Doc. A/65
23 Dec 1997

PROGRAM AND SYSTEM INFORMATION PROTOCOL FOR TERRESTRIAL BROADCAST AND CABLE

ADVANCED TELEVISION SYSTEMS COMMITTEE

Blank Page

PROGRAM AND SYSTEM INFORMATION PROTOCOL FOR TERRESTRIAL BROADCAST AND CABLE

ATSC STANDARD

Table of Contents

PROGRAM AND SYSTEM INFORMATION PROTOCOL FOR TERRESTRIAL BROADCAST AND CABLE

ATSC STANDARD

1. SCOPE

1.1 Purpose

This document 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 will be referred to as **Program and System Information Protocol (PSIP).** Prior to being approved as an ATSC Standard, this document was designated T3/S8-193 and later, after approval by T3, as Doc. T3-442.

This standard was prepared by the Advanced Television Systems Committee (ATSC) Technology Group on Distribution (T3). The document was approved by T3 on 22 October 1997 for submission by letter ballot to the membership of the full ATSC. The document was approved by the members of the ATSC on 23 December 1997.

For an informative description of the purpose, concepts, and tables defined in this protocol, first time readers are encouraged to start with Annex D.

1.2 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.

1.2.1 Terrestrial Broadcast

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

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.

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

1.2.2 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.
- The System Time Table (STT), defining the current date and time of day.

1.3 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.
- **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.

2. REFERENCES

The following documents are applicable to this Standard:

1. ATSC Standard A/52 (1995), Digital Audio Compression (AC-3) (*normative*).
2. ATSC Standard A/53 (1995), ATSC Digital Television Standard (*normative*).
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*).
6. ISO 639, Code for the Representation of Names of Languages, 1988 (*informative*).
7. ISO CD 639.2, Code for the Representation of Names of Languages: alpha-3 code, Committee Draft, dated December 1994 (*normative*).
8. ISO/IEC 10646-1:1993, Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane (*normative*).
9. ISO/IEC 8859**,** Information Processing — 8-bit Single-Octet Coded Character Sets, Parts 1 through 10 (*normative*).
10. 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*).
11. ITU-T Rec. H.262 | ISO/IEC 13818-2:1996, Information Technology — Generic coding of moving pictures and associated audio — Part 2: Video (*normative*).
12. Digital Video Transmission Standard for Cable Television, SCTE DVS-031, Rev. 2, 29 May 1997 (*informative*).
13. EIA 708 *Specification for Advanced Television Closed Captioning (ATVCC),* Electronic Industry Association.
14. EIA 752 *Specification for Transport of Transmission Signal Identifier (TSID) Using Extended Data Service),* Electronic Industry Association.

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
CAT Conditional Access Table
CRC Cyclic Redundancy Check
CVCT Cable Virtual Channel Table
DTV Digital Television
EPG Electronic Program Guide
EIT Event Information Table
ETM Extended Text Message
ETT Extended Text Table
GA Grand Alliance
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
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

UTC Coordinated Universal Time[1]
uimsbf unsigned integer, most significant bit first
VCT Virtual Channel Table. Used in reference to either TVCT or CVCT.
unicode Unicode™

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 6.16). 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*.

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.

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

[1] 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.

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: PSIP is a collection of tables describing virtual channel attributes, event features, and others. PSIP tables are compliant with the private section syntax of ISO/IEC 13818-1.

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

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. It also lists valid table_id and PID values for every table that belongs to PSIP.

4.1 Table Format

Tables defined in this Standard are structured in the same manner used for carrying *ISO/IEC 13818-1* -defined PSI tables, shown in Table 4.1. The structure conforms to the generic private section syntax defined in ISO/IEC 13818-1

Table 4.1 Table format used in PSIP

	Bits	Format
typical_PSI_table( ){		
table_id	8	uimsbf
section_syntax_indicator	1	'1'
private_indicator	1	'0'
zero	2	'00'
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
actual_table_data	*	
CRC_32	32	rpchof
}		

4.2 Table ID Ranges and Values

Table 4.2 defines Table ID ranges and values.

Table 4.2 ID Ranges and Values

Table ID Value (hex)	Tables	PID	Ref.
	ISO/IEC 13818-1 Sections:		
0x00	PROGRAM ASSOCIATION TABLE (PAT)	0	Ref. [10]
0x01	CONDITIONAL ACCESS TABLE (CAT)	1	Ref. [10]
0x02	TS PROGRAM MAP TABLE (PMT)	per PAT	Ref. [10]
0x03-0x3F	[ISO Reserved]		
	User Private Sections:		
0x40-0x7F	[User Private for other systems]		
0x80-0xBF	[User Private]		
	Other documents:		
0xC0-0xC6	[Used in other systems]		
	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-0xDF	[Reserved for future ATSC use]		
0xE0-0xE5	[Used in other systems]		
0xE6-0xFE	[Reserved for future ATSC use]		
0xFF	**Inter-message Filler**		

Tables defined in this PSIP Standard, and any created as user extensions to it are considered "private" with respect to *ISO/IEC 13818-1*. Table types 0x40 through 0xBF are user defined (outside the scope of this PSIP Standard).

4.3 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:

1. **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 below.
2. **Standard Table Types:** As indicated in Table 4.1, 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 that builds upon this one.

3. **User Private Table Types:** As indicated in Table 4.1, table_id values in the range 0x40 through 0xBF shall be reserved for "user private" use.
4. **User Private Descriptors:** Privately defined descriptors may be placed at designated locations throughout the tables described in this Standard. Ownership of one or more user private descriptors may be indicated by the presence of an MPEG registration_descriptor() preceding the descriptor(s).
5. **Protocol Version Field:** Initially this field is set to 0, but after approval, future structural modifications shall be accommodated by defining different protocol version numbers.

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 they are defined and supported.

user_private — Indicates that the bit or bit field is not defined within the scope of this Standard. The owner of the bit, and hence the entity defining its meaning, is derived via its context within a message.

zero — Indicates that the bit or bit field shall have the value zero.

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), the Master Guide Table (MGT), the Rating Region Table (RRT), and the Virtual Channel Table (VCT). 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.

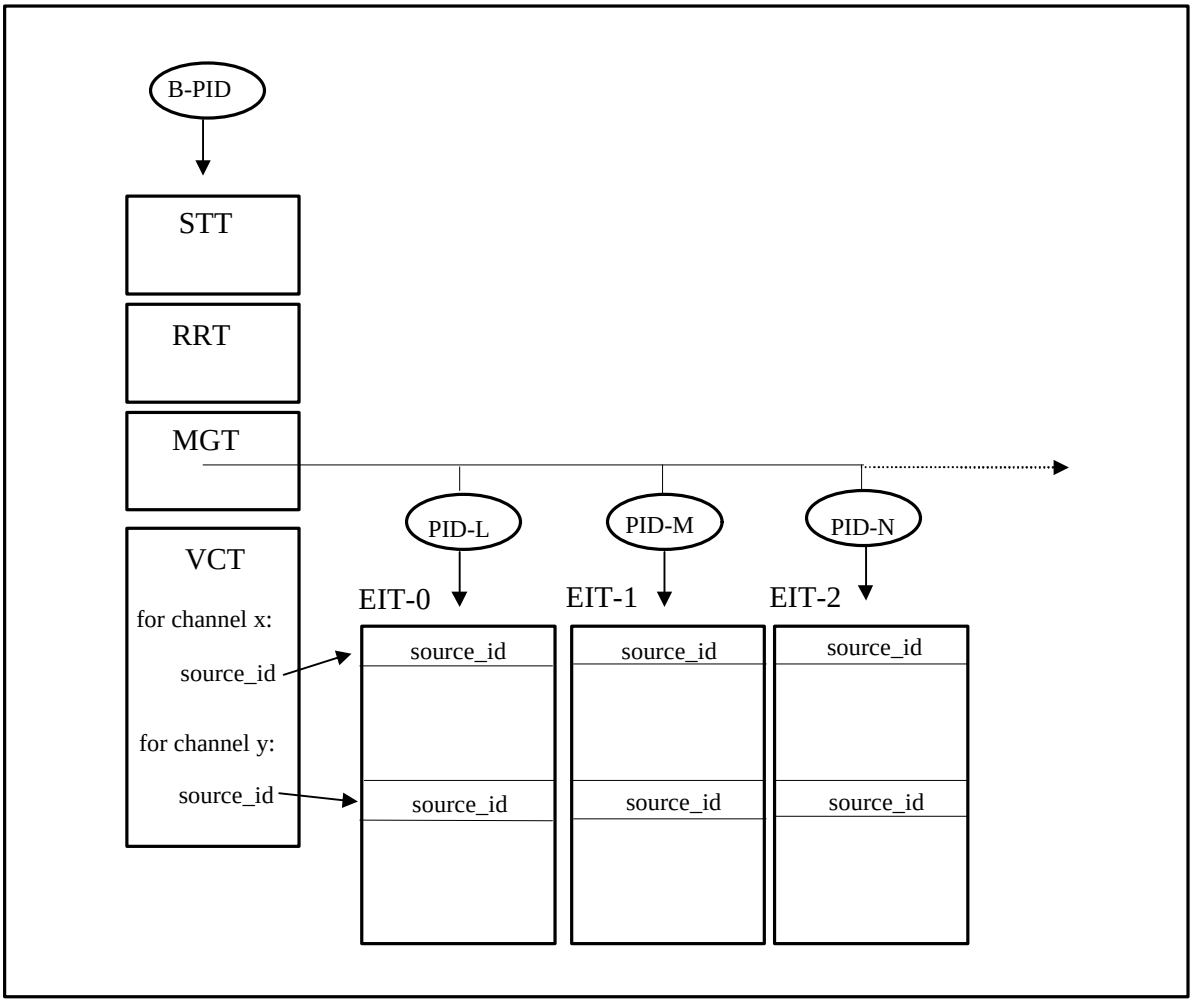

Figure 5.1 Table hierarchy for the Program and System Information Protocol (PSIP)

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.

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). These tables 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.

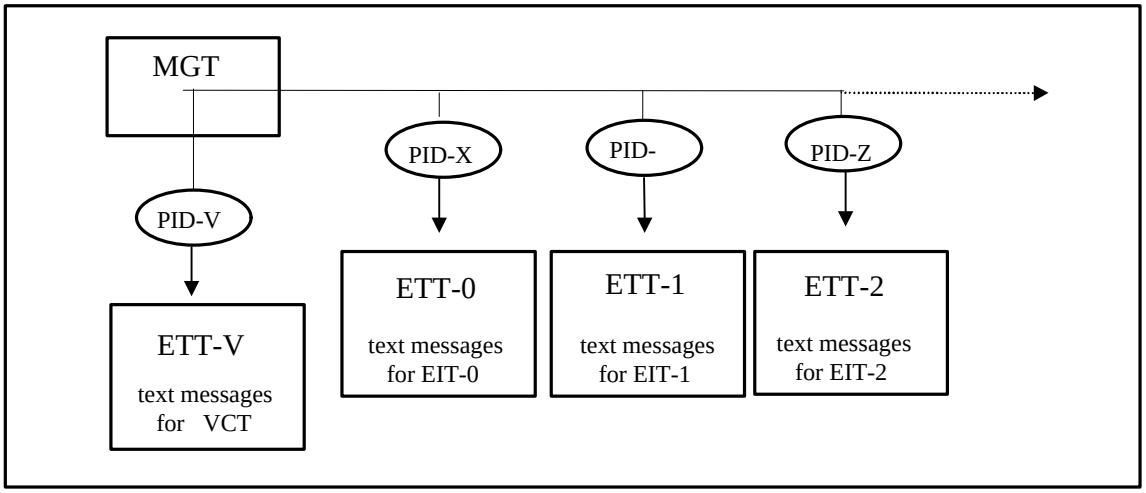

Figure 5.2 Extended Text Tables (ETTs) defined to carry text messages for describing virtual channels and events.

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 RRT, the TVCT, the MGT, and the first four Event Information Tables (EIT-0, EIT-1, EIT-2 and EIT-3). All of the other EITs and the whole collection of ETTs are optional.*

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, the RRT, the CVCT, and the MGT.*

Requirement 7: *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.*

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), 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	Bits	Format
system_time_table_section () {		
table_id	8	0xCD
section_syntax_indicator	1	'1'
private_indicator	1	'1'
zero	2	'00'
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++) {		
descriptors()	var	
}		
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 only be processed by decoders designed to accommodate the later versions as they become standardized.

system_time — A 32-bit unsigned integer quantity representing the current system time as the number of GPS seconds since 12 am, January 6th, 1980. The count of GPS seconds and leap second count shall be accurate and correct to within plus or minus four seconds, as timed at the arrival in the decoder of the Transport Stream packet carrying the last byte of the CRC.

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.

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 of the Private Section as described in Section 2.4.4.10 and 2.4.4.11 of ISO/IEC 13818-1. The following constraints apply to the Transport Stream packet 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 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	Bits	Format
master_guide_table_section () {		
table_id	8	0xC7
section_syntax_indicator	1	'1'
private_indicator	1	'1'
zero	2	'00'
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()	var	
}		
reserved	4	'1111'
descriptors_length	12	uimsbf
for (I = 0;I< N;I++)		
descriptor()	var	
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 only be processed by decoders designed to accommodate the later versions as they become standardized.

tables_defined — This 16-bit unsigned integer in the range 0 to 65535 represents the number of tables in the following loop.

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**
0x0005-0x00FF	**[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]**
0x1000-0xFFFF	**[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. 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). 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.

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).

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.

6.3 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, carrier frequency, 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.

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 of the Private Section as described in Section 2.4.4.10 and 2.4.4.11 of ISO/IEC 13818-1. 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_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.

Table 6.4 Bit Stream Syntax for the Terrestrial Virtual Channel Table

Syntax	Bits	Format
terrestrial_virtual_channel_table_section () {		
table_id	8	0xC8
section_syntax_indicator	1	'1'
private_indicator	1	'1'
zero	2	'00'
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	unicode™ BMP
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	6	'111111'
service_type	6	uimsbf
source_id	16	uimsbf
reserved	6	'111111'
descriptors_length	10	uimsbf
for (i=0;i<N;i++) {		
descriptors()		
}		
}		
reserved	6	'111111'
additional_descriptors_length	10	uimsbf
for(j=0; j<N;j++) {		
additional_descriptors()		
}		
CRC_32	32	rpchof
}		

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.

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 only be processed by decoders designed to accommodate the later versions as they become standardized.

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 character codes coded in accordance with the Basic Multilingual Plane (BMP) of Unicode™, as specified in ISO 10646-1. If the name of the virtual channel is shorter than seven Unicode™ characters, one or more instances of the null character value 0x0000 shall be used to pad the string to its fixed 14-byte length.

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 must 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. Values of modulation_mode 0x80 and above are

outside the scope of ATSC. These may be used to specify non-standard modulation modes in private systems. A value of 0x80 for modulation_mode indicates that modulation parameters are specified in a private descriptor.

Table 6.5 Modulation Modes

modulation_mode	meaning	terrestrial broadcast	cable
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. [12] (Mode 1). Typically, mode 1 will be used for 64-QAM.	Not valid	
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. [12] (Mode 2). Typically, mode 2 will be used for 256-QAM.	Not valid	
0x04	**ATSC (8 VSB)** — The virtual channel uses the 8-VSB modulation method conforming to the ATSC Digital Television Standard.		Not valid
0x05 -0x7F	[Reserved for future use by ATSC]		
0x80	Modulation parameters are defined by a private descriptor		
0x81-0xFF	[User Private]		

carrier_frequency— A 32-bit unsigned integer that represents the carrier frequency associated with the analog or digital transmission associated with this virtual channel, in units of one Hz. For VSB-modulated signals, the given carrier_frequency represents the location of the pilot tone; for analog signals, it represents the frequency of the picture carrier. In the case of a digital terrestrial broadcast signal that is transmitted at multiple carrier frequencies (via one or more translators), the carrier_frequency may be specified as zero. In such cases, the receiver is expected to associate the Transport Stream identified by the given transport_stream_id with the frequency tuned to acquire it.

For the ATSC Digital Television Standard, where the PTC bandwidth is 6 MHz, the pilot tone is located 310 kHz above the lower edge of the physical transmission channel, or 2.690 MHz below the specified center of the band. Similarly, for analog NTSC transmitted in the US, the picture carrier is 1.25 MHz above the lower edge of the 6 MHz physical transmission channel.

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. The receiver may use the channel_TSID to verify that a TS acquired at the referenced carrier frequency is actually the desired multiplex. Analog signals

may have a TSID provided that it is different from any DTV Transport Stream identifier; that is, it shall be truly unique if present.[2] A value of 0xFFFF for channel_TSID shall be specified for analog channels that do not have a valid 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.

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 entry. Typical applications for hidden channels are test signals and NVOD services.

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

[2] A method to include such a unique 16-bit "Transmission Signal ID" in the NTSC VBI is specified in the EIA-752 specification.

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 data) conforming to the ATSC Digital Television Standard
0x03	**ATSC_audio_only** — The virtual channel conforms to the ATSC Digital Television Standard, and has one or more standard audio and data components but no video.
0x04	**ATSC_data_broadcast_service** — Conforming to the ATSC data broadcast standard under development by T3/S13.
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.

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.

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 of the Private Section as described in Section 2.4.4.10 and 2.4.4.11 of ISO/IEC 13818-1. 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 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 — A 10-bit number in the range 1 to 999 that represents the "major" virtual channel number associated with the virtual channel being defined in this iteration of the "for" loop. Each virtual channel must be associated with a major and a minor virtual channel number.

The major virtual channel number, along with the minor virtual channel number, act as the user's reference number for the virtual channel.

minor_channel_number — A 10-bit number in the range 0 to 999 that represents the "minor" or "sub-" virtual channel number. This field, together with major_channel_number, performs a two-part virtual channel number, where minor_channel_number represents the second or right-hand part of the number

Table 6.8 Bit Stream Syntax for the Cable Virtual Channel Table

Syntax	Bits	Format
cable_virtual_channel_table_section () {		
table_id	8	0xC9
section_syntax_indicator	1	'1'
private_indicator	1	'1'
zero	2	'00'
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	unicode™ BMP
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
reserved	4	'1111'
service_type	6	uimsbf
source_id	16	uimsbf
reserved	6	'111111'
descriptors_length	10	uimsbf
for (i=0;i<N;i++) {		
descriptors()		
}		
}		
reserved	6	'111111'
additional_descriptors_length	10	uimsbf
for(j=0; j<N;j++) {		
additional_descriptors()		
}		
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.

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 an out-of-band physical transmission channel whose frequency is indicated by carrier_frequency. When clear, the virtual channel is carried within a standard tuned multiplex at that frequency.

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.

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_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.

Table 6.10 Bit Stream Syntax for the Rating Region Table

Syntax	Bits	Format
rating_region_table_section () {		
table_id	8	0xCA
section_syntax_indicator	1	'1'
private_indicator	1	'1'
zero	2	'00'
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++) {		
descriptors()	var	
}		
CRC_32	32	rpchof
}		

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. Values of rating_region are defined in Table 6.11.

Table 6.11 Rating Regions

rating_region	Rating Region Name
0x00	Forbidden
0x01	**US (50 states + possessions)**
0x02-0xFF	[Reserved]

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 6.8. 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 6.8. 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 6.8. 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 6.8. 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.

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, however its definition may be extended to include particular data broadcasting sessions and other information segments. 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.

The PSIP shall have at least four EITs and no more than 128 EITs, each of which provides the event information for a certain time span. 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..

The Event Information Table is carried in private sections with table ID 0xCB, and obeys the syntax and semantics of the Private Section as described in Section 2.4.4.10 and 2.4.4.11 of ISO/IEC 13818-1. 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 6.12.

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 only be processed by decoders designed to accommodate the later versions as they become standardized.

Table 6.12 Bit Stream Syntax for the Event Information Table

Syntax	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
zero	2	'00'
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
}		

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 12 am, January 6th, 1980.

ETM_location — This 2-bit field specifies the existence and the location of an Extended Text Message (ETM), based on Table 6.13

Table 6.13 ETM_location

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

length_in_seconds — Duration (in seconds) of this event.

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 6.8).

descriptors_length — Total length (in bytes) of the event 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-13818-1 "MPEG-2 Systems" after processing the entire Event Information Table section.

6.6 Extended Text Table

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 6.8), 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.

Within a Transport Stream, the Extended Text Message is carried on a private section with table ID 0xCC. 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 6.14.

Table 6.14 Bit Stream Syntax for the Extended Text Table

Syntax	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
table_id_extension	16	0x00
reserved	2	'11'
version_number	5	0x00
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 a 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.

table_id_extension — This 16-bit field shall be set to 0x00.

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 only be processed by decoders designed to accommodate the later versions as they become standardized.

ETM_id — Unique 32-bit identifier of this extended text message. This identifier is assigned by the rule shown in Table 6.15.

Table 6.15 ETM ID

	MSB				LSB	
Bit	**31**	**16**	**15**	**2**	**1**	**0**
channel ETM_id	source_id		0 ……	0	0	0
event ETM_id	source_id		event_id		1	0

extended_text_message() — The extended text message in the format of a multiple string structure (see Section 6.8).

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.

6.7 Core Descriptors

Table 6.16 lists all of the core descriptors and their descriptor tags. Asterisks mark the tables where the descriptors may appear. The range of MPEG-2 defined or reserved descriptor tags is between 0x02 and 0x3F.

Table 6.16 List of Descriptors for PSIP Tables.

Descriptor Name	Descriptor tag	Terrestrial				Cable		
		PMT	MGT	VCT	EIT	PMT	MGT	VCT
stuffing descriptor	0x80	*	*	*	*	*	*	*
AC-3 audio descriptor	0x81	*			*	*		
program identifier descriptor	0x85	*				*		
caption service descriptor	0x86	*			*	*		
content advisory descriptor	0x87	*			*	*		
extended channel name descriptor	0xA0			*				*
service location descriptor	0xA1			*				
time-shifted service descriptor	0xA2			*				*
component name descriptor	0xA3					*		
user private	0xC0-0xFF		*	*	*		*	*

6.7.1 AC-3 Audio Descriptor

The AC-3 audio descriptor, as defined in Ref. [1] and constrained in Annex B of Ref. [2] , may be used in the PMT and/or in EITs.

6.7.2 Program Identifier Descriptor

The program_identifier_descriptor, as defined in Ref. [5], may be used in the PMT.

6.7.3 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 not appear on events with no closed captioning service.

The bit stream syntax for the closed captioning service descriptor is shown in Table 6.17.

Table 6.17 Bit Stream Syntax for the Caption Service Descriptor

Syntax	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. [7]) 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.

6.7.4 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.

The bit stream syntax for the Content Advisory Descriptor is shown in Table 6.18.

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.

Table 6.18 Bit Stream Syntax for the Content Advisory Descriptor

Syntax	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	
}		
}		

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 6.8). 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().

6.7.5 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 6.19.

Table 6.19 Bit Stream Syntax for the Extended Channel Name Descriptor

Syntax	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 6.8).

6.7.6 Service Location Descriptor

This descriptor specifies the stream types, PID and language code for each elementary stream. This descriptor shall appear in the TVCT, and must be valid for the current event in the corresponding virtual channel.

The bit stream syntax for the service location descriptor is shown in Table 6.20.

Table 6.20 Bit Stream Syntax for the Service Location Descriptor

Syntax	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.

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 according to Table 6.21.

Table 6.21 Stream Type Assignments

Value	Description
0x00	ITU-T | ISO/IEC Reserved
0x01-0x7F	As specified in Table 2.29 (Stream type assignments) of Ref. [10]
0x80	[Used in other systems]
0x81	ATSC A/53 audio
0x82-0x84	[Used in other systems]
0x85	UPID (Ref.[5])
0x86-0xBF	Reserved
0xC0-0xFF	User Private

elementary_PID — Packet Identifier for the elementary stream.

ISO_639_language_code — This 3-byte (24 bits) field, based on 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.

6.7.7 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 bit stream syntax for the time_shifted_service_descriptor() is shown in Table 6.22.

Table 6.22 Bit Stream Syntax for the Time Shifted Service Descriptor

Syntax	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.

6.7.8 Component Name Descriptor

Table 6.23 defines the component_name_descriptor(), which serves to define an optional textual name tag for any component of the service.

Table 6.23 Bit Stream Syntax for the Component Name Descriptor

Syntax	Bits	Format
component_name_descriptor() {		
descriptor_tag		0xA3
descriptor_length		uimsbf
component_name_string()		
	ar	
}		

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 6.8).

6.7.9 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.

6.8 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 6.24.

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, based on 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.

Table 6.24 Bit Stream Syntax for the Multiple String Structure

Syntax	Bits	Format
multiple_string_structure () {		
number_strings	8	uimsbf
for (i= 0;i< number_strings;i++) {		
ISO_639_language_code	8*3	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
}		
}		
}		

compression_type — This 8-bit field identifies the compression type for the j^{th} segment. Allowed values for this field are shown in Table 6.25.

Table 6.25 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	user private

mode — An 8-bit value representing the text mode to be used to interpret characters in the segment to follow. See Table 6.26 for definition. Mode values in the range zero through 0x3E select 8-bit Unicode™ character code pages. Mode value 0x3F selects 16-bit Unicode™ character coding. Mode values 0x40 through 0xDF are reserved for future use by ATSC. Mode values 0xE0 through 0xFE are user private. Mode value 0xFF indicates the text mode is not applicable. Decoders shall ignore string bytes associated with unknown or unsupported mode values.

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

compressed_string_byte[k] — The k^{th} byte of the j^{th} segment.

Table 6.26 Modes

Mode	Meaning	Language(s) or Script
0x00	Select ISO/IEC 10646-1 Page 0x00	ASCII, ISO Latin-1 (Roman)[3]
0x01	Select ISO/IEC 10646-1 Page 0x01	European Latin (many)[4]
0x02	Select ISO/IEC 10646-1 Page 0x02	Standard Phonetic
0x03	Select ISO/IEC 10646-1 Page 0x03	Greek
0x04	Select ISO/IEC 10646-1 Page 0x04	Russian, Slavic
0x05	Select ISO/IEC 10646-1 Page 0x05	Armenian, Hebrew
0x06	Select ISO/IEC 10646-1 Page 0x06	Arabic[5]
0x07-0x08	Reserved	-
0x09	Select ISO/IEC 10646-1 Page 0x09	Devanagari[6], Bengali
0x0A	Select ISO/IEC 10646-1 Page 0x0A	Punjabi, Gujarati
0x0B	Select ISO/IEC 10646-1 Page 0x0B	Oriya, Tamil
0x0C	Select ISO/IEC 10646-1 Page 0x0C	Telugu, Kannada
0x0D	Select ISO/IEC 10646-1 Page 0x0D	Malayalam
0x0E	Select ISO/IEC 10646-1 Page 0x0E	Thai, Lao
0x0F	Reserved	-
0x10	Select ISO/IEC 10646-1 Page 0x10	Tibetan, Georgian
0x11-0x1F	Reserved	-
0x20	Select ISO/IEC 10646-1 Page 0x20	Miscellaneous
0x21	Select ISO/IEC 10646-1 Page 0x21	Misc. symbols, arrows
0x22	Select ISO/IEC 10646-1 Page 0x22	Mathematical operators
0x23	Select ISO/IEC 10646-1 Page 0x23	Misc. technical
0x24	Select ISO/IEC 10646-1 Page 0x24	OCR, enclosed alpha-num.
0x25	Select ISO/IEC 10646-1 Page 0x25	Form and chart components
0x26	Select ISO/IEC 10646-1 Page 0x26	Miscellaneous dingbats
0x27	Select ISO/IEC 10646-1 Page 0x27	Zapf dingbats
0x28-0x2F	Reserved	-
0x30	Select ISO/IEC 10646-1 Page 0x30	Hiragana, Katakana
0x31	Select ISO/IEC 10646-1 Page 0x31	Bopomopho, Hangul elem.
0x32	Select ISO/IEC 10646-1 Page 0x32	Enclosed CJK Letters, ideo.
0x33	Select ISO/IEC 10646-1 Page 0x33	Enclosed CJK Letters, ideo.
0x34-0x3E	Reserved	-
0x3F	Select 16-bit ISO/IEC 10646-1 mode	all
0x40-0xDF	Reserved	
0xE0-0xFE	User private	
0xFF	Not applicable	

[3] The languages supported by ASCII plus the Latin-1 supplement include Danish, Dutch, English, Faroese, Finnish, Flemish, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish and Swedish. Many other languages can be written with this set of characters, including Hawaiian, Indonesian, and Swahili.

[4] When combined with page zero (ASCII and ISO Latin-1), covers Afrikaans, Breton, Basque, Catalan, Croatian, Czech, Esperanto, Estonian, French , Frisian, Greenlandic, Hungarian, Latin, Latvian, Lithuanian, Maltese, Polish, Provencal, Rhaeto-Romanic, Romanian, Romany, Sami, Slovak, Slovenian, Sorbian, Turkish, Welsh, and many others.

[5] Also Persian, Urdu, Pashto, Sindhi, and Kurdish.

[6] Devanagari script is used for writing Sanskrit and Hindi, as well as other languages of northern India (such as Marathi) and of Nepal (Nepali). In addition, at least two dozen other Indian languages use Devanagari script.

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.

ANNEX A

(Normative)

DAYLIGHT SAVINGS TIME CONTROL

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 A.1, and the basic use of daylight savings fields through the year is shown in Table A.2.

Table A.1 Structure of Daylight Savings Time Control

Syntax	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 A.2 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

(Normative)

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

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

- For broadcasters with existing NTSC licenses, the major_channel_number for the existing NTSC channels, as well as the Digital TV 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 will use major_channel_number 13 for identification of the analog NTSC channel on RF channel 13, as well as the digital channels it is controlling on RF channel 39.
- For a new broadcaster without an existing NTSC license, the major_channel_number for the Digital TV 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 will use major_channel_number 49 for identification of the digital channels that it is controlling on RF channel 49.
- The two provisions above assign major_channel_number values 2 through 69 uniquely to broadcasters with license to broadcast NTSC and/or Digital ATSC signals.
- 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.

ANNEX C

(Normative)

STANDARD HUFFMAN TABLES FOR TEXT COMPRESSION[7]

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 C.2 defines standard Huffman encode and decode tables optimized for English-language text such as that typically found in program titles. Section C.3 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 C.5 and C.7 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.

C1. 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 C.1 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.

[7] Tables C.4 through C.7 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.

C1.1 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 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.

C1.2 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 C.1.
- **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 C.2.

  Decode tables have the following format:

Table C.2 Decode Table Format

Syntax	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.

C1.2.1 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.

C1.2.2 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 following format:

Table C.3 Decode Tree Format

Syntax	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_character_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_character_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 Table F.3 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.

C2. 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 C.4 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: 0111011100
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 C.5 English-language Program Title Decode Table

0	1	79	220	158	3	237	34	316	155	395	4	474	155
1	0	80	1	159	100	238	7	317	155	396	155	475	160
2	1	81	230	160	3	239	44	318	155	397	226	476	4
3	58	82	1	161	122	240	7	319	155	398	5	477	243
4	1	83	232	162	3	241	70	320	155	399	6	478	228
5	60	84	1	163	148	242	7	321	155	400	7	479	185
6	1	85	234	164	3	243	84	322	155	401	8	480	1
7	62	86	1	165	152	244	7	323	155	402	9	481	244
8	1	87	240	166	3	245	124	324	155	403	213	482	160
9	64	88	1	167	164	246	7	325	155	404	10	483	155
10	1	89	242	168	3	247	138	326	155	405	214	484	2
11	66	90	1	169	200	248	7	327	155	406	11	485	3
12	1	91	244	170	3	249	140	328	155	407	217	486	155
13	68	92	2	171	222	250	7	329	155	408	12	487	155
14	1	93	6	172	3	251	142	330	155	409	166	488	155
15	70	94	2	173	230	252	7	331	155	410	233	489	155
16	1	95	18	174	3	253	144	332	155	411	203	490	1
17	72	96	2	175	244	254	7	333	155	412	197	491	2
18	1	97	20	176	4	255	146	334	155	413	207	492	155
19	74	98	2	177	4	256	27	335	155	414	13	493	193
20	1	99	28	178	4	257	28	336	155	415	14	494	200
21	76	100	2	179	6	258	180	337	155	416	202	495	211
22	1	101	40	180	4	259	164	338	155	417	201	496	155
23	78	102	2	181	12	260	178	339	155	418	15	497	155
24	1	103	48	182	4	261	183	340	155	419	199	498	155
25	80	104	2	183	16	262	218	341	155	420	16	499	160
26	1	105	52	184	4	263	1	342	155	421	17	500	7
27	82	106	2	185	18	264	209	343	155	422	225	501	8
28	1	107	54	186	4	265	2	344	155	423	18	502	177
29	84	108	2	187	20	266	3	345	155	424	19	503	210
30	1	109	56	188	4	267	155	346	155	425	198	504	211
31	86	110	2	189	22	268	4	347	155	426	210	505	212
32	1	111	58	190	4	269	213	348	155	427	200	506	213
33	88	112	2	191	24	270	217	349	155	428	206	507	173
34	1	113	60	192	4	271	5	350	155	429	193	508	205
35	90	114	2	193	26	272	203	351	155	430	196	509	193
36	1	115	62	194	4	273	214	352	155	431	208	510	1
37	92	116	2	195	28	274	6	353	155	432	204	511	2
38	1	117	70	196	4	275	207	354	155	433	20	512	3
39	94	118	2	197	82	276	7	355	155	434	21	513	160
40	1	119	72	198	4	277	8	356	155	435	239	514	4
41	96	120	2	199	106	278	202	357	155	436	194	515	155
42	1	121	74	200	4	279	9	358	155	437	215	516	5
43	98	122	2	201	142	280	201	359	155	438	22	517	6
44	1	123	76	202	4	281	197	360	155	439	205	518	160
45	100	124	2	203	174	282	198	361	155	440	23	519	5
46	1	125	78	204	4	283	10	362	155	441	244	520	201
47	102	126	2	205	238	284	210	363	155	442	212	521	215
48	1	127	80	206	5	285	196	364	155	443	24	522	211
49	104	128	2	207	6	286	199	365	155	444	25	523	1
50	1	129	82	208	5	287	204	366	155	445	26	524	2
51	106	130	2	209	40	288	208	367	155	446	195	525	155
52	1	131	84	210	5	289	200	368	155	447	211	526	174
53	108	132	2	211	68	290	215	369	155	448	27	527	128
54	1	133	126	212	5	291	206	370	155	449	28	528	3
55	110	134	2	213	114	292	11	371	155	450	29	529	4
56	1	135	146	214	5	293	193	372	155	451	30	530	155
57	112	136	2	215	118	294	12	373	155	452	31	531	155
58	1	137	172	216	5	295	194	374	155	453	32	532	2
59	114	138	2	217	144	296	205	375	155	454	33	533	3
60	1	139	186	218	5	297	195	376	41	455	34	534	173
61	116	140	2	219	190	298	13	377	42	456	35	535	155
62	1	141	210	220	5	299	14	378	216	457	36	536	1
63	118	142	2	221	214	300	15	379	229	458	37	537	128
64	1	143	228	222	6	301	16	380	185	459	38	538	160
65	120	144	2	223	10	302	211	381	1	460	39	539	176
66	1	145	250	224	6	303	17	382	167	461	40	540	4
67	206	146	3	225	68	304	212	383	177	462	1	541	5
68	1	147	6	226	6	305	18	384	236	463	128	542	128
69	210	148	3	227	100	306	19	385	209	464	160	543	155
70	1	149	30	228	6	307	20	386	2	465	155	544	177
71	212	150	3	229	102	308	21	387	173	466	155	545	178
72	1	151	38	230	6	309	22	388	178	467	155	546	160
73	214	152	3	231	154	310	23	389	218	468	155	547	176
74	1	153	50	232	6	311	24	390	227	469	155	548	185
75	216	154	3	233	208	312	25	391	179	470	177	549	1
76	1	155	62	234	6	313	26	392	3	471	155	550	2
77	218	156	3	235	252	314	155	393	228	472	155	551	3
78	1	157	82	236	7	315	155	394	230	473	155	552	2

553	3	634	15	715	4	796	9	877	1	958	229	1039	225
554	177	635	16	716	5	797	10	878	236	959	240	1040	155
555	186	636	17	717	225	798	2	879	2	960	232	1041	155
556	1	637	18	718	6	799	3	880	3	961	10	1042	155
557	176	638	8	719	7	800	155	881	160	962	11	1043	155
558	155	639	9	720	8	801	245	882	155	963	12	1044	155
559	128	640	193	721	9	802	1	883	4	964	13	1045	155
560	128	641	211	722	7	803	225	884	5	965	244	1046	155
561	1	642	155	723	8	804	239	885	245	966	14	1047	155
562	176	643	1	724	160	805	229	886	6	967	15	1048	155
563	155	644	195	725	155	806	5	887	7	968	232	1049	155
564	155	645	2	726	204	807	233	888	238	969	10	1050	155
565	184	646	233	727	1	808	225	889	8	970	173	1051	155
566	155	647	236	728	229	809	239	890	11	971	206	1052	25
567	155	648	3	729	2	810	245	891	12	972	155	1053	26
568	155	649	242	730	236	811	238	892	160	973	1	1054	155
569	155	650	245	731	245	812	155	893	243	974	214	1055	186
570	155	651	4	732	239	813	229	894	249	975	2	1056	229
571	176	652	239	733	3	814	1	895	174	976	245	1057	234
572	155	653	225	734	233	815	2	896	210	977	247	1058	248
573	160	654	5	735	242	816	3	897	199	978	3	1059	1
574	2	655	229	736	4	817	4	898	1	979	4	1060	2
575	3	656	6	737	5	818	4	899	155	980	225	1061	230
576	177	657	7	738	225	819	5	900	2	981	229	1062	167
577	179	658	11	739	6	820	160	901	245	982	233	1063	3
578	185	659	12	740	9	821	155	902	3	983	5	1064	250
579	176	660	193	741	10	822	1	903	4	984	242	1065	232
580	1	661	249	742	174	823	245	904	5	985	6	1066	4
581	155	662	1	743	236	824	2	905	233	986	239	1067	247
582	155	663	194	744	249	825	229	906	236	987	7	1068	5
583	160	664	207	745	193	826	239	907	6	988	8	1069	245
584	155	665	229	746	232	827	3	908	229	989	9	1070	226
585	155	666	245	747	1	828	225	909	7	990	238	1071	6
586	155	667	155	748	155	829	233	910	239	991	3	1072	235
587	155	668	233	749	2	830	8	911	8	992	236	1073	7
588	155	669	2	750	3	831	9	912	225	993	174	1074	240
589	155	670	160	751	4	832	170	913	9	994	1	1075	8
590	155	671	3	752	225	833	212	914	242	995	155	1076	128
591	155	672	4	753	245	834	1	915	10	996	2	1077	246
592	155	673	5	754	233	835	155	916	1	997	240	1078	231
593	128	674	242	755	5	836	227	917	245	998	6	1079	9
594	155	675	6	756	229	837	2	918	155	999	233	1080	228
595	155	676	236	757	6	838	242	919	214	1000	160	1081	10
596	19	677	7	758	242	839	3	920	4	1001	195	1082	160
597	20	678	225	759	239	840	229	921	5	1002	239	1083	233
598	170	679	8	760	7	841	4	922	232	1003	155	1084	11
599	173	680	9	761	8	842	245	923	155	1004	229	1085	227
600	174	681	232	762	239	843	249	924	1	1005	1	1086	249
601	246	682	10	763	5	844	233	925	245	1006	128	1087	12
602	231	683	239	764	128	845	5	926	2	1007	2	1088	13
603	244	684	5	765	155	846	239	927	225	1008	3	1089	237
604	226	685	6	766	245	847	6	928	233	1009	225	1090	14
605	233	686	249	767	1	848	7	929	239	1010	4	1091	15
606	1	687	155	768	2	849	225	930	3	1011	5	1092	243
607	2	688	1	769	233	850	229	931	229	1012	6	1093	16
608	194	689	245	770	225	851	8	932	16	1013	7	1094	17
609	240	690	2	771	3	852	206	933	17	1014	198	1095	236
610	155	691	242	772	229	853	160	934	170	1015	215	1096	18
611	243	692	233	773	4	854	198	935	236	1016	1	1097	244
612	227	693	229	774	238	855	245	936	241	1017	155	1098	242
613	230	694	239	775	11	856	1	937	174	1018	242	1099	19
614	247	695	3	776	186	857	2	938	160	1019	2	1100	238
615	3	696	225	777	212	858	155	939	247	1020	3	1101	20
616	245	697	4	778	174	859	194	940	237	1021	232	1102	21
617	4	698	10	779	242	860	3	941	238	1022	229	1103	22
618	5	699	11	780	227	861	225	942	1	1023	225	1104	23
619	6	700	241	781	1	862	4	943	2	1024	4	1105	24
620	242	701	245	782	160	863	239	944	155	1025	233	1106	10
621	7	702	243	783	2	864	5	945	235	1026	239	1107	11
622	8	703	1	784	128	865	233	946	3	1027	5	1108	243
623	9	704	237	785	155	866	6	947	4	1028	155	1109	155
624	10	705	249	786	237	867	7	948	5	1029	155	1110	245
625	11	706	195	787	3	868	9	949	6	1030	2	1111	226
626	12	707	2	788	201	869	10	950	227	1031	239	1112	1
627	228	708	236	789	243	870	228	951	7	1032	225	1113	128
628	160	709	238	790	244	871	243	952	239	1033	155	1114	160
629	13	710	228	791	4	872	230	953	8	1034	1	1115	2
630	236	711	248	792	5	873	246	954	233	1035	229	1116	229
631	238	712	3	793	6	874	247	955	245	1036	1	1117	242
632	14	713	155	794	7	875	240	956	9	1037	239	1118	233
633	237	714	246	795	8	876	242	957	225	1038	155	1119	3

1120	236	1201	161	1282	229	1363	240	1444	6	1525	238	1606	237
1121	4	1202	173	1283	8	1364	5	1445	7	1526	225	1607	167
1122	249	1203	232	1284	9	1365	6	1446	8	1527	13	1608	155
1123	5	1204	234	1285	10	1366	7	1447	243	1528	243	1609	228
1124	239	1205	241	1286	15	1367	225	1448	9	1529	14	1610	1
1125	6	1206	245	1287	16	1368	8	1449	245	1530	233	1611	249
1126	225	1207	250	1288	186	1369	230	1450	10	1531	15	1612	243
1127	7	1208	1	1289	249	1370	242	1451	239	1532	16	1613	242
1128	8	1209	2	1290	167	1371	237	1452	11	1533	244	1614	244
1129	9	1210	3	1291	244	1372	246	1453	12	1534	128	1615	2
1130	16	1211	4	1292	155	1373	9	1454	128	1535	228	1616	232
1131	17	1212	186	1293	1	1374	228	1455	249	1536	229	1617	3
1132	195	1213	248	1294	231	1375	10	1456	225	1537	17	1618	236
1133	204	1214	167	1295	236	1376	239	1457	13	1538	18	1619	240
1134	199	1215	226	1296	2	1377	244	1458	228	1539	231	1620	4
1135	155	1216	233	1297	238	1378	236	1459	233	1540	160	1621	225
1136	227	1217	5	1298	3	1379	243	1460	160	1541	19	1622	233
1137	1	1218	6	1299	239	1380	231	1461	14	1542	20	1623	5
1138	128	1219	7	1300	245	1381	229	1462	15	1543	21	1624	6
1139	236	1220	230	1301	4	1382	11	1463	236	1544	22	1625	128
1140	249	1221	237	1302	242	1383	227	1464	229	1545	23	1626	160
1141	2	1222	231	1303	5	1384	12	1465	16	1546	27	1627	7
1142	243	1223	235	1304	6	1385	13	1466	17	1547	28	1628	8
1143	3	1224	8	1305	233	1386	14	1467	18	1548	174	1629	9
1144	245	1225	9	1306	7	1387	15	1468	19	1549	250	1630	10
1145	4	1226	246	1307	243	1388	16	1469	20	1550	191	1631	229
1146	5	1227	240	1308	225	1389	17	1470	10	1551	1	1632	239
1147	242	1228	10	1309	8	1390	18	1471	11	1552	167	1633	11
1148	6	1229	239	1310	9	1391	19	1472	249	1553	155	1634	12
1149	233	1230	11	1311	10	1392	238	1473	155	1554	2	1635	13
1150	160	1231	227	1312	11	1393	20	1474	245	1555	233	1636	155
1151	7	1232	12	1313	229	1394	239	1475	243	1556	248	1637	245
1152	8	1233	13	1314	128	1395	1	1476	1	1557	249	1638	24
1153	239	1234	14	1315	12	1396	155	1477	2	1558	3	1639	25
1154	244	1235	249	1316	232	1397	225	1478	226	1559	229	1640	186
1155	9	1236	15	1317	160	1398	11	1479	237	1560	232	1641	172
1156	10	1237	228	1318	13	1399	12	1480	128	1561	4	1642	246
1157	225	1238	236	1319	14	1400	212	1481	3	1562	225	1643	155
1158	11	1239	16	1320	229	1401	239	1482	240	1563	235	1644	240
1159	232	1240	229	1321	13	1402	230	1483	239	1564	5	1645	226
1160	235	1241	17	1322	226	1403	236	1484	4	1565	226	1646	1
1161	229	1242	244	1323	245	1404	247	1485	160	1566	6	1647	230
1162	12	1243	247	1324	247	1405	225	1486	5	1567	7	1648	2
1163	13	1244	18	1325	155	1406	1	1487	233	1568	227	1649	167
1164	14	1245	19	1326	236	1407	186	1488	6	1569	8	1650	174
1165	15	1246	225	1327	1	1408	2	1489	225	1570	231	1651	231
1166	14	1247	20	1328	249	1409	155	1490	7	1571	244	1652	3
1167	15	1248	21	1329	238	1410	249	1491	8	1572	9	1653	227
1168	174	1249	22	1330	2	1411	3	1492	9	1573	128	1654	245
1169	245	1250	238	1331	3	1412	4	1493	229	1574	246	1655	4
1170	247	1251	243	1332	4	1413	5	1494	24	1575	240	1656	237
1171	1	1252	23	1333	242	1414	243	1495	25	1576	10	1657	5
1172	236	1253	128	1334	5	1415	6	1496	226	1577	228	1658	6
1173	2	1254	24	1335	128	1416	7	1497	234	1578	11	1659	7
1174	228	1255	25	1336	6	1417	8	1498	242	1579	243	1660	235
1175	231	1256	242	1337	160	1418	233	1499	232	1580	247	1661	8
1176	242	1257	26	1338	225	1419	160	1500	236	1581	12	1662	9
1177	3	1258	27	1339	239	1420	9	1501	237	1582	13	1663	238
1178	155	1259	160	1340	7	1421	128	1502	250	1583	239	1664	242
1179	239	1260	28	1341	244	1422	229	1503	155	1584	236	1665	10
1180	4	1261	29	1342	233	1423	10	1504	1	1585	160	1666	228
1181	246	1262	160	1343	8	1424	21	1505	245	1586	14	1667	11
1182	5	1263	11	1344	9	1425	22	1506	2	1587	15	1668	249
1183	6	1264	245	1345	10	1426	167	1507	3	1588	237	1669	236
1184	249	1265	155	1346	11	1427	186	1508	246	1589	230	1670	12
1185	243	1266	1	1347	12	1428	227	1509	4	1590	16	1671	13
1186	7	1267	236	1348	21	1429	247	1510	186	1591	245	1672	244
1187	233	1268	243	1349	22	1430	242	1511	230	1592	17	1673	128
1188	225	1269	242	1350	161	1431	173	1512	5	1593	18	1674	14
1189	8	1270	128	1351	248	1432	226	1513	6	1594	19	1675	239
1190	9	1271	225	1352	233	1433	1	1514	235	1595	20	1676	243
1191	128	1272	2	1353	235	1434	2	1515	239	1596	21	1677	160
1192	10	1273	3	1354	1	1435	155	1516	7	1597	242	1678	225
1193	11	1274	244	1355	128	1436	230	1517	167	1598	22	1679	15
1194	229	1275	233	1356	155	1437	3	1518	249	1599	238	1680	233
1195	12	1276	239	1357	250	1438	237	1519	8	1600	23	1681	16
1196	13	1277	230	1358	226	1439	246	1520	9	1601	24	1682	17
1197	160	1278	4	1359	2	1440	4	1521	10	1602	25	1683	229
1198	30	1279	5	1360	3	1441	235	1522	11	1603	26	1684	18
1199	31	1280	6	1361	4	1442	5	1523	227	1604	14	1685	19
1200	155	1281	7	1362	160	1443	244	1524	12	1605	15	1686	20

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

C3. 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 C.6 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: 'I' 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 C.7 English-language Program Description Decode Table

0	1	79	242	158	3	237	134	316	155	395	197	474	52
1	0	80	1	159	16	238	6	317	155	396	198	475	53
2	1	81	248	160	3	239	146	318	155	397	177	476	54
3	44	82	1	161	26	240	6	319	155	398	10	477	55
4	1	83	250	162	3	241	170	320	155	399	238	478	155
5	46	84	1	163	40	242	6	321	155	400	203	479	155
6	1	85	252	164	3	243	184	322	155	401	11	480	3
7	48	86	1	165	42	244	6	323	155	402	212	481	4
8	1	87	254	166	3	245	220	324	155	403	12	482	128
9	50	88	2	167	52	246	6	325	155	404	196	483	174
10	1	89	0	168	3	247	236	326	155	405	200	484	200
11	52	90	2	169	74	248	6	327	155	406	210	485	212
12	1	91	4	170	3	249	238	328	155	407	13	486	1
13	54	92	2	171	90	250	6	329	155	408	14	487	2
14	1	93	22	172	3	251	240	330	155	409	15	488	155
15	56	94	2	173	94	252	6	331	155	410	199	489	160
16	1	95	32	174	3	253	242	332	155	411	202	490	155
17	58	96	2	175	100	254	6	333	155	412	206	491	155
18	1	97	34	176	3	255	244	334	155	413	208	492	155
19	60	98	2	177	110	256	20	335	155	414	215	493	155
20	1	99	44	178	3	257	21	336	155	415	16	494	155
21	62	100	2	179	112	258	155	337	155	416	194	495	155
22	1	101	50	180	3	259	214	338	155	417	17	496	155
23	64	102	2	181	114	260	201	339	155	418	204	497	155
24	1	103	56	182	3	261	207	340	155	419	236	498	2
25	66	104	2	183	116	262	215	341	155	420	229	499	243
26	1	105	60	184	3	263	199	342	155	421	231	500	160
27	68	106	2	185	118	264	1	343	155	422	18	501	244
28	1	107	64	186	3	265	162	344	155	423	205	502	155
29	70	108	2	187	120	266	206	345	155	424	19	503	1
30	1	109	68	188	3	267	203	346	155	425	20	504	155
31	72	110	2	189	122	268	2	347	155	426	195	505	155
32	1	111	70	190	3	269	3	348	155	427	21	506	172
33	74	112	2	191	124	270	197	349	155	428	22	507	155
34	1	113	74	192	3	271	204	350	155	429	23	508	155
35	76	114	2	193	126	272	198	351	155	430	237	509	155
36	1	115	76	194	3	273	200	352	155	431	24	510	155
37	78	116	2	195	128	274	4	353	155	432	25	511	155
38	1	117	84	196	3	275	196	354	155	433	242	512	1
39	80	118	2	197	180	276	5	355	155	434	26	513	160
40	1	119	86	198	3	277	194	356	155	435	211	514	155
41	82	120	2	199	206	278	6	357	155	436	27	515	162
42	1	121	88	200	3	279	195	358	155	437	28	516	7
43	84	122	2	201	240	280	210	359	155	438	228	517	8
44	1	123	90	202	4	281	7	360	155	439	29	518	226
45	86	124	2	203	26	282	211	361	155	440	193	519	228
46	1	125	92	204	4	283	8	362	56	441	227	520	229
47	88	126	2	205	88	284	202	363	57	442	30	521	230
48	1	127	94	206	4	285	212	364	173	443	233	522	160
49	90	128	2	207	110	286	9	365	175	444	240	523	242
50	1	129	96	208	4	287	205	366	183	445	226	524	225
51	92	130	2	209	142	288	208	367	218	446	247	525	1
52	1	131	98	210	4	289	10	368	168	447	31	526	2
53	94	132	2	211	172	290	193	369	179	448	243	527	243
54	1	133	118	212	4	291	11	370	181	449	230	528	227
55	96	134	2	213	216	292	12	371	1	450	32	529	3
56	1	135	132	214	4	293	13	372	2	451	33	530	4
57	98	136	2	215	224	294	14	373	155	452	34	531	5
58	1	137	148	216	4	295	15	374	180	453	232	532	155
59	100	138	2	217	244	296	16	375	241	454	239	533	6
60	1	139	162	218	5	297	17	376	162	455	35	534	4
61	102	140	2	219	36	298	18	377	213	456	36	535	128
62	1	141	178	220	5	299	19	378	214	457	37	536	202
63	104	142	2	221	64	300	155	379	217	458	38	537	211
64	1	143	186	222	5	301	155	380	3	459	39	538	162
65	106	144	2	223	118	302	155	381	4	460	40	539	1
66	1	145	200	224	5	303	155	382	5	461	41	540	155
67	222	146	2	225	174	304	155	383	207	462	42	541	2
68	1	147	210	226	5	305	155	384	6	463	244	542	3
69	224	148	2	227	206	306	155	385	201	464	43	543	160
70	1	149	222	228	5	307	155	386	249	465	44	544	155
71	234	150	2	229	208	308	155	387	234	466	45	545	160
72	1	151	234	230	6	309	155	388	235	467	46	546	3
73	236	152	2	231	6	310	155	389	245	468	47	547	4
74	1	153	242	232	6	311	155	390	246	469	225	548	155
75	238	154	2	233	52	312	155	391	7	470	48	549	183
76	1	155	252	234	6	313	155	392	8	471	49	550	244
77	240	156	3	235	96	314	155	393	9	472	50	551	160
78	1	157	8	236	6	315	155	394	178	473	51	552	176

553	243	634	245	715	229	796	155	877	2	958	236	1039	243
554	1	635	1	716	233	797	232	878	155	959	160	1040	12
555	2	636	2	717	245	798	233	879	155	960	4	1041	233
556	185	637	225	718	225	799	1	880	155	961	233	1042	13
557	2	638	239	719	1	800	242	881	239	962	242	1043	14
558	184	639	229	720	239	801	236	882	155	963	245	1044	15
559	155	640	233	721	2	802	2	883	155	964	5	1045	16
560	160	641	242	722	4	803	239	884	155	965	249	1046	229
561	1	642	3	723	5	804	3	885	155	966	225	1047	17
562	174	643	4	724	160	805	229	886	155	967	6	1048	18
563	2	644	6	725	201	806	4	887	155	968	239	1049	160
564	182	645	7	726	243	807	5	888	155	969	7	1050	29
565	155	646	155	727	155	808	155	889	155	970	229	1051	30
566	1	647	233	728	174	809	155	890	155	971	8	1052	169
567	160	648	249	729	242	810	3	891	155	972	9	1053	232
568	160	649	242	730	1	811	4	892	155	973	10	1054	245
569	1	650	245	731	2	812	155	893	155	974	15	1055	155
570	155	651	1	732	3	813	174	894	155	975	16	1056	1
571	176	652	2	733	238	814	1	895	155	976	241	1057	173
572	174	653	3	734	239	815	233	896	24	977	174	1058	187
573	1	654	236	735	5	816	2	897	25	978	196	1059	235
574	155	655	239	736	155	817	225	898	232	979	249	1060	250
575	160	656	225	737	174	818	229	899	239	980	172	1061	2
576	174	657	4	738	233	819	239	900	248	981	1	1062	167
577	1	658	232	739	229	820	9	901	155	982	227	1063	230
578	160	659	5	740	1	821	10	902	167	983	2	1064	226
579	155	660	5	741	245	822	246	903	247	984	155	1065	231
580	155	661	6	742	2	823	249	904	250	985	242	1066	3
581	155	662	249	743	225	824	1	905	1	986	3	1067	4
582	155	663	242	744	3	825	174	906	2	987	4	1068	5
583	1	664	245	745	4	826	227	907	3	988	160	1069	6
584	172	665	155	746	229	827	233	908	4	989	236	1070	233
585	174	666	229	747	3	828	245	909	229	990	245	1071	248
586	155	667	239	748	225	829	155	910	174	991	5	1072	7
587	155	668	1	749	233	830	229	911	5	992	6	1073	172
588	2	669	2	750	242	831	239	912	230	993	233	1074	239
589	3	670	233	751	155	832	2	913	226	994	7	1075	240
590	155	671	225	752	1	833	3	914	6	995	235	1076	8
591	160	672	3	753	2	834	225	915	246	996	8	1077	237
592	181	673	4	754	3	835	4	916	235	997	244	1078	246
593	182	674	6	755	4	836	232	917	245	998	9	1079	249
594	184	675	7	756	155	837	5	918	233	999	229	1080	9
595	1	676	225	757	233	838	6	919	7	1000	10	1081	247
596	155	677	233	758	245	839	244	920	240	1001	239	1082	10
597	160	678	238	759	1	840	7	921	249	1002	225	1083	11
598	155	679	246	760	229	841	8	922	231	1003	232	1084	174
599	160	680	228	761	2	842	232	923	8	1004	11	1085	12
600	155	681	236	762	239	843	7	924	9	1005	12	1086	227
601	155	682	243	763	225	844	229	925	228	1006	13	1087	13
602	155	683	1	764	225	845	247	926	10	1007	14	1088	229
603	155	684	2	765	5	846	214	927	227	1008	19	1089	244
604	155	685	242	766	155	847	225	928	11	1009	20	1090	14
605	155	686	3	767	227	848	155	929	237	1010	167	1091	15
606	155	687	4	768	239	849	233	930	12	1011	187	1092	228
607	160	688	155	769	1	850	242	931	243	1012	230	1093	16
608	155	689	5	770	245	851	1	932	13	1013	237	1094	236
609	155	690	2	771	229	852	2	933	14	1014	247	1095	17
610	8	691	3	772	2	853	3	934	15	1015	231	1096	225
611	9	692	229	773	3	854	4	935	236	1016	246	1097	18
612	230	693	236	774	233	855	239	936	16	1017	1	1098	19
613	245	694	155	775	4	856	5	937	244	1018	2	1099	20
614	243	695	239	776	229	857	6	938	17	1019	155	1100	21
615	244	696	1	777	3	858	174	939	18	1020	238	1101	22
616	155	697	242	778	155	859	1	940	242	1021	3	1102	238
617	228	698	5	779	233	860	155	941	160	1022	4	1103	243
618	1	699	6	780	1	861	238	942	19	1023	236	1104	23
619	237	700	245	781	225	862	233	943	20	1024	5	1105	24
620	2	701	239	782	239	863	2	944	21	1025	245	1106	242
621	3	702	155	783	2	864	229	945	238	1026	6	1107	160
622	4	703	236	784	3	865	155	946	22	1027	172	1108	25
623	242	704	233	785	4	866	160	947	23	1028	228	1109	26
624	5	705	1	786	167	867	1	948	11	1029	249	1110	27
625	6	706	225	787	238	868	3	949	12	1030	242	1111	28
626	236	707	242	788	236	869	4	950	228	1031	7	1112	9
627	238	708	2	789	242	870	155	951	243	1032	8	1113	10
628	7	709	229	790	243	871	232	952	155	1033	9	1114	174
629	160	710	3	791	1	872	229	953	174	1034	174	1115	155
630	5	711	4	792	155	873	225	954	226	1035	10	1116	236
631	6	712	3	793	2	874	239	955	1	1036	239	1117	1
632	155	713	4	794	225	875	1	956	2	1037	11	1118	245
633	236	714	155	795	6	876	233	957	3	1038	225	1119	2

1120	244	1201	155	1282	244	1363	249	1444	18	1525	243	1606	5
1121	230	1202	174	1283	172	1364	5	1445	242	1526	14	1607	6
1122	3	1203	250	1284	4	1365	6	1446	19	1527	15	1608	7
1123	225	1204	1	1285	5	1366	235	1447	20	1528	16	1609	8
1124	229	1205	235	1286	230	1367	239	1448	21	1529	225	1610	244
1125	233	1206	2	1287	237	1368	7	1449	238	1530	239	1611	174
1126	4	1207	160	1288	246	1369	8	1450	22	1531	17	1612	245
1127	242	1208	3	1289	6	1370	9	1451	23	1532	233	1613	9
1128	239	1209	4	1290	174	1371	10	1452	24	1533	18	1614	10
1129	5	1210	240	1291	240	1372	172	1453	25	1534	19	1615	242
1130	6	1211	5	1292	7	1373	11	1454	14	1535	229	1616	225
1131	7	1212	6	1293	8	1374	12	1455	15	1536	20	1617	243
1132	160	1213	230	1294	243	1375	227	1456	173	1537	160	1618	11
1133	8	1214	246	1295	9	1376	174	1457	237	1538	21	1619	12
1134	14	1215	7	1296	10	1377	13	1458	249	1539	22	1620	13
1135	15	1216	228	1297	228	1378	238	1459	155	1540	23	1621	233
1136	173	1217	237	1298	11	1379	233	1460	174	1541	24	1622	14
1137	231	1218	231	1299	12	1380	14	1461	1	1542	160	1623	15
1138	155	1219	8	1300	249	1381	225	1462	243	1543	22	1624	239
1139	167	1220	225	1301	13	1382	15	1463	2	1544	162	1625	229
1140	249	1221	239	1302	239	1383	243	1464	3	1545	167	1626	16
1141	1	1222	242	1303	14	1384	16	1465	245	1546	226	1627	160
1142	236	1223	9	1304	225	1385	17	1466	244	1547	235	1628	232
1143	2	1224	10	1305	15	1386	244	1467	240	1548	237	1629	17
1144	172	1225	11	1306	16	1387	18	1468	4	1549	238	1630	18
1145	242	1226	236	1307	233	1388	231	1469	239	1550	155	1631	19
1146	3	1227	12	1308	236	1389	229	1470	5	1551	247	1632	17
1147	174	1228	229	1309	17	1390	19	1471	233	1552	1	1633	18
1148	243	1229	227	1310	160	1391	20	1472	6	1553	2	1634	239
1149	245	1230	13	1311	229	1392	228	1473	232	1554	3	1635	246
1150	4	1231	244	1312	18	1393	21	1474	160	1555	187	1636	155
1151	5	1232	14	1313	19	1394	22	1475	225	1556	249	1637	235
1152	239	1233	243	1314	20	1395	23	1476	236	1557	240	1638	249
1153	6	1234	15	1315	21	1396	160	1477	7	1558	4	1639	1
1154	7	1235	16	1316	12	1397	24	1478	242	1559	5	1640	160
1155	233	1236	17	1317	13	1398	26	1479	8	1560	236	1641	226
1156	225	1237	238	1318	167	1399	27	1480	229	1561	6	1642	2
1157	8	1238	18	1319	187	1400	194	1481	9	1562	7	1643	225
1158	9	1239	19	1320	155	1401	155	1482	10	1563	8	1644	3
1159	232	1240	3	1321	1	1402	173	1483	11	1564	245	1645	237
1160	10	1241	239	1322	249	1403	172	1484	12	1565	225	1646	4
1161	11	1242	155	1323	174	1404	248	1485	13	1566	9	1647	227
1162	229	1243	225	1324	226	1405	1	1486	155	1567	172	1648	233
1163	12	1244	229	1325	2	1406	174	1487	245	1568	227	1649	5
1164	160	1245	245	1326	237	1407	2	1488	25	1569	10	1650	228
1165	13	1246	1	1327	243	1408	3	1489	26	1570	232	1651	229
1166	13	1247	2	1328	3	1409	229	1490	169	1571	11	1652	231
1167	14	1248	8	1329	245	1410	231	1491	187	1572	233	1653	6
1168	167	1249	9	1330	239	1411	232	1492	246	1573	12	1654	236
1169	172	1250	236	1331	240	1412	249	1493	230	1574	239	1655	240
1170	243	1251	249	1332	4	1413	233	1494	1	1575	243	1656	7
1171	173	1252	167	1333	5	1414	235	1495	155	1576	174	1657	8
1172	1	1253	238	1334	233	1415	4	1496	173	1577	13	1658	9
1173	2	1254	1	1335	6	1416	227	1497	226	1578	14	1659	10
1174	155	1255	172	1336	7	1417	225	1498	240	1579	229	1660	11
1175	249	1256	155	1337	8	1418	5	1499	2	1580	15	1661	243
1176	245	1257	174	1338	9	1419	246	1500	167	1581	16	1662	12
1177	174	1258	2	1339	160	1420	6	1501	3	1582	17	1663	244
1178	3	1259	3	1340	225	1421	228	1502	4	1583	244	1664	238
1179	238	1260	4	1341	229	1422	7	1503	5	1584	18	1665	13
1180	4	1261	243	1342	10	1423	226	1504	245	1585	19	1666	242
1181	242	1262	5	1343	11	1424	240	1505	227	1586	20	1667	14
1182	5	1263	233	1344	25	1425	8	1506	172	1587	21	1668	15
1183	6	1264	6	1345	26	1426	9	1507	231	1588	20	1669	16
1184	244	1265	160	1346	173	1427	243	1508	242	1589	21	1670	5
1185	7	1266	7	1347	187	1428	244	1509	6	1590	187	1671	229
1186	8	1267	229	1348	226	1429	247	1510	235	1591	226	1672	243
1187	9	1268	22	1349	234	1430	239	1511	7	1592	173	1673	249
1188	239	1269	23	1350	237	1431	10	1512	236	1593	237	1674	155
1189	225	1270	167	1351	242	1432	11	1513	237	1594	1	1675	1
1190	160	1271	173	1352	250	1433	12	1514	238	1595	155	1676	239
1191	10	1272	238	1353	230	1434	13	1515	249	1596	167	1677	2
1192	233	1273	227	1354	236	1435	236	1516	8	1597	227	1678	3
1193	11	1274	235	1355	1	1436	14	1517	174	1598	172	1679	225
1194	12	1275	242	1356	2	1437	15	1518	9	1599	236	1680	4
1195	229	1276	155	1357	3	1438	16	1519	10	1600	238	1681	233
1196	20	1277	226	1358	155	1439	245	1520	228	1601	2	1682	10
1197	21	1278	1	1359	245	1440	237	1521	11	1602	247	1683	11
1198	172	1279	2	1360	4	1441	17	1522	12	1603	3	1684	174
1199	226	1280	245	1361	167	1442	230	1523	244	1604	4	1685	155
1200	248	1281	3	1362	246	1443	160	1524	13	1605	249	1686	236

1687	237
1688	1
1689	2
1690	243
1691	238
1692	242
1693	3
1694	229
1695	4
1696	232
1697	160
1698	225
1699	5
1700	239
1701	6
1702	7
1703	8
1704	233
1705	9
1706	5
1707	6
1708	160
1709	172
1710	173
1711	244
1712	233
1713	1
1714	2
1715	225
1716	229
1717	3
1718	155
1719	4
1720	17
1721	160
1722	191
1723	225
1724	226
1725	230
1726	237
1727	228
1728	233
1729	247
1730	167
1731	1
1732	2
1733	187
1734	3
1735	4
1736	236
1737	5
1738	155
1739	238
1740	6
1741	239
1742	7
1743	172
1744	229
1745	243
1746	8
1747	9
1748	10
1749	174
1750	11
1751	12
1752	13
1753	14
1754	15
1755	16
1756	6
1757	7
1758	160
1759	174
1760	225
1761	229
1762	236
1763	250
1764	155
1765	239
1766	233
1767	1
1768	2
1769	3
1770	4
1771	5
1772	155
1773	155
1774	155
1775	155
1776	155
1777	155
1778	155
1779	155
1780	155
1781	155

ANNEX D

(Informative)

AN OVERVIEW OF PSIP FOR TERRESTRIAL BROADCAST WITH APPLICATION EXAMPLES

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.

D1. INTRODUCTION

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.

D2. ELEMENTS OF PSIP

PSIP is a collection of hierarchically-associated tables each of which describes particular elements of typical digital TV services. Figures D.1 and D.2 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 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.

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..

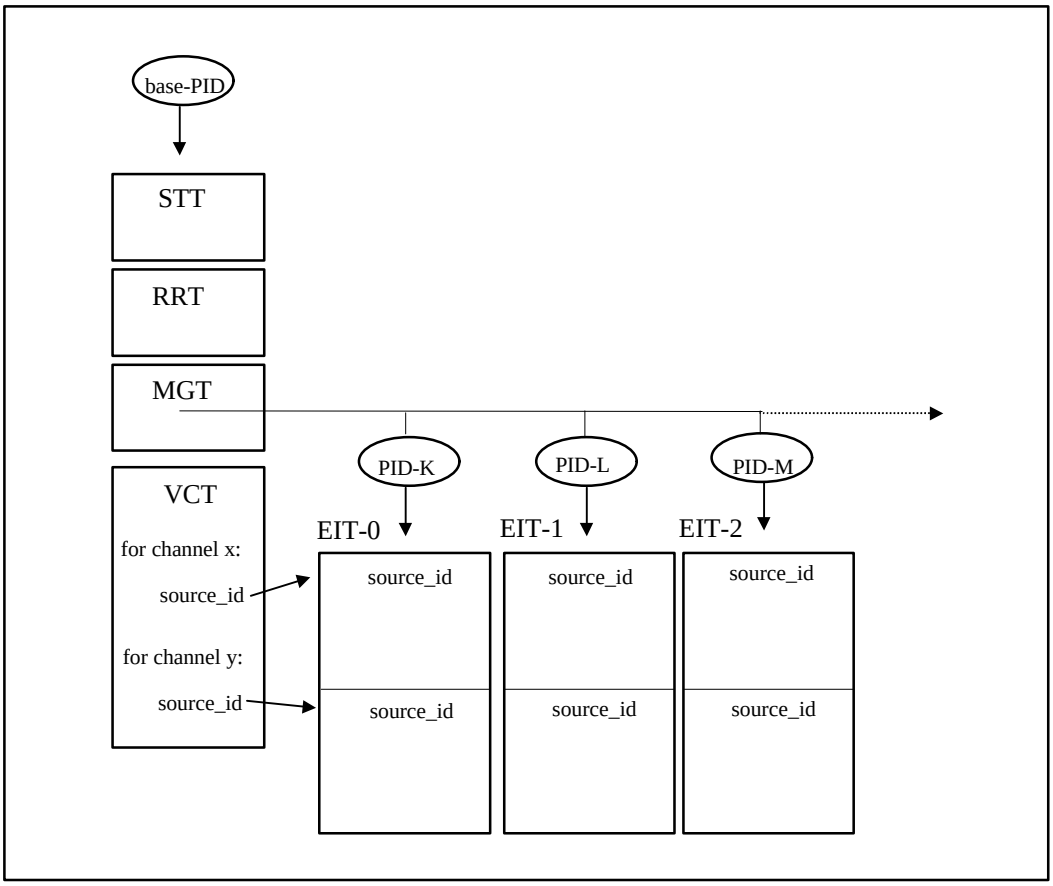

Figure D.1 Main Structure for the PSIP tables

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; 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 D.1 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.

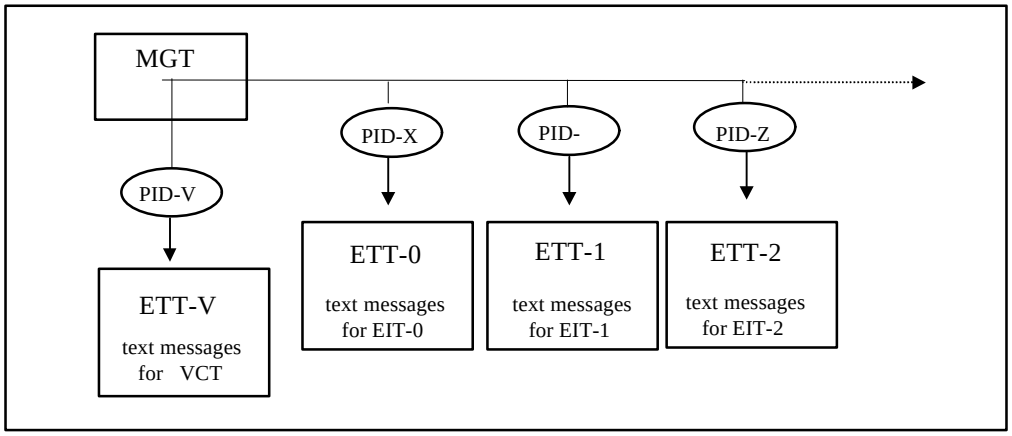

Figure D.2 Extended Text Tables in the PSIP hierarchy.

As illustrated in Fig. D.2, 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.

D3. 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 D.2 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	NBZ	City Life	City Life	Travel Show	Travel Show	News	News
PTC 39 VC #1	NBZ	City Life	City Life	Travel Show	Travel Show	News	News
PTC 39 VC #2	NBZ	Soccer	Golf Report	Golf Report	Car Racing	Car Racing	Car Racing
PTC 39 VC #3	NBZ	Secret Agent	Secret Agent	Lost Worlds	Lost Worlds	Lost Worlds	Lost Worlds
PTC 39 VC #4	NBZ	headlines	headlines	headlines	headlines	headlines	headlines

Table D.3 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	NBZ	Music Today	NY Comedy	World View	World View	News	News
PTC 39 VC #1	NBZ	Music Today	NY Comedy	World View	World View	News	News
PTC 39 VC #2	NBZ	Car Racing	Car Racing	Sports News	Tennis Playoffs	Tennis Playoffs	Tennis Playoffs
PTC 39 VC #3	NBZ	Preview	The Bandit	The Bandit	The Bandit	The Bandit	Preview
PTC 39 VC #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 as VC), one matching the analog transmission (simulcast), another for sports, and a third one for movies. The fourth one supports a service displaying headlines with text and images.

D3.1 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). Fig. D.3 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 - USA	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 D.3 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.

D3.2 The Virtual Channel Table (VCT)

Figure D.4 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 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	carrier freq. (MHz)	channel TS ID	progr. num	flags	service type	source id	descriptors
12	0	NBZ	205.25	0x0AA0	0x0AA0	--	analog	20	ch_name
12	1	NBZD	620.31	0x0AA1	0x00F1	--	digital	21	ch_name serv_locat.
12	5	NBZ-S	620.31	0x0AA1	0x00F2	--	digital	2	ch_name serv_locat.
12	12	NBZ-M	620.31	0x0AA1	0x00F3	--	digital	23	ch_name serv_locat.
12	31	NBZ-H	620.31	0x0AA1	0x00F8	--	digital	24	ch_name serv_locat.

Figure D.4 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. Although carrying the next VCT is optional, its use is recommended to give receivers advance notification of the new parameters that become operational during a VCT update.

Assume for example that a Transport Stream contains a VCT with a version number of 6 which has been operational for 20 hours. At 10:00 p.m., a football game using much more bandwidth will be broadcast, and for this reason, the number of available channels and PIDs will

be redefined. Around 9:30 p.m., simultaneous transmission of the next VCT can start with a version number of 7. By continuously monitoring the MGT, a receiver can be informed that a next VCT is available. The receiver may want to cache the new VCT for future use. The receiver continues monitoring the MGT and when this table signals a version change for the current VCT (from 6 to 7), then the cached information can be used.

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."[8] For the example VCT in Figure D.4, the Transmission Signal Identifier for channel 12.0 is 0x0AA0. A receiver can use the Transmission Signal ID given in the analog channel's channel_TSID field to verify that the NTSC signal received at the frequency given in the VCT is actually the desired signal. In the case that the Transmission Signal ID is not known or not available, the channel_TSID field may contain 0xFFFF to indicate "unknown."

D3.3 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 D.5 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 D.5. 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 EST 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).

[8] A method to include such a unique 16-bit “Transmission Signal ID” in the NTSC VBI is specified in the EIA-752 specification.

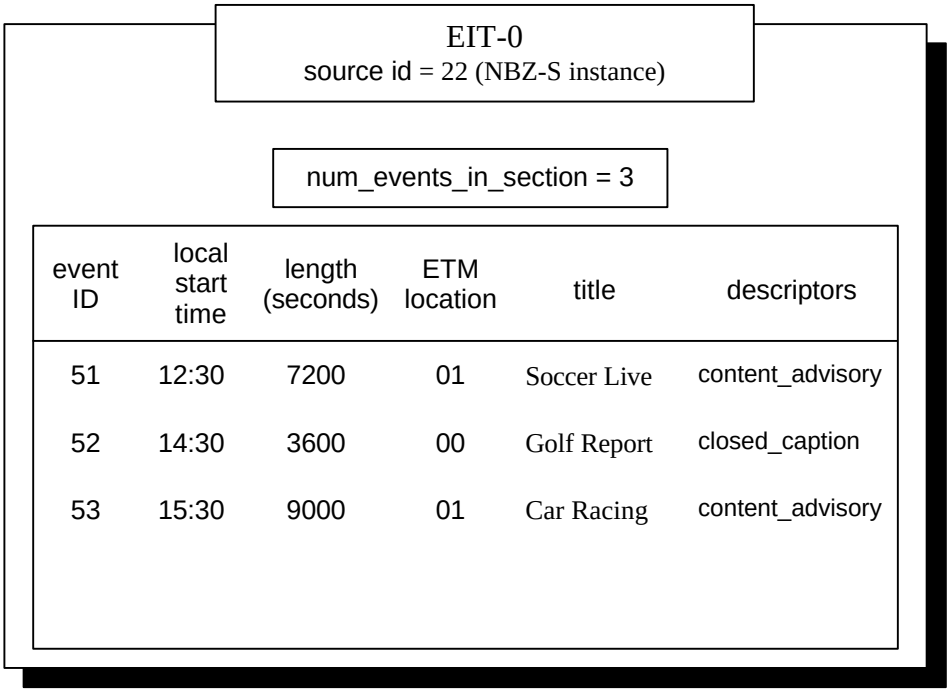

Figure D.5 Content of EIT-0 for NBZ-S

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. The most important 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. When a closed caption descriptor is included, it signals the existence of closed captioning and lists the necessary parameters for decoding.

D3.4 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 D.6 shows an example of one instance of an RRT, defined as the first rating region and carrying the MPAA standard rating system. Changes in the content of the RRT must be defined and approved by the ATSC. 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.

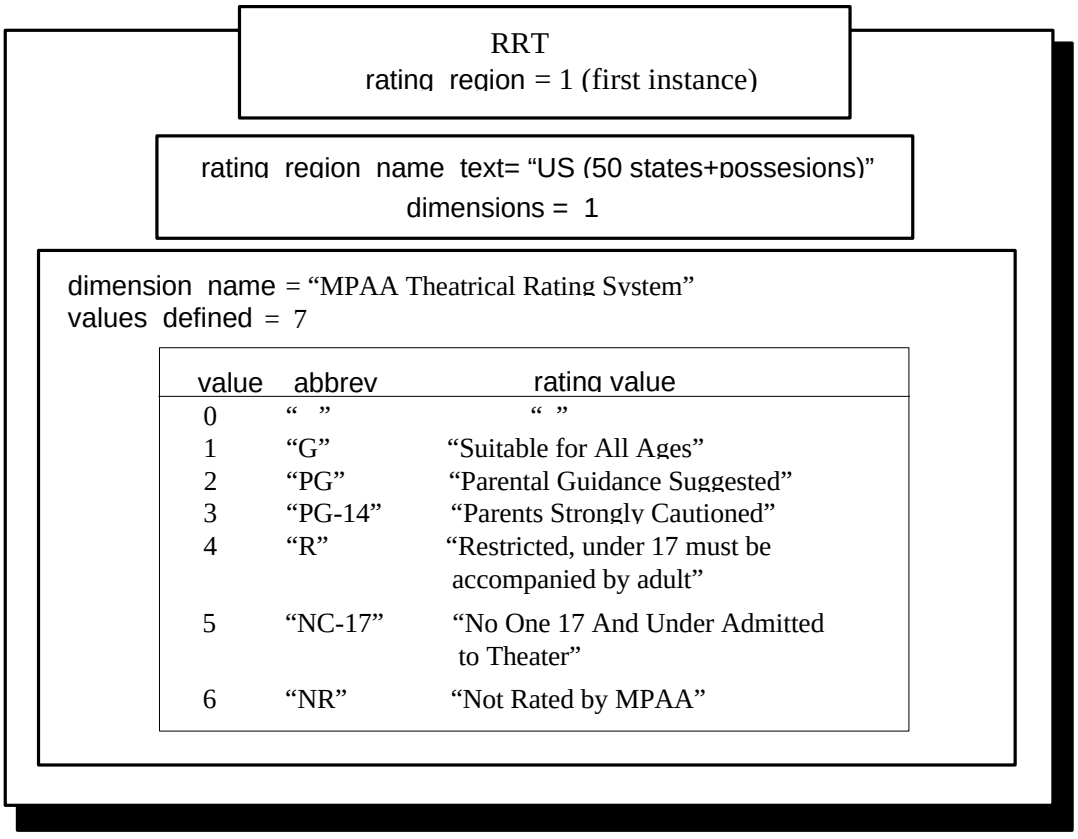

Figure D.6 An instance of a Rating Region Table (RRT).

D4. 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 D.1 and D.2. 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 D.7 illustrates this process.

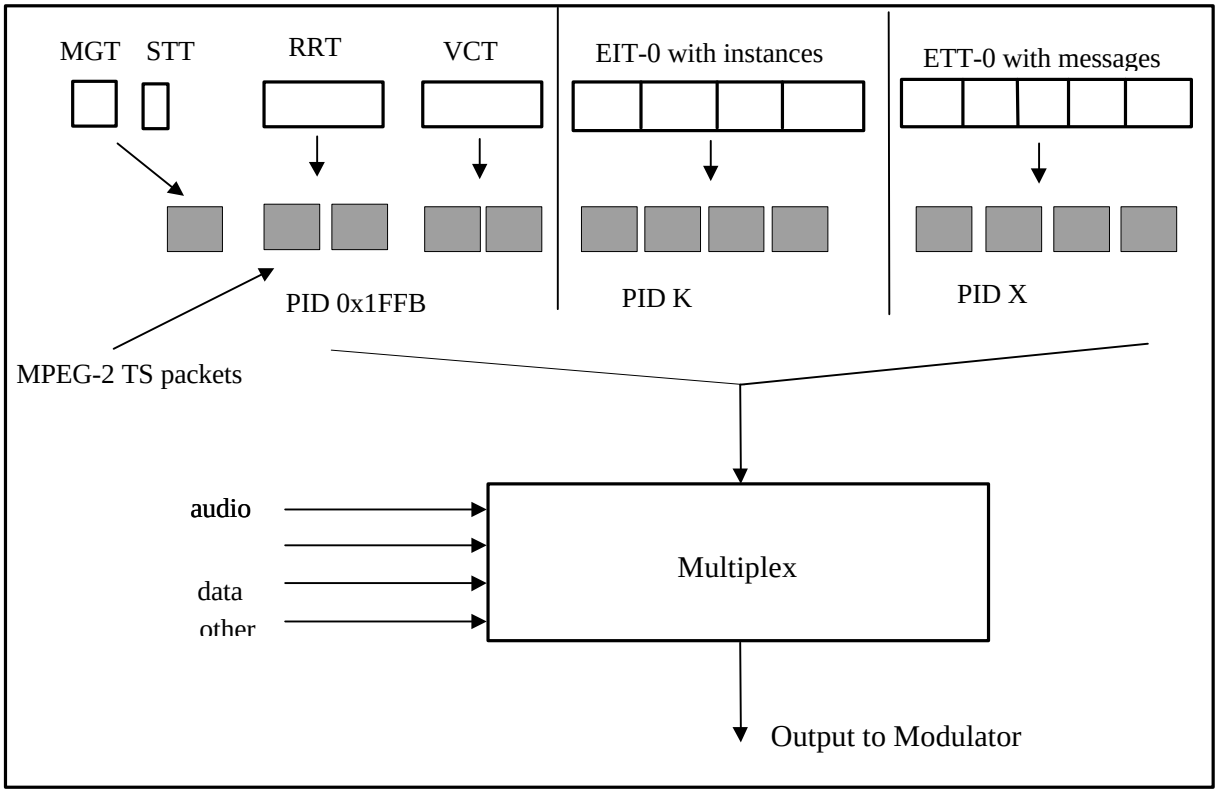

Figure D.7 Packetization and transport of the PSIP tables

D.5 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 D.8 shows this process.

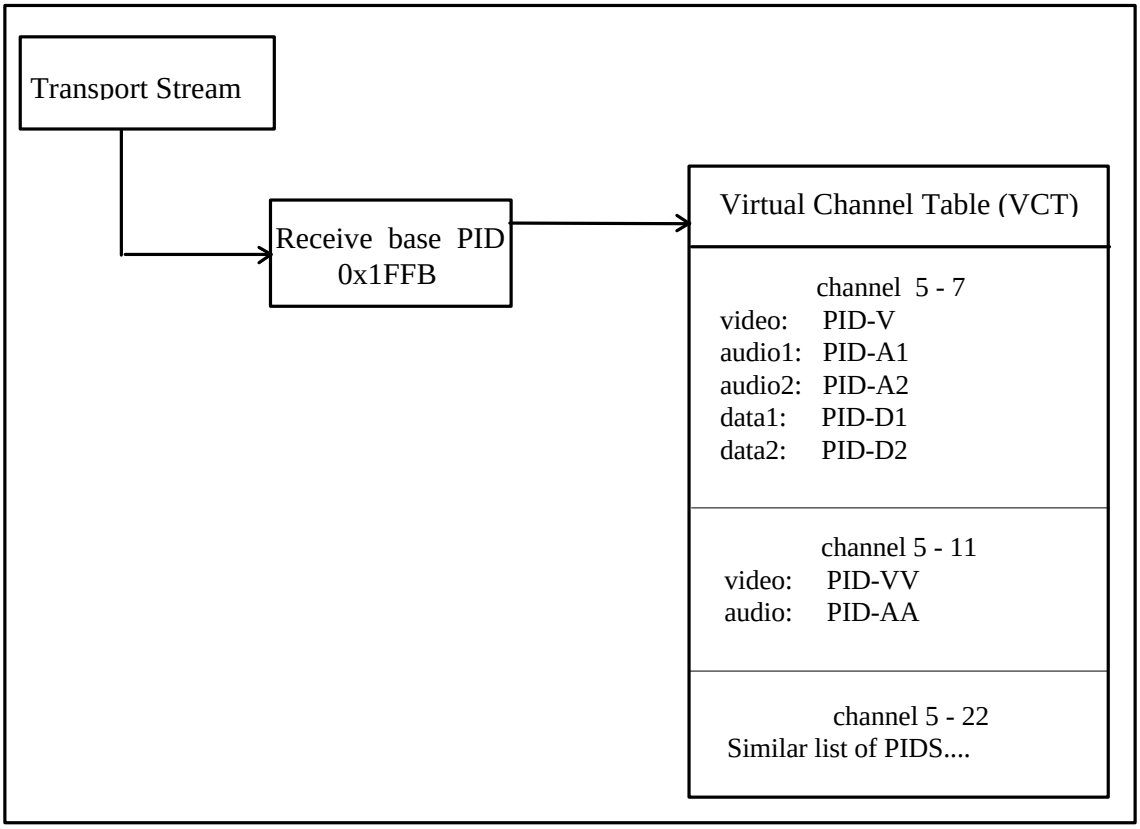

Figure D.8 Extraction of the VCT from the Transport Stream

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 Fig. D.9. For terrestrial broadcast, the existence of a service location descriptor in the VCT is mandatory and therefore there is no need to access the PAT or PMT for tuning. This feature has been included in PSIP to minimize the time required for changing and tuning to channels. However, PAT and PMT information must be present in the Transport Stream to support the general MPEG-2 compliance

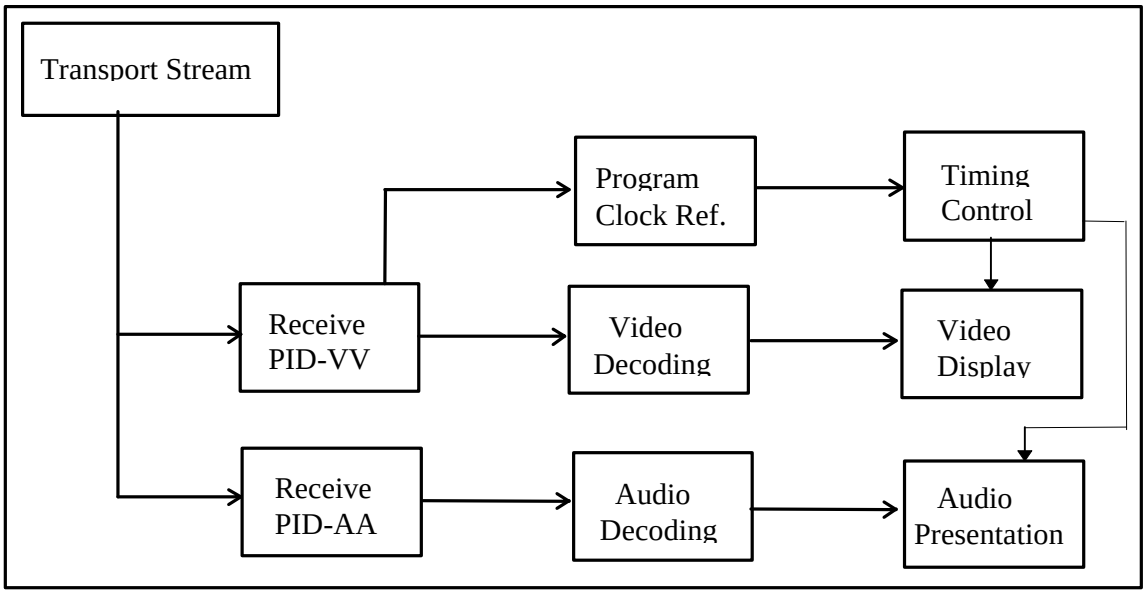

Figure D.9 Acquisition of audiovisual components

ANNEX E

(Informative)

TYPICAL SIZE OF PSIP TABLES

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 E.1.

Table E.1 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 virtual channel
R	number of rating regions
D	average number of rating dimensions per rating region
L	average number of rating values per rating dimension

E1. SYSTEM TIME TABLE (STT)

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

E2. 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 E.2

Table E.2 Typical size (bytes) of MGT

Part	Size (bytes)	Assumption
PSI header and trailer	12	
message body	38+22*P	1. With one Terrestrial VCT, one channel ETT, one RRT instance, P EITs and P event ETTs 2. No descriptors
Total	50+22*P	

E3. 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 E.3.

Table E.3 Typical TVCT size (bytes)

Part	Size (bytes)	Assumption
PSI header and trailer	12	1. All TVCT messages are carried in one section.
message body	4+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	

E4. 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 E.4.

Table E.4 Typical size (in bytes per rating region) of RRT

Part	Size (bytes per rating region)	Assumption
PSI header and trailer	12	1. One section only.
message body	25+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)	

E5. 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 E.5.

Table E.5 Typical size (bytes per virtual channel per EIT) of EIT

Part	Size (bytes per virtual channel per EIT)	Assumption
PSI header and trailer	12	1. One section only
message body	2+30*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. No AC-3 and service location descriptors.
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+(42+R*(3+2*D))*E	

E6. EXTENDED TEXT TABLE (ETT)

The typical size for the ETT (in bytes per virtual channel per EIT, or bytes per event per EIT), based on the assumptions listed in the column labeled "Assumptions", is shown in Table E.6.

Table E.6 Typical size (bytes per virtual channel or bytes per event) of ETT

Part	Size (bytes per virtual channel per EIT, or bytes per event per EIT)	Assumptions
PSI header and trailer	12	
message body	508	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 500 bytes.
Total	520	

E7. 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, each with 6 events in EIT-0, EIT-1, EIT-2 and EIT-3. For each virtual channel and each event an extended text message is available.

Regarding the Rating Region Table, suppose that a single rating region is defined with six dimensions and five values per dimension. Based on these assumptions, typical sizes for every PSIP table can be calculated. The results are listed in Table E.7 and Table E.8.

Table E.7 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	138	1
TVCT	443	3
RRT	901	5
Subtotal for tables identified by the base_PID	**1502**	**10**
EIT-0	2136	12
EIT-1	2136	12
EIT-2	2136	12
EIT-3	2136	12
Total	**10046**	**58**

Table E.8 Typical sizes of ETTs for the example

Part	Size in bytes (excluding Transport Stream packet header)	Size in Transport Stream packets
Channel ETT	3120	17
Event ETT-0	18720	102
Event ETT-1	18720	102
Event ETT-2	18720	102
Event ETT-3	18720	102
Total	**78000**	**425**

ANNEX F

(Informative)

AN OVERVIEW OF HUFFMAN-BASED TEXT COMPRESSION

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.

F1. 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.

F2. OVERVIEW OF CONTEXT-SENSITIVE HUFFMAN CODING

F2.1 Overview

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.

F2.2 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, assume that an alphabet contains the following twelve characters which occur a certain number of times in the sample database:

Table F.1 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 F.1 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.

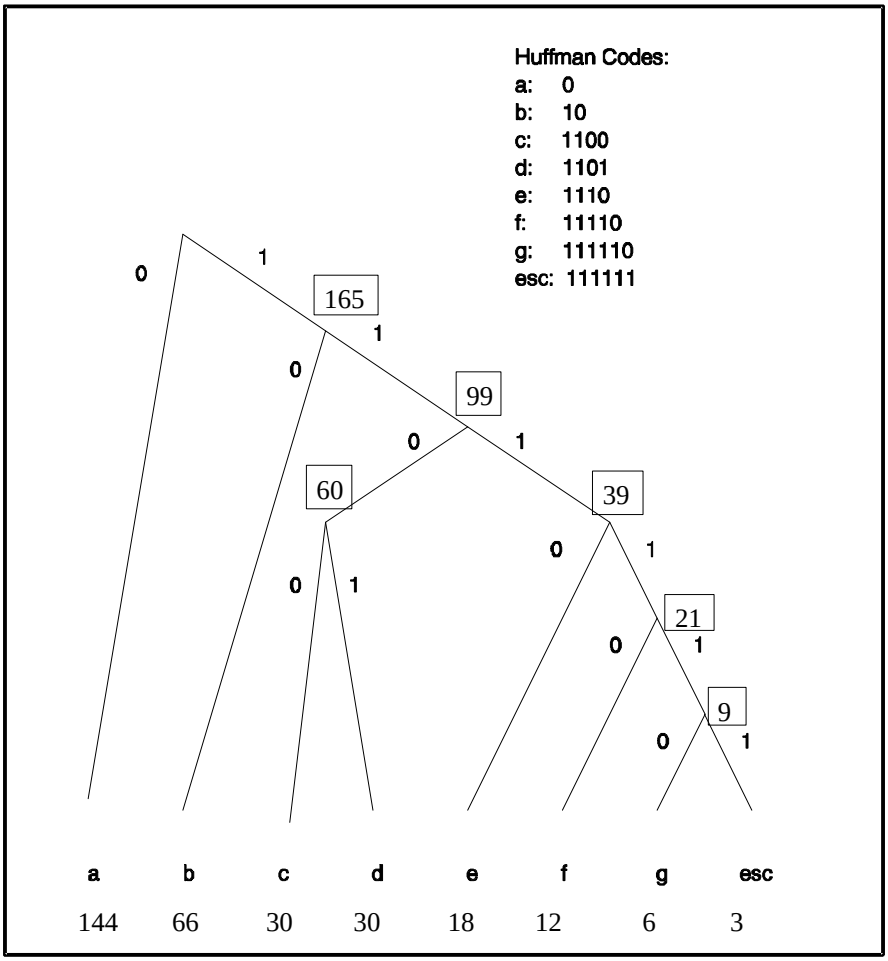

Figure F.1 Example Huffman Tree

F2.3 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 F.1. 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).

Table F.2 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)

F2.4 Encoding/Character Decoding Examples with 1st-order Huffman tables

As an example of using the Huffman table defined in Table C.4 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 F.2.

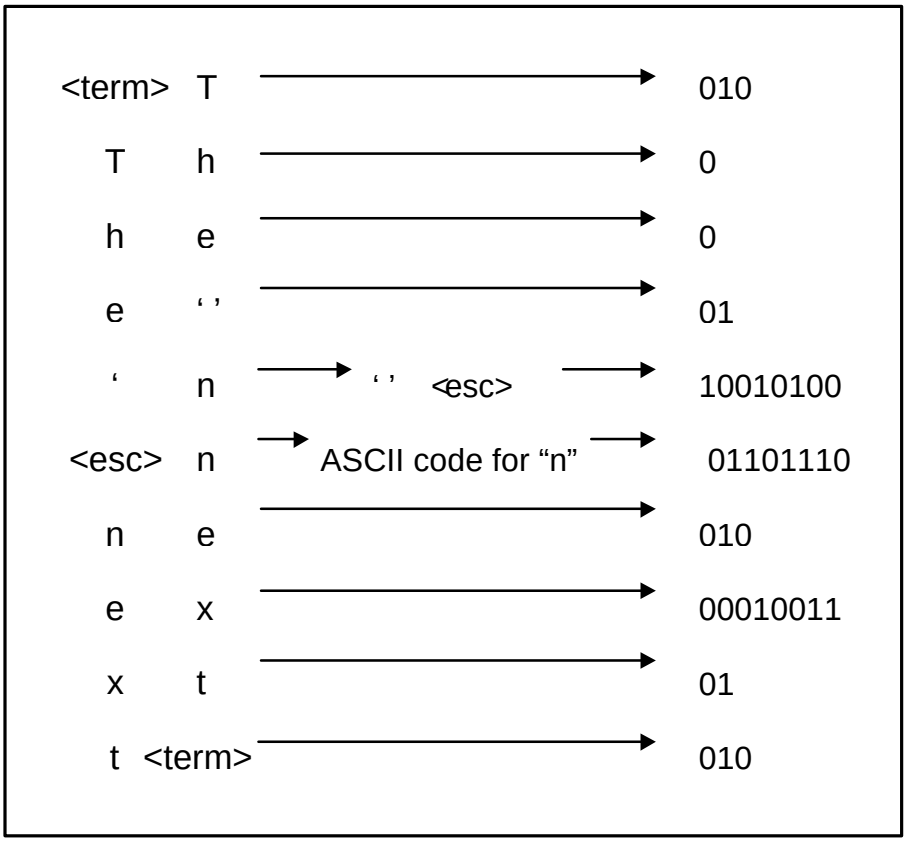

Figure F.2 Coding Example for the string “The next”

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 F.3 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.

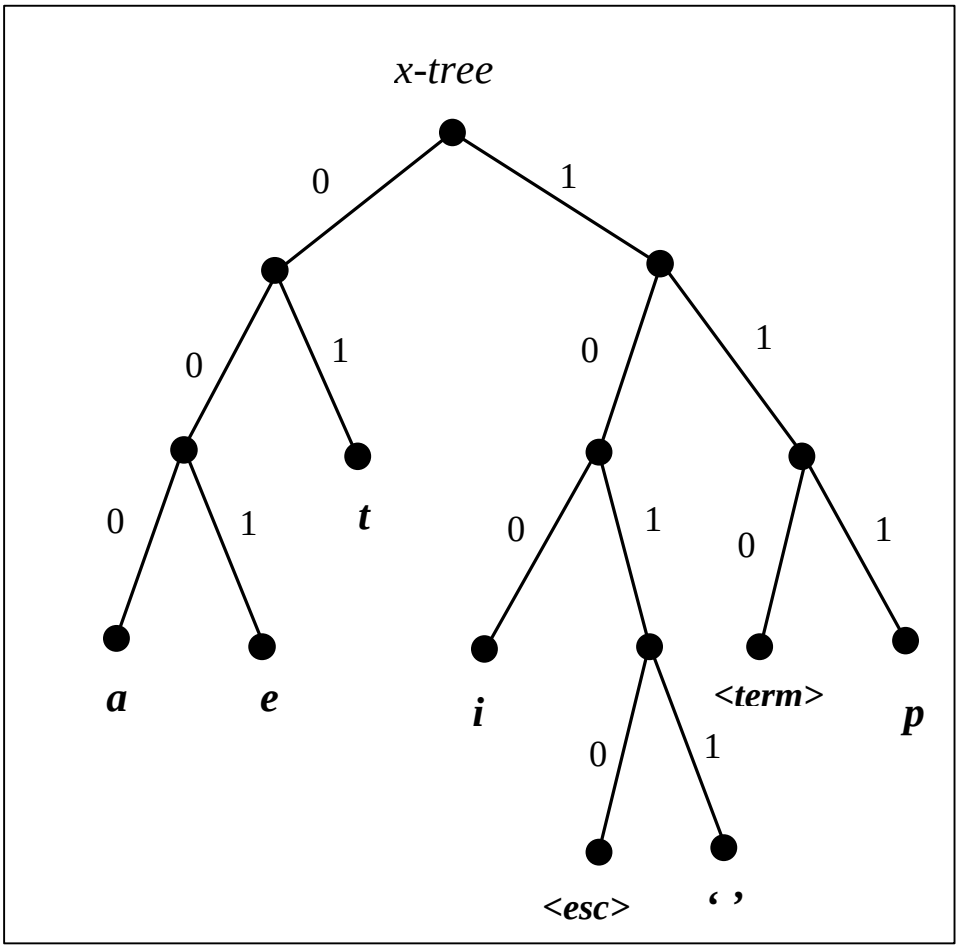

Figure F.3 Huffman tree for prior symbol “x”