovis

103 Naguabo
105 Naranjito
107 Orocovis
109 Patillas
111 Penuelas
113 Ponce
115 Quebradillas
117 Rincon
119 Rio Grande
121 Sabana
Grande
123 Salinas
125 San German
127 San Juan
129 San Lorenzo
131 San Sebastian
133 Santa Isabel
135 Toa Alta
137 Toa Baja
139 Trujillo Alto
141 Utuado
143 Vega Alta
145 Vega Baja
147 Vieques
149 Villalba
151 Yabucoa
153 Yauco

---- ----
74 U.S. MINOR
OUTLYING
ISLANDS (UM)
---- ----
050 Baker Island
100 Howland
Island
150 Jarvis Island
200 Johnston
Island
250 Kingman Reef
300 Midway
Islands
350 Navassa
Island
400 Palmyra Atoll
450 Wake Island

---- ----
78 VIRGIN
ISLANDS OF THE
UNITED STATES
(VI)
---- ----
010 St. Croix
020 St. John
030 St. Thomas

Doc. A/65B Amendment 1
28 April 2005

Amendment No. 1 to ATSC Standard: Program and System Information Protocol for Terrestrial Broadcast and Cable (Doc. A/65B)

Advanced Television Systems Committee
1750 K Street, N.W.
Suite 1200
Washington, D.C. 20006
www.atsc.org

The Advanced Television Systems Committee, Inc., is an international, non-profit organization developing voluntary standards for digital television. The ATSC member organizations represent the broadcast, broadcast equipment, motion picture, consumer electronics, computer, cable, satellite, and semiconductor industries.

Specifically, ATSC is working to coordinate television standards among different communications media focusing on digital television, interactive systems, and broadband multimedia communications. ATSC is also developing digital television implementation strategies and presenting educational seminars on the ATSC standards.

ATSC was formed in 1982 by the member organizations of the Joint Committee on InterSociety Coordination (JCIC): the Electronic Industries Association (EIA), the Institute of Electrical and Electronic Engineers (IEEE), the National Association of Broadcasters (NAB), the National Cable Television Association (NCTA), and the Society of Motion Picture and Television Engineers (SMPTE). Currently, there are approximately 140 members representing the broadcast, broadcast equipment, motion picture, consumer electronics, computer, cable, satellite, and semiconductor industries.

ATSC Digital TV Standards include digital high definition television (HDTV), standard definition television (SDTV), data broadcasting, multichannel surround-sound audio, and satellite direct-to-home broadcasting.

Amendment No. 1 to ATSC A/65B

This document contains proposed changes to A/65B to add PSIP support for delivery of data per Amendment 1 to A/53C, Annex D (Enhanced VSB). Change instructions are given in *italics*. New text is shown in blue underline. Deleted text is shown in ~~red strikethrough~~.

1. Section 3.2, Add New Acronyms

Add the following acronyms into Section 3.2:

EIT-E Event Information Table in the TS-E

ETT-E Event Text Table in the TS-E

MGT-E Master Guide Table in the TS-E

PSIP-E Program and System Information in the TS-E

STT-E System Time Table in the TS-E

TVCT-E Terrestrial Virtual Channel Table in the TS-E

TS-E The collection of packets delivered by enhanced VSB

TS-Ea The collection of packets delivered using one-half rate coding per Amendment 1 to A/53C (Annex D)

TS-Eb The collection of packets delivered using one-quarter rate coding per Amendment 1 to A/53C (Annex D)

2. Revise Text in Section 4.3 to Allow Certain Use of a New PID to Identify Certain Packets

Revise text in Section 4.3 to read (note new footnote):

Certain fields in this Standard are defined to include "user private" ranges:

- table_ID values in the range 0x40 through 0xBF
- MGT table_type values in the range 0x0400 through 0x0FFF

Table sections with table_ID values in the user private range (0x40 through 0xBF) shall not appear in transport packets identified with the base_PID PID value (0x1FFB), the base_PID-E PID value (0x1FF9)[1], or the STT_PID_E value (0x1FF8). The MGT and the MGT-E may refer to private table sections with any value of table_ID including values in the user private range. Refer to Section 6.2 for a discussion of the use of the MPEG-2 Registration Descriptor in the MGT. Refer to Section 6.11.1 for discussion and constraints that apply to the MGT-E.

3. Add Requirements for E-VSB Support:

1. Add the following paragraph below Requirement 4 in Section 5.1 (note new footnote):

Requirement 4E: When there is a service of service_type 0x02, service_type 0x03, or service_type 0x06 in the digital Transport Stream delivered via E-VSB (per Amendment No. 1 to A/53C,

[1] Other restrictions on use of PID values exist in [1] (A/53).

Annex D) the transmission shall include the STT-E, TVCT-E, the MGT-E, and should include the first Event Information Table (EIT-0-E)[2].

2. Add the following paragraph below Requirement 5 in Section 5:

Requirement 5E: The PSIP-E tables shall describe all services of service_type 0x02, service_type 0x03, or service_type 0x06 delivered via E-VSB (per Amendment No. 1 to A/53C, Annex D).

4. Section 2, Update References

Update the reference to A/53 and add certain sections to the normatively referenced sections (the expected revised version of A/53) to make normative TS, TS-E, and the reference receiver model section.

[1] ATSC Standard A/53C as modified by Amendments No. 1, 2 and 3 (2004), "ATSC Digital Television Standard, Revision C"; Annex C, Section 3, Section 5.7.5, Section 6.8.5, and Section 8 (*normative*).

5. Edit Semantics for minor_channel_number in Section 6.3.1

Edit as shown below:

minor_channel_number — A 10-bit number in the range 0 to 999 that represents the "minor" or "sub-" channel number. This field, together with major_channel_number, performs as a two-part channel number, where minor_channel_number represents the second or right-hand part of the number. When the service_type is analog television, minor_channel_number shall be set to 0. Services whose service_type is either ATSC_digital_television, ATSC_audio_only, or unassociated/small_screen_service shall use minor numbers between 1 and 99. For other types of services, such as data broadcasting, valid minor virtual channel numbers are between 1 and 999.

6. Define service_type value for Unassociated/Small Screen Service

1) *Add footnote to first column* (service_type field) *of Table 6.7:*

> "[1] Other values may be defined in other ATSC standards; refer to the ATSC Code Point Registry."

2) *Define* service_type *value 0x06 and add to Table 6.7 the following row:*

unassociated/small_screen_service – The virtual channel in the TVCT-E carries digital television program (audio, video and optional associated data) targeting receiving devices with smaller displays, as defined in Section 6.11.3.

3) *Delete the last row of Table 6.7, "0x05-0x3F [Reserved for future ATSC Use]."*

7. Add New Section 6.11

Add new Section 6.11 and its sub sections as shown below:

[2] Exception: information about test signals may or may not be included in EIT-E/ETT-E data.

6.11 Enhanced Service PSIP Signaling and Announcement (PSIP-E)

This section describes the features of PSIP that shall be used when the optional E-VSB delivery means are used.

When there are services of service_type 0x02 or of service_type 0x03 in TS-E (per Annex C of [1]) the TS-E shall include the table sections that make up STT-E, TVCT-E, the MGT-E, and should include the first Event Information Table (EIT-0-E). When EIT-Es are sent they should contain valid information about the events.

Table sections of PSIP-E shall be carried in TS-E packets. Table sections that make up PSIP-E packets shall be present in TS-Eb[3] for all services in TS-Eb.

When a service element is in TS-Eb, all PSIP-E table sections that reference that element shall be in TS-Eb. Furthermore, the MGT-E and the TVCT-E shall be in TS-Eb if any service element is in TS-Eb.

Unless otherwise specified herein, placement and contents of each descriptor in PSIP-E table sections shall conform to usage defined for that descriptor in PSIP table sections. See Section 6.9.

6.11.1 Enhanced Service MGT (MGT-E)

The MGT-E shall use the MGT table_ID and shall meet all the mandatory specifications that apply to the MGT, except as explicitly stated in this section.

When services of service_type 0x02 or service_type 0x03 are sent using the TS-E (defined in Annex C of [1]), an Enhanced Master Guide Table (MGT-E) shall be present. The PID for the table sections that make up the MGT-E shall have the value 0x1FF9 (base_PID_E). The MGT-E shall only refer to tables that are in TS-E (per [1]).

6.11.2 System Time Table for Enhanced Modes (STT-E)

The STT-E shall meet all the mandatory specifications of the STT (Section 6.1), except as explicitly stated in this section.

The PID for STT-E shall have the value 0x1FF8 (STT_PID_E).

The system_time semantics when this field is in an STT-E are:

system_time — This field shall have the same value and accuracy as system_time in the STT except it is timed at the arrival at the output of the Reference Receiver of the Transport Stream-E packet carrying the last byte of the CRC (note the Reference Receiver has zero processing time). The STT-E system_time should be set to the next second and sent approximately 2T milliseconds before the seconds count is due to increment, where T represents the average number of milliseconds between TS-E packets identified with STT_PID_E (0x1FF8). If one or more translators and/or repeaters are in the RF delivery path that introduce processing delays that impact the overall STT-E timing accuracy, the STT-E timing should be adjusted in the translated/repeated signal.

When sent, the STT-E shall be sent with this field set in precise synchronization to the value of the system_time in the STT, but the STT-E may be sent as seldom as once a minute. Note that if the interval is increased, accuracy of random access to events may be decreased.

[3] The most robust mode.

6.11.3 Virtual Channel Table for Enhanced Modes (TVCT-E)

The TVCT-E shall meet all the mandatory specifications that apply to the TVCT, except as explicitly stated in this section.

When present, the TVCT-E shall be transmitted at a rate of no less than once per minute. The TVCT-E shall include virtual channels in which some or all program elements are transmitted in TS-Ea or TS-Eb packets, and only those (it shall not include virtual channels composed only of program elements transmitted using the main 8-VSB mode).

The Service Location Descriptor in TVCT-E in the TS-Ea shall list only program elements delivered in the TS-Ea.

The Service Location Descriptor in TVCT-E in the TS-Eb shall list only program elements delivered in the TS-Ea or TS-Eb.

The service_type field value 0x06 shall be used in the TVCT-E if and only if the programming identified by the PSIP-E table sections is different from the programming identified by the PSIP table sections.

For virtual channels that are based on the same video and/or audio content as that carried in TS-M, or that identify components that are also Linked components per [1], Section 5.7.5, the contents of the fields shall have the following constraints:

major_channel_number — The value shall be the same as the corresponding 10-bit number in the TVCT.

minor_channel_number — The value shall be the same as the corresponding 10-bit number in the TVCT.

carrier_frequency — These 32 bits shall be set to zero.

program_number — The value shall be the same as the corresponding 16-bit number in the TVCT.

source_id — The value shall be the same as the corresponding 16-bit number in the TVCT.

6.11.4 Event Information Table for Enhanced Modes (EIT-E)

The first Event Information Table (EIT-0-E) should be transmitted. All instances of EIT-E should contain content describing the programming.

For events associated with a virtual channel that has a service type 0x02 or 0x03, when sent, EIT-E table sections shall have the same contents as the corresponding EIT table sections.

For events associated with a virtual channel that has a service type 0x06, EIT-E table sections may exist.

6.11.5 Event Text Table for Enhanced Modes (ETT-E)

ETT-Es may be present in the TS-E. The text in an ETT-E may be an abbreviated version of corresponding text for a Program described in an ETT.

6.11.6 Directed Channel Change for Enhanced Modes

This standard does not define how to use directed channel change in or with the TS-E.

---

8. Section 7.1, New Timings for PSIP-E Tables

1. Add the following text and new Table 7.3 after Table 7.2 as shown below:

Table 7.3 lists the maximum cycle time for those PSIP-E tables that have such.

Table 7.3 Maximum Cycle Time for the STT-E, MGT-E, VCT-E, and RRT-E

Table	STT-E	MGT-E	TVCT-E	EIT-0-E	RRT-E
Cycle time (ms)	60,000	10,000	30,000	60,000	43,200,000

2. Add the following text and new Table 7.3 after Table 7.2 as shown below:

Table 7.4 lists the maximum transmission rate for PSIP-E packet streams according to their PIDs.

Table 7.4 Maximum Rate for Each PSIP Packet Stream

PID	base_PID_E	EIT-E_PID	ETT-E_PID
Rate (bps)	250,000	250,000	250,000

3. Revise the current paragraph below Table 7.2 in Section 7.1 to be after Table 7.4 and to read:

For terrestrial broadcast applications the following constraints apply:

- In terrestrial broadcast applications, the PSIP elementary streams identified by Transport Stream packets with PID 0x1FFB (base_PID), PID 0x1FF9 (base_PID_E), EIT PIDs, EIT-E PIDs, ~~and~~ETT PIDs, and ETT-E PIDs shall adhere to an STD model with the following parameters:
- sb_leak_rate shall be 625 (indicating a leak rate of 250,000 bps)
- sb_size shall be 1024 (indicating a smoothing buffer size of 1024 bytes)

– End of Document –