INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.94

Amendment 1
(10/2000)

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

Ancillary digital services for television transmission

Service information for digital broadcasting in cable television systems

Amendment 1: Annex B – Service information delivered out of band in cable television systems

---

ITU-T Recommendation J.94 – Amendment 1

(Formerly CCITT Recommendation)

ITU-T J-SERIES RECOMMENDATIONS

TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER MULTIMEDIA SIGNALS

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

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

ITU-T Recommendation J.94

Service information for digital broadcasting in cable television systems

AMENDMENT 1

ANNEX B

Service information delivered out of band for digital cable television systems

Summary

This amendment provides the contents of Annex B that define a standard for Service Information (SI) on cable compatible with digital multiplex bit streams constructed in accordance with ITU-T H.222.0 | ISO/IEC 13818-1 (MPEG-2). It is designed to support "navigation devices" on cable. It defines the syntax and semantics for a standard set of tables providing the data necessary for such a device to discover and access digital and analogue services offered on cable.

This SI protocol is nominally delivered in a separate physical channel (i.e. out-of-band) outside the actual transport multiplexes containing the program content.

Source

Amendment 1 to ITU-T Recommendation J.94 was prepared by ITU-T Study Group 9 (1997-2000) and approved by the World Telecommunication Standardization Assembly (Montreal, 27 September – 6 October, 2000).

Keywords

Cable television, service information.

FOREWORD

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

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

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

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

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

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

CONTENTS

ITU-T Recommendation J.94

Service information for digital broadcasting in cable television systems

AMENDMENT 1

ANNEX B

Service information delivered out of band for digital cable television systems

B.1 Purpose, scope and organization

B.1.1 Purpose

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

B.1.2 Scope

This annex defines SI tables delivered via an out-of-band path to support service selection and navigation by digital cable set-top boxes and other "digital cable-ready" devices. The SI tables defined in this annex are formatted in accordance with the Program Specific Information (PSI) data structures defined in MPEG-2 Systems, ITU-T H.222.0 | ISO/IEC 13818-1.

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

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

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

SI data delivered out of band is transported in accordance with the Extended Channel interface defined in SCTE DVS 131r7 (1998) and SCTE DVS 216r4 (2000). To have access to the Extended Channel interface, the cable-ready device must act as a Host to a POD security module. The Extended Channel interface presents the needed SI data to the Host. This data can be used by the Host for channel navigation, construction of electronic program guides and other associated functions.

Figure B.1 is a high-level block diagram illustrating the POD module to Host interface via the Extended Channel interface. The Host is responsible for providing a standard receiver/QPSK demodulator function for the POD module. The choice of transport format of bits coming across from the receiver/QPSK demodulator to the POD module is by mutual agreement between the POD and the cable head-end equipment. The transport format of data travelling between the Host and

POD module on the Extended Channel interface conforms to standards defined in SCTE DVS 131r7 (1998) and SCTE DVS 216r4 (2000).

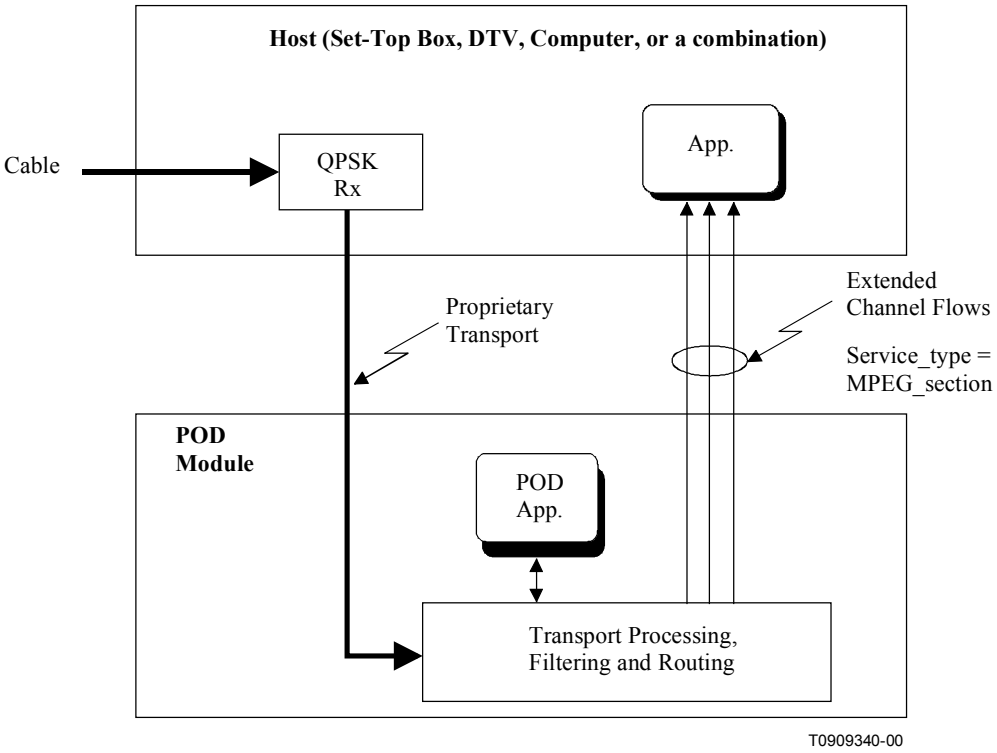

Figure B.1/J.94 – A framework for the extended channel service information stream

The POD module may perform various transport, filtering, and error checking/correction functions on the out-of-band data stream as depicted by the box labelled "Transport Processing, Filtering, and Routing." As described in SCTE DVS 216r4 (2000), the Host may request from the POD module to open one or several "flows" in which to receive PSI sections taken from the cable out-of-band data stream. Each flow is associated with a PID value, in accordance with MPEG-2 Transport Stream concepts.

Data flowing to the Host from the POD module that is associated with Service_type=MPEG_section is required to be in the form of MPEG PSI data structures. However, data delivered into the POD from cable out-of-band may or may not be organized in a Transport Stream compliant with ITU-T H.222.0 | ISO/IEC 13818-1. In other words, PID values associated with MPEG-2 tables on the Extended Channel interface *may or may not* correspond to MPEG-2 Transport Stream packet header PID values from the cable out of band.

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

B.1.3 Organization

This annex is organized as follows:

- **Clause B.1** – Provides this general introduction.
- **Clause B.2** – Lists applicable references.
- **Clause B.3** – Provides a list of definitions used in this annex.

- **Clause B.4 –** Provides a list of acronyms and abbreviations used in this annex.
- **Clause B.5 –** Describes the basic structure of sections.
- **Clause B.6 –** Describes formats of sections carried in the Base PID.[1]
- **Clause B.7 –** Explains descriptors applicable to the tables defined in this annex.
- **Clause B.8 –** Describes multilingual character string coding.
- **Annex B.A –** Defines profiles of choice for cable operator compliance with this annex.
- **Annex B.B –** Specifies packet rates for delivery of SI data.
- **Annex B.C –** Defines the standard Huffman tables used for text compression.
- **Appendix B.I –** Discusses recommendations for receiver implementations.
- **Appendix B.II –** Provides an overview of tables defined in this Service Information Annex B.
- **Appendix B.III –** Defines the daylight savings time control fields in the System Timetable.

B.2 References

Normative references

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

– ITU-T H.222.0 (2000) | ISO/IEC 13818-1:2000, *Information technology – Generic coding of moving pictures and associated audio information: Systems*.

– ITU-T H.262 (2000) | ISO/IEC 13818-2: 2000, *Information technology – Generic coding of moving pictures and associated audio information: Video*.

– ISO 639:1988, *Code for the representation of names of languages*.

– ISO 639-2:1998, *Codes for the representation of names of languages – Part 2: Alpha-3 code*.

– ISO/IEC 8859-1 to 10, *Information technology – 8-bit single-byte coded graphic character sets*.

– ISO/IEC 10646-1:2000, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*.

– ISO/IEC 13818-3:1998, *Information technology – Generic coding of moving pictures and associated audio – Part 3: Audio*.

Informative references

– ITU-T J.83 (1997), *Digital multi-programme systems for television, sound and data services for cable distribution*.

– SCTE DVS 031, *Digital Video Transmission Standard for Cable Television,* Rev.2, 29 May 1997.

---

[1] The Base PID is the PID associated with the "base" Service Information tables. In this protocol, the base_PID is fixed at 0x1FFC. Refer to Table B.2.

– SCTE DVS 097 (1997), *Program and System Information Protocol for Terrestrial Broadcast and Cable*.

– SCTE DVS 131r7 (1998), *Point of Deployment (POD) Module Interface*.

– SCTE DVS 208r6 (1999), *Cable Emergency Alert Message* (EIA-814).

– SCTE DVS 216r4 (2000), *POD Extended Channel Specification*.

Bibliography

– ATSC Standard A/52 (1995), *Digital Audio Compression (AC-3)*.

– ATSC Standard A/53 (1995), *ATSC Digital Television Standard*.

– EIA-708, *Specification for Advanced Television Closed Captioning (ATVCC),* Electronic Industry Association.

– EIA-752, *Transport of Transmission Signal Identifier (TSID) Using Extended Data Service (XDS)*.

– EIA 766, *U.S. Rating Region Table (RRT) and Content Advisory Descriptor for Transport of Content Advisory Information Using ATSC A/65 Program and System Information Protocol (PSIP)*.

B.3 Definitions

B.3.1 Compliance notation

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

B.3.2 Definition of terms

The following terms are used throughout this annex:

B.3.2.1 conditional access: The control and security of subscriber access to cable or broadcast services and events in the form of video, data and voice communications.

B.3.2.2 host: A device capable of supporting a POD module by implementing the interface protocol defined in SCTE DVS 131r7 (1998) and SCTE DVS 216r4 (2000). These protocols define the Extended Channel data path through which the SI tables defined in this annex are passed.

B.3.2.3 navigation: The process of selection and movement among analogue and digital services offered on the cable network. The service information tables defined in this protocol assist in the navigation process by providing physical service locations, channel names and numbers for user reference. Those tables supporting electronic program guides also assist the navigation process.

B.3.2.4 program element: A generic term for one of the elementary streams or other data streams that may be included in a program.

B.3.2.5 program: A collection of program elements. Program elements may be elementary streams. Program elements need not have any defined time base. Those that do have a common time base are intended for synchronized presentation. The term *program* is also used in the context of a "television program" such as a scheduled daily news broadcast. The distinction between the two usages should be understood by context.

B.3.2.6 region: As used in this annex, a region is a geographical area consisting of one or more countries.

B.3.2.7 section or table section: A data structure comprising a portion of an ITU-T H.222.0 | ISO/IEC 13818-1-defined table, such as the Program Association Table (PAT), Conditional Access Table (CAT), or Program Map Table (PMT). The term conforms to MPEG terminology. All sections begin with the table_ID and end with the CRC_32 field. Sections are carried in Transport Stream packets in which the starting point within a packet payload is indicated by the pointer_field mechanism defined in the ITU-T H.222.0 | ISO/IEC 13818-1 Systems document. The Network Information Table, for example, defines portions of several types of tables.

B.3.2.8 service: ITU-T H.222.0 | ISO/IEC 13818-1 uses the term *program* to refer to a collection of program elements with no regard to time. In this Service Information annex, the term *service* is used in this same context to denote a collection of elementary components. Usage of the term *service* clarifies certain discussions that also involve the notion of the term *program* in its traditional meaning – for example, in the statement, "A video service carries a series of programs." In a broader sense, *service* is also intended for multimedia services of video, voice and data, as these services become prevalent.

B.3.2.9 stream: An ordered series of bytes. The usual context for the term *stream* involves specification of a particular PID (such as the "Program Map PID stream"), in which case the term indicates a series of bytes extracted from the packet multiplex from packets with the indicated PID value.

B.3.3 Section and data structure syntax notation

This annex 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 annex are described using a C-like notational method employed in ITU-T H.222.0 | ISO/IEC 13818-1. Extensions to this method are described in the following clauses.

B.3.3.1 Field sizes

Each data structure is described in a table format wherein the size in bits of each variable within that section is listed in a column labelled "Bits." The column adjacent to the Bits column is labelled "Bytes" and indicates the size of the item in bytes. For convenience, several bits within a particular byte or multi-byte variable may be aggregated for the count. Table B.1 is an example:

Table B.1/J.94 – Field sizes example

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

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

B.4 Acronyms and abbreviations

The following acronyms and abbreviations are used within this annex:

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

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

B.5 Table structure

This clause describes details of the structure of MPEG-2 tables defined in this annex.

Tables and table sections defined in this Service Information annex are structured in the same manner used for carrying ITU-T H.222.0 | ISO/IEC 13818-1-defined PSI tables. The MPEG-defined 32-bit CRC is required.

B.5.1 Table ID ranges and values

Table B.2 defines table_ID ranges and values for tables defined in MPEG and in this annex.

Table B.2/J.94 – Table ID ranges and values for out-of-band transport

Table ID Value (hex)	Tables	PID	Reference
	ITU-T H.222.0 | ISO/IEC 13818-1 Sections:		
0x00	Program Association Table (PAT)	0	ITU-T H.222.0
0x01	Conditional Access Table (CAT)	1	ITU-T H.222.0
0x02	TS Program Map Table (PMT)	per PAT	ITU-T H.222.0
0x03-0x3F	[ISO Reserved]		
	User Private Sections:		
0x40-0x7F	[User Private for other systems]		
0x80-0xBF	[SCTE User Private]		
	Other Standards:		
0xC0-0xC1	[Used in other standards]		
	Service Information Tables:		
0xC2	Network Information Table (NIT)	0x1FFC	B.6.1
0xC3	Network Text Table (NTT)	0x1FFC	B.6.2
0xC4	Short-form Virtual Channel Table (S-VCT)	0x1FFC	B.6.3
0xC5	System Timetable (STT)	0x1FFC	B.6.4
0xC6	[Used in other standards]	–	–
0xC7	Master Guide Table (MGT)	0x1FFC	B.6.5

Table B.2/J.94 – Table ID ranges and values for out-of-band transport *(concluded)*

Table ID Value (hex)	Tables	PID	Reference
0xC8	Reserved	–	–
0xC9	Long-form Virtual Channel Table (L-VCT)	0x1FFC	B.6.6
0xCA	Rating Region Table (RRT)	0x1FFC	B.6.7
0xCB-0xD5	[Used in ATSC]	–	–
0xD6	Aggregate Event Information Table (AEIT)	per MGT	B.6.8
0xD7	Aggregate Extended Text Table (AETT)	per MGT	B.6.9
0xD8	Cable Emergency Alert Message	0x1FFC	SCTE DVS 208r6 (1999)
0xD9-0xFE	[Reserved for future use]	–	–

Table sections defined in this Service Information annex, and any created as user extensions to it are considered "private" with respect to ITU-T H.222.0 | ISO/IEC 13818-1. Table section types 0x80 through 0xBF are user-defined (outside the scope of this Service Information annex).

The maximum total length of any table section defined in this annex is 1024 bytes, except for the MGT, L-VCT, AEIT and AETT, each of which has a maximum total length of 4096 bytes. This total includes table_ID, CRC, and all fields contained within the specific table section.

B.5.2 Extensibility

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

1) **Reserved Fields**: Fields in this Service Information annex marked reserved are reserved for use either when revising this annex, or when another Recommendation is issued that builds upon this one. See B.5.4.

2) **Standard Table Types**: As indicated in Table B.2, table_ID values in the range 0xCE through 0xFE are reserved for use either when revising this Service Information annex, or when another Recommendation is issued that builds upon this one.[2]

3) **User Private Table Types**: As indicated in Table B.2, table_ID values in the range 0x80 through 0xBF are reserved for "user private" use. The format of user private tables carried in the Network PID shall conform to the syntax described in Table B.3.

4) **User Private Descriptors**: Privately defined descriptors may be placed at designated locations throughout the table sections described in this Service Information annex. Ownership of one or more user private descriptors is indicated by the presence of an MPEG registration_descriptor() preceding the descriptor(s).

[2] NOTE – Assignment of table_ID values in the 0xCE to 0xFE range requires coordination between ATSC and SCTE.

Table B.3/J.94 – Network private table section format

	Bits	Bytes	Format
Network_private_table section(){			
private_table_ID	8	1	uimsbf (0x80 <= table_ID <= 0xBF)
section_syntax_indicator	1	2	bslbf
Zero	1		bslbf
Reserved	2		bslbf
section_length	12		uimsbf
if (section_syntax_indicator==1) {			
table_extension	16	(2)	uimsbf
Reserved	2	(1)	bslbf
version_number	5		uimsbf
current_next_indicator	1		bslbf {next, current}
section_number	8	(1)	uimsbf
last_section_number	8	(1)	uimsbf
}			
Zero	3	1	bslbf
protocol_version	5		See B.5.4.1.
format_identifier	32	4	uimsbf
private_message_body()	N*8	N	
CRC_32	32	4	rpchof
}			

B.5.3 Reserved fields

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

zero: Indicates the bit or bit field shall be "0".

B.5.4 Private table section syntax

Table B.3 defines the syntax for user private table sections. The MPEG-defined CRC is required. Refer to ITU-T H.222.0 | ISO/IEC 13818-1 for definition of MPEG-standard fields.

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

B.5.4.1 Protocol version

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

B.5.4.2 Format identifier

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

B.5.4.3 Private Message Body

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

B.5.4.4 CRC

CRC_32: The 32-bit CRC value defined in ITU-T H.222.0 | ISO/IEC 13818-1 for PSI sections. The MPEG-2 CRC shall be checked in the POD, and only messages that pass the CRC check shall be forwarded to the Host. The Host shall not check the CRC.

B.6 Table section formats

The following clauses define the formats of table sections as they are delivered across the Extended Channel interface from POD module to Host.

B.6.1 Network Information Table

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

Table B.4 shows the format of the Network Information Table section.

Table B.4/J.94 – Network Information Table section format

	Bits	Bytes	Format
network_info_table_section(){			
table_ID	8	1	uimsbf value 0xC2
Zero	2	2	bslbf
Reserved	2		bslbf
section_length	12		uimsbf
Zero	3	1	bslbf
protocol_version	5		See B.5.4.1.
first_index	8	1	uimsbf range 1-255
number_of_records	8	1	uimsbf
transmission_medium	4	1	uimsbf
table_subtype	4		uimsbf (See Table B.5.)
for (i=0; i<number_of_records; i++) {			
if (table_subtype==CDS) {			
CDS_record()		((5))	
}			
if (table_subtype==MMS) {			
MMS_record()		((6))	
}			
Descriptors_count	8	(1)	uimsbf range 0-255
for (i=0; i<descriptors_count; i++) {			
descriptor()	*	((*))	Optional
}			
}			
for (i=0; i<N; i++) {			
descriptor()	*	(*)	Optional
}			
CRC_32	32	4	rpchof
}			

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

first_index: An 8-bit unsigned integer number in the range 1 to 255 that indicates the index of the first record to be defined in this table section. If more than one record is provided, the additional records define successive table entries following first_index. The value zero is illegal and shall not be specified.

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

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

table_subtype: A 4-bit value that defines the type of table delivered in the table section. One instance of a Network Information Table section can define entries within at most one type of table. The table_subtype parameter is defined in Table B.5.

Table B.5/J.94 – Network Information Table Subtype

table_subtype	Meaning
0	invalid
1	**CDS** – Carrier Definition Subtable
2	**MMS** – Modulation Mode Subtable
3-15	Reserved

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

B.6.1.1 Carrier Definition Subtable (CDS)

Table B.6 defines the structure of the CDS_record(). Each CDS defines a set of carrier frequencies. A full frequency plan table shall be constructed from one or more CDS_record() structures, each defining a starting frequency, a number of carriers, and a frequency spacing for carriers in this group.

The specified carrier represents the nominal centre of the spectral band for all modulation methods, including analogue. Carrier frequencies in the table thus represent the data carrier frequency for digital transmissions modulated using QAM or PSK.[3]

Each CDS_record represents a definition of N carriers. The first_index parameter reflects the index in a flat space between 1 and 255, representing the first carrier in the CDS_record. Starting from the first CDS_record defining carriers C_1, C_2, C_3, …, C_N, where N = number_of_carriers, the carrier index for C_I is equal to first_index + I – 1. If the table section includes more than one CDS_record(), the carrier index of the second CDS_record would be first_index plus the number of carriers defined in the first CDS_record(), namely, first_index + number_of_carriers. References to the Carrier Definition Subtable, such as the CDS_reference in the virtual_channel() of Table B.20, are to the carrier index (a carrier defined within a CDS_record()), between 1 and N, where N is normally much smaller than 255. These references are *not* to the index of a CDS_record() itself, which is sequenced from first_index and is not reset to 1 until it exceeds 255.

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

---

[3] Note that transmission systems using VSB modulation transmit spectra are not symmetrical about the carrier or pilot tone. Acquisition of a VSB-modulated signal involves computation of the pilot tone (or in analogue VSB, the picture carrier) location relative to the centre of the band. For example, for the ATSC Digital Television Standard (ASTC A/53), where the channel bandwidth is 6 MHz, the pilot tone is located 310 kHz above the lower edge of the channel, or 2.690 MHz below the specified centre of the band. Similarly, for analogue NTSC, the picture carrier is 1.25 MHz above the lower edge of the channel, or 1.75 MHz below the specified centre of the band.

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

Table B.6/J.94 – CDS record format

	Bits	Bytes	Format
CDS_record(){			
number_of_carriers	8	1	uimsbf
spacing_unit	1	2	bslbf (See Table B.7.)
Zero	1		bslbf
Frequency_spacing	14		uimsbf range 1-16 383 units of 10 or 125 kHz
Frequency_unit	1	2	bslbf (See Table B.8.)
first_carrier_frequency	15		uimsbf range 0-32 767 units of 10 or 125 kHz
}			

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

spacing_unit: A 1-bit field identifying the units for the frequency_spacing field. Table B.7 defines the coding for spacing_unit.

Table B.7/J.94 – Spacing Unit

spacing_unit	Meaning
0	10 kHz spacing
1	125 kHz spacing

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

frequency_unit: A 1-bit field identifying the units for the first_carrier_frequency field. Table B.8 defines the coding for frequency_unit.

Table B.8/J.94 – Frequency Unit

Frequency_unit	Meaning
0	10 kHz units
1	125 kHz units

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

B.6.1.2 Modulation Mode Subtable (MMS)

Table B.9 defines the structure of the MMS_record().

Table B.9/J.94 – MMS record format

	Bits	Bytes	Format
MMS_record(){			
transmission_system	4	1	uimsbf (See Table B.10.)
inner_coding_mode	4		uimsbf (See Table B.11.)
split_bitstream_mode	1	1	bslbf {no, yes}
Zero	2		bslbf
modulation_format	5		uimsbf (See Table B.12.)
Zero	4	4	bslbf
symbol_rate	28		uimsbf units: symbols per second
}			

transmission_system: A 4-bit field that identifies the transmission standard employed for the waveform defined by this MMS record. Table B.10 defines the coding for transmission_system.

Table B.10/J.94 – Transmission System

transmission_system	Meaning
0	**unknown** – The transmission system is not known.
1	Reserved (ETSI)
2	**ITU-T J.83 Annex B** – The transmission system conforms to the ITU North American standard specified in Annex B/J.83.
3	Defined for use in other systems
4	**ATSC** – The transmission system conforms to the ATSC Digital Television Standard.
5-15	Reserved (satellite)

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

Table B.11/J.94 – Inner Coding Mode

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

modulation_format: A 5-bit field that defines the basic modulation format for the carrier. Table B.12 defines the parameter.

Table B.12/J.94 – Modulation Format

modulation_format	Meaning
0	**unknown** – The modulation format is unknown.
1	**QPSK** – The modulation format is QPSK (Quadrature Phase Shift Keying).
2	**BPSK** – The modulation format is BPSK (Binary Phase Shift Keying).
3	**OQPSK** – The modulation format is offset QPSK.
4	**VSB 8** – The modulation format is 8-level VSB (Vestigial Sideband).
5	**VSB 16 –** The modulation format is 16-level VSB.
6	**QAM 16** – Modulation format 16-level Quadrature Amplitude Modulation (QAM).
7	**QAM 32** – 32-level QAM
8	**QAM 64** – 64-level QAM
9	**QAM 80** – 80-level QAM
10	**QAM 96** – 96-level QAM
11	**QAM 112** – 112-level QAM
12	**QAM 128** – 128-level QAM
13	**QAM 160** – 160-level QAM
14	**QAM 192** – 192-level QAM
15	**QAM 224** – 224-level QAM
16	**QAM 256** – 256-level QAM
17	**QAM 320** – 320-level QAM
18	**QAM 384** – 384-level QAM
19	**QAM 448** – 448-level QAM

Table B.12/J.94 – Modulation Format *(concluded)*

modulation_format	Meaning
20	**QAM 512** – 512-level QAM
21	**QAM 640** – 640-level QAM
22	**QAM 768** – 768-level QAM
23	**QAM 896** – 896-level QAM
24	**QAM 1024** – 1024-level QAM
25-31	Reserved

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

B.6.1.3 Descriptors Count

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

descriptor(): The table section may include at its end one or more structures of the form tag, length, data. The number of descriptors present is determined indirectly by processing the section_length field. Descriptors are defined in B.7.

B.6.2 Network Text Table

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

Table B.13 shows the format of the Network Text Table.

Table B.13/J.94 – Network Text Table section format

	Bits	Bytes	Format
network_text_table_section(){			
table_ID	8	1	uimsbf value 0xC3
Zero	2	2	bslbf
Reserved	2		bslbf
section_length	12		uimsbf
Zero	3	1	
protocol_version	5		See B.5.4.1.
ISO_639_language_code	24	3	Per ISO 639-2/B
transmission_medium	4	1	uimsbf
table_subtype	4		uimsbf (See Table B.14.)
if (table_subtype==SNS) {			
source_name_subtable()	*	(*)	
}			
for (i=0; i<N; i++) {			
descriptor()	*	(*)	Optional
}			
CRC_32	32	4	rpchof
}			

The Network Text Table carries Multilingual Text Strings, formatted as defined in B.8.2. Text Strings included in the Network Text Table shall not include format effectors (defined in B.8.1.2). If format effectors are present in a text block, the Host is expected to disregard them.

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

ISO_639_language_code: A 3-byte language code per ISO 639-2/B defining the language associated with the text carried in this Network Text Table. The ISO_639_language_code field contains a three-character code as specified by ISO 639-2/B. Each character is coded into 8 bits according to ISO 8859-1 (ISO Latin-1) and inserted, in order, into the 24-bit field. The value 0xFFFFFF shall be used in case the text is available in one language only. The value 0xFFFFFF shall represent a "wild card" match when filtering by language.

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

table_subtype: A 4-bit value that defines the type of table delivered in the table section. One instance of a Network Text Table section can define entries within at most one type of table. The table_subtype parameter is defined in Table B.14.

Table B.14/J.94 – Network Text Table Subtype

table_subtype	Meaning
0	Invalid
1-5	Reserved
6	**SNS** – Source Name Subtable
7-15	Reserved

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

The SNS can provide a textual name associated with each service defined in the Short-form Virtual Channel Table, by reference to its source_ID. The format of the source_name_subtable() is given in Table B.15.

Table B.15/J.94 – Source Name Subtable format

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

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

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

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

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

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

source_name(): A Multilingual Text String defining the name of the source or application, formatted according to the rules defined in B.8.1.

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

descriptor(): The table section may include, at its end, one or more structures of the form tag, length, data. The number of descriptors present is determined indirectly by processing the section_length field. Descriptors are defined in B.7.

B.6.3 Short-form Virtual Channel Table Section

The Short-form Virtual Channel Table section delivers portions of the Virtual Channel Map (VCM), the Defined Channels Map (DCM) and the Inverse Channel Map (ICM). Sections of the Short-form Virtual Channel Table shall be associated on the POD-Host interface with PID value 0x1FFC, the SI_base PID.

Table B.16 shows the syntax of the Short-form Virtual Channel Table section.

Table B.16/J.94 – Short-form Virtual Channel Table section format

	Bits	Bytes	Format
shortform_virtual_channel_table_section(){			
table_ID	8	1	uimsbf value 0xC4
Zero	2	2	bslbf
Reserved	2		bslbf
section_length	12		uimsbf
Zero	3	1	bslbf
protocol_version	5		See B.5.4.1.
transmission_medium	4	1	uimsbf
table_subtype	4		uimsbf (See Table B.17.)
VCT_ID	16	2	uimsbf
if (table_subtype==DCM) {			
DCM_structure()	*	(*)	
}			
if (table_subtype== VCM) {			
VCM_structure()	*	(*)	
}			

Table B.16/J.94 – Short-form Virtual Channel Table section format *(concluded)*

	Bits	Bytes	Format
if (table_subtype== ICM) {			
ICM_structure()	*	(*)	
}			
for (i=0; i<N; i++) {			
descriptor()	*	(*)	Optional
}			
CRC_32	32	4	rpchof
}			

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

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

table_subtype: A 4-bit field that indicates the map type being delivered in this S-VCT section. Three map types are currently defined: the Virtual Channel Map (VCM), the Defined Channels Map (DCM), and the Inverse Channel Map (ICM). Table B.17 defines table_subtype.

Table B.17/J.94 – S-VCT Table Subtypes

table_subtype	Meaning
0	**VCM** – Virtual Channel Map
1	**DCM** – Defined Channels Map
2	**ICM** – Inverse Channel Map
3-15	Reserved

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

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

B.6.3.1 Defined Channels Map

Table B.18 shows the format of the DCM_structure().

Table B.18/J.94 – DCM structure format

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

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

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

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

For example, if channels 2-90, 200-210, 400-410, 600-610, 800-810, and 999 were defined, and first_virtual_channel was zero, the DCM data sequence (in decimal) would be the following, where underlined numbers have the range_defined bit set: 2, 89, 109, 11, 127, 62, 11, 127, 62, 11, 127, 62, 11, 127, 61, 1.

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

channels_count: An unsigned 7-bit integer number, in the range 1 to 127, that indicates the number of defined (or undefined) channels in a group.

B.6.3.2 Virtual Channel Map

Table B.19 shows the format of the VCM_structure().

Table B.19/J.94 – VCM structure format

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

descriptors_included: A Boolean flag that indicates, when set, that one or more record-level descriptors are present in the table section. Record-level descriptors are those defined in Table B.20 following the "if (descriptors_included)" statement. When the flag is clear, the record-level descriptor block is absent. The descriptors_included flag is not applicable to the section level descriptors shown at the bottom of Table B.16.

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

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

activation_time: A 32-bit unsigned integer field providing the absolute second the virtual channel data carried in the table section will be valid, defined as the number of seconds since 0000 Hours UTC, January 6th, 1980. If the GPS_UTC_offset delivered in the System Timetable is zero, activation_time includes the correction for leap seconds. Otherwise, activation_time can be converted to UTC by subtracting the GPS_UTC_offset. If the activation_time is in the past, the data in the table section shall be considered valid immediately. An activation_time value of zero shall be used to indicate immediate activation.

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

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

virtual_channel(): Table B.20 defines the virtual_channel() record structure.

Table B.20/J.94 – Virtual channel record format

	Bits	Bytes	Format
virtual_channel(){			
Zero	4	2	bslbf
virtual_channel_number	12		uimsbf range 0-4095
application_virtual_channel	1	1	bslbf {no, yes}
Zero	1		bslbf
path_select	1		bslbf (See Table B.21.)
transport_type	1		bslbf (See Table B.22.)
channel_type	4		uimsbf (See Table B.23.)
if (application_virtual_channel) {			
application_ID	16	(2)	
} else {			
source_ID	16	(2)	
}			
if (transport_type==MPEG_2) {			
CDS_reference	8	((1))	uimsbf range 1-255
program_number	16	((2))	
MMS_reference	8	((1))	uimsbf range 1-255
} else { /* non-MPEG-2 */			
CDS_reference	8	((1))	uimsbf range 0-255
Scrambled	1	((1))	bslbf {no, yes}
Zero	3		bslbf
video_standard	4		uimsbf (See Table B.24.)
Zero	16	((2))	bslbf
}			
if (descriptors_included) {			

Table B.20/J.94 – Virtual channel record format *(concluded)*

	Bits	Bytes	Format
descriptors_count	8	(1)	uimsbf
for (i=0; i<descriptors_count; i++) {			
descriptor()	*	((*))	
}			
}			
}			

virtual_channel_number: An unsigned 12-bit integer, in the range 0 to 4095, reflecting the virtual channel whose definition is being provided by this virtual channel record, for the map identified by the VCT_ID field.

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

path_select: A 1-bit field that associates the virtual channel with a transmission path. For the cable transmission medium, path_select identifies which physical cable carries the Transport Stream associated with this virtual channel. Table B.21 defines path_select.

Table B.21/J.94 – Path Select

path_select	meaning
0	path 1
1	path 2

transport_type: A 1-bit field identifying the type of transport carried on this carrier as either being an MPEG-2 transport (value zero), or not (value one). Table B.22 defines the coding.

Table B.22/J.94 – Transport Type

transport_type	Meaning
0	MPEG-2 transport
1	non-MPEG-2 transport

channel_type: A 4-bit field defining the channel type. Table B.23 defines channel_type.

Table B.23/J.94 – Channel Type

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

application_ID: A 16-bit unsigned integer number, in the range 0x0001 to 0xFFFF, that identifies the application associated with the virtual channel, on a system-wide basis. One particular program guide application, for example, may look for a program carrying data in its native transmission format by searching through the Short-form Virtual Channel Table for a match on its assigned application_ID. In some cases, one application may be able to process streams associated with more than one application ID. The application ID may be used to distinguish content as well as format, for the benefit of processing within the application. The value zero for application_ID shall not be assigned; if specified in a Virtual Channel record, the value zero indicates "unknown" or "inapplicable" for the application_ID/source_ID field.

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

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

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

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

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

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

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

MMS_reference: An 8-bit unsigned integer value, in the range 0 to 255, that references an entry in the Modulation Mode Subtable (MMS). The value zero is illegal and shall not be specified. For digital waveforms, the MMS_reference associates the carrier with a digital modulation mode. For Host implementations that support only one set of modulation parameters, in systems in which one modulation method is used for all carriers, storage and processing of the MMS_reference is unnecessary. The MMS_reference field shall be disregarded for inactive channels.

video_standard: A 4-bit field that indicates the video standard associated with this non-Standard virtual channel. Table B.24 defines video_standard.

Table B.24/J.94 – Video Standard

video_standard	Meaning
0	**NTSC** – The video standard is NTSC.
1	**PAL 625** – The video standard is 625-line PAL.
2	**PAL 525** – The video standard is 525-line PAL.
3	**SECAM** – The video standard is SECAM.
4	**MAC** – The video standard is MAC.
5-15	Reserved

descriptor(): The table section may include, at its end, one or more structures of the form tag, length, data. The number of descriptors present is determined indirectly by processing the section_length field. Descriptors are defined in B.7.

B.6.3.3 Inverse Channel Map

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

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

Table B.25 describes the format of the ICM_structure().

Table B.25/J.94 – ICM structure format

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

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

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

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

virtual_channel_number: A 12-bit unsigned integer value, in the range 0 to 4095, that represents the virtual channel, in the Short-form Virtual Channel Table section (see Table B.16) given by VCT_ID, associated with the given source_ID through the virtual_channel() record (see Table B.20). A virtual_channel_number of zero indicates that the program given by source_ID is currently not carried in this Short-form Virtual Channel Table. Such placeholders are useful in the case where the existence of a certain program within a VCM may come and go.

B.6.4 System Timetable Section

The System Timetable is used to synchronize Hosts with accurate calendar time. The System Timetable shall be associated on the POD-Host interface with PID value 0x1FFC, the SI_base PID. Rate of transmission is typically once per minute, at second *00* of each minute.

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

Table B.26 shows the format of the System Timetable section.

Table B.26/J.94 – System Timetable section format

	Bits	Bytes	Format
system_time_table_section(){			
table_ID	8	1	uimsbf value 0xC5
Zero	2	2	bslbf
Reserved	2		bslbf
section_length	12		uimsbf
Zero	3	1	
protocol_version	5		See B.5.4.1.
Zero	8	1	bslbf
system_time	32	4	uimsbf
GPS_UTC_offset	8	1	uimsbf seconds
for (i=0; i<N; i++) {			
descriptor()	*	(*)	Optional
}			
CRC_32	32	4	rpchof
}			

table_ID: The table_ID of the System Timetable shall be 0xC5.

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

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

descriptor(): The table section may include at its end one or more structures of the form tag, length, data. The number of descriptors present is determined indirectly by processing the section_length field. Descriptors are defined in B.7.

B.6.5 Master Guide Table (MGT)

The Master Guide Table is used to indicate the location, size, and version of tables it references. The MGT shall be associated on the POD-Host interface with PID value 0x1FFC, the SI_base PID. The MGT syntax is shown in Table B.27. Syntax and semantics are identical to SCTE DVS 097, ATSC Standard A/65 (1997), except that additional table types are added to refer to all tables defined in this protocol.

Table B.27/J.94 – Master Guide Table section format

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

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

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

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

section_length: 12-bit field specifying the number of remaining bytes in this section immediately following the section_length field up to the end of the section. The value of the section_length shall be no larger than 4093.

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

NOTE – The map_ID may be considered to be an identifier for this instance of the Master Guide Table. In some applications, the POD module may receive multiple Master Guide Table sections corresponding to distinct channel maps. In this case, the POD module is responsible for accepting one MGT and discard the others. It may use the map_ID to filter them, using information provided outside the scope of this annex.

In every case, the Host will receive just one MGT across the POD to Host interface, and the map_ID parameter may be ignored.

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

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

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

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

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

tables_defined: This 16-bit unsigned integer in the range 0 to 65 535 represents the number of tables in the following loop.

table_type: This 16-bit unsigned integer specifies the type of table, based on Table B.28.

Table B.28/J.94 – MGT Table Types

table_type	Meaning
0x0000-0x0001	**[Assigned by ATSC]**
0x0002	**Long-form Virtual Channel Table** with current_next_indicator=**1**
0x0003	**Long-form Virtual Channel Table** with current_next_indicator=**0**
0x0004	**[Assigned by ATSC]**
0x0005-0x000F	**[Reserved]**
0x0010	**Short-form Virtual Channel Table-VCM Subtype**
0x0011	**Short-form Virtual Channel Table-DCM Subtype**
0x0012	**Short-form Virtual Channel Table-ICM Subtype**
0x0013-0x01F	**[Reserved]**
0x0020	**Network Information Table-CDS Table Subtype**
0x0021	**Network Information Table-MMS Table Subtype**
0x0021-0x02F	**[Reserved]**
0x0030	**Network Text Table-SNS Subtype**
0x0031-0x00FF	**[Reserved]**
0x0100-0x017F	**[Assigned by ATSC]**
0x0180-0x01FF	**[Reserved]**
0x0200-0x027F	**[Assigned by ATSC]**
0x028F-0x0300	**[Reserved]**
0x0301-0x03FF	**Rating Region Table with rating_region 1-255**
0x0400-0x0FFF	**[User private]**
0x1000-0x10FF	**Aggregate Event Information Table with MGT_tag 0 to 255**
0x1100-0x11FF	**Aggregate Extended Text Table with MGT_tag 0 to 255**
0x1200-0xFFFF	**[Reserved]**

For table types formatted with the MPEG short-form syntax, the revision_detection_descriptor() shall be used to indicate the section number and version. For example, table_type 0x0020 indicates the Network Information Table, CDS table subtype. One MGT reference to CDS would cover all sections of the delivered CDS.

MGT table types 0x1000 through 0x10FF reference AEIT instances with MGT_tag values 0x00 through 0xFF, respectively. Table types 0x1100 through 0x11FF reference AETT instances with MGT_tag values 0x00 through 0xFF, respectively. A table_type value of 0x1023 in the MGT, for example, refers to the instance of the AEIT with MGT_tag value 0x23.

Note that the choice of value of the MGT_tag is independent of the timeslot number. For example, the MGT_tag value used to deliver AEIT-0 may be zero or any other value up to 255.

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 L-VCT (current_next_indicator = 0) shall be one unit more (modulo 32) than the version number for the current L-VCT (current_next_indicator = 1).

number_bytes: This 32-bit unsigned integer field indicates the total number of bytes used for the table_type described in the loop. There may be more than one instance of the indicated table_type.

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).

descriptor(): The table section may include, at its end, one or more structures of the form tag, length, data. Descriptors are defined in B.7.

CRC_32: This is a 32-bit field that contains the CRC value to ensure a zero output from the registers in the decoder defined in Annex A of ITU-T H.222.0 | ISO/IEC 13818-1 "MPEG-2 Systems" after processing the entire Master Guide Table section.

B.6.5.1 Restrictions on PID values

Certain restrictions apply to the PID values specified in the MGT. These restrictions are necessary to ensure the Host can collect EPG data using a minimum number of concurrent flows on the Extended Channel.

- All AEIT and AETT table sections with common MGT_tag values shall share a common PID.
- AEIT-0, AETT-0, AEIT-1 and AETT-1 instances shall share a common PID value.[4]
- AEIT-2, AETT-2, AEIT-3 and AETT-3 instances shall be associated with a second separate PID value.
- EPG data describing events farther into the future may be associated with one or more PID values; the second PID value may be used for all or some of the AEIT/AETT-4 through AEIT/AETT-N instances (N < 256).

B.6.5.2 Restrictions on order of occurrence of table references

For all table references except AEIT and AETT, the order of appearance in the MGT of various table references is not specified or restricted. For AEIT and AETT references, the following restriction applies:

---

[4] Please refer to B.6.8 for definition of the AEIT-*n* and AETT-*n* notation convention used in this annex.

- The order of appearance of AEIT/AETT references in the MGT shall correspond to increasing time slot assignments.

NOTE – This rule allows a Host to know, before processing the AEIT/AETT data, which table instances correspond to near-term data and which correspond to data farther into the future. This information is useful if the Host has insufficient RAM to hold all data transmitted.

B.6.6 Long-form Virtual Channel Table

The Long-form Virtual Channel Table is carried in MPEG-2 table sections with table ID 0xC9, and conforms to the syntax and semantics of the MPEG-2 Private Section as described in 2.4.4.10 and 2.4.4.11 of ITU-T H.222.0 | ISO/IEC 13818-1. The Long-form Virtual Channel Table shall be associated on the POD-Host interface with PID value 0x1FFC, the SI_base PID.

The bit stream syntax for the Long-form Virtual Channel Table is shown in Table B.29.

Table B.29/J.94 – Long-form Virtual Channel Table section format

Syntax	Bits	Bytes	Format
longform_virtual_channel_table_section () {			
table_id	8	1	0xC9
section_syntax_indicator	1	2	'1'
private_indicator	1		'1'
Reserved	2		'11'
section_length	12		uimsbf
map_ID	16	2	uimsbf
Reserved	2	1	'11'
version_number	5		uimsbf
current_next_indicator	1		bslbf
section_number	8	1	uimsbf
last_section_number	8	1	uimsbf
protocol_version	8	1	uimsbf
num_channels_in_section	8	1	uimsbf
For(i=0; i<num_channels_in_section;i++) {			
short_name	7*16	(14)	unicode™BMP
reserved	4	(3)	'1111'
major_channel_number	10		uimsbf
minor_channel_number	10		uimsbf
modulation mode	8	(1)	uimsbf
carrier_frequency	32	(4)	uimsbf
channel_TSID	16	(2)	uimsbf
program_number	16	(2)	uimsbf
reserved	2	(2)	'11'
access_controlled	1		bslbf
hidden	1		bslbf
path_select	1		bslbf
out_of_band	1		bslbf
hide_guide	1		bslbf
reserved	3		'111'
service_type	6		uimsbf
source_id	16	(2)	uimsbf
reserved	6	(2)	'111111'
descriptors_length	10		uimsbf
for (i=0;i<N;i++) {			
descriptors()			
}			
}			

Table B.29/J.94 – Long-form Virtual Channel Table section format *(concluded)*

Syntax	Bits	Bytes	Format
reserved	6	2	'111111'
additional_descriptors_length	10		uimsbf
For(j=0; j<N;j++) {			
additional_descriptors()		var	
}			
CRC_32	32	4	rpchof
}			

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

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

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

section_length: This is a twelve-bit field that 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 4093.

map_ID: A 16-bit identifier for this Long-form Virtual Channel Table. In some applications, the POD module may receive multiple Long-form Virtual Channel Table sections corresponding to distinct channel maps. In this case, the POD may use the map_ID to distinguish them, using information provided outside the scope of this annex. In every case, the Host will receive just one L-VCT across the POD to Host interface, and the map_ID parameter may be ignored.

version_number: This 5-bit field is the version number of the Long-form Virtual Channel Table. For the current L-VCT (current_next_indicator = 1), the version number shall be incremented by 1 whenever the value of the current L-VCT changes. Upon reaching the value 31, it wraps around to 0. For the next L-VCT (current_next_indicator = 0), the version number shall be one unit more than that of the current L-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 1-bit indicator, which when set to '1' indicates that the Long-form 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 Long-form Virtual Channel Table shall be 0x00. It shall be incremented by one with each additional section in the Long-form 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 Long-form 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 Hosts 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 the L-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/IEC 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, minor_channel_number: These two 10-bit fields represent either a two-part or a one-part virtual channel number associated with the virtual channel being defined in this iteration of the "for" loop. One-part numbers range from 0 to 16 383. Two-part numbers consist of a major and a minor number part; the range of each is 0 to 999. The one- or two-part number acts as the user's reference number for the virtual channel. Some channels may be represented with a one-part number while others in the VCT are represented with two-part numbers.

The six MSBs of the major_channel_number field, when all 1, indicate that a one-part number is being specified. The value of the one-part number is given, in C syntax, by:

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

When the six MSBs of the major_channel_number field are not all 1, and the 10-bit major_channel_number field is less than 1000, two fields specify a two-part channel number. The value of the two-part number is given by major_channel_number and minor_channel_number.

Table B.30 summarizes the coding of the major_channel_number and minor_channel_number fields.

Table B.30/J.94 – Major and minor channel number field coding

	20-bit major/minor field (10-bit major + 10-bit minor)		**User channel number**
Two-part channel numbers	Major Number (10 bits)	Minor Number (10 bits)	Two-part user channel number
	000d	000d	0-0
	000d	001d	0-1
(1000 major numbers, each with 1000 minor numbers)	…	…	…
	000d	999d	0-999
	001d	000d	1-0
	…	…	…
	999d	999d	999-999
[Reserved]	000d to 999d	1000d-1023d	N/A
	1000-1007d	All values	N/A
One-part channel numbers	6-bit flag (set = 111111b)	One-Part Number (14 bits)	One-part user channel number
	Set	0d	0
(16 383 linear space numbers)	Set	1d	1
	Set	…	…
	Set	16383d	16383

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 annex in Table B.31. 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 SCTE. 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. The modulation_mode field shall be disregarded for inactive channels.

Table B.31/J.94 – Modulation modes

Modulation_mode	Meaning
0x00	[Reserved]
0x01	**analogue** – The virtual channel is modulated using standard analogue methods for analogue television.
0x02	**SCTE_mode_1** – The virtual channel has a symbol rate of 5.057 Msymb/s, transmitted in accordance with *Digital Transmission Standard for Cable Television*, Ref. SCTE DVS 031 (Mode 1). Typically, mode 1 will be used for 64-QAM.
0x03	**SCTE_mode_2** – The virtual channel has a symbol rate of 5.361 Msymb/s, transmitted in accordance with *Digital Transmission Standard for Cable Television*, Ref. SCTE DVS 031 (Mode 2). Typically, mode 2 will be used for 256-QAM.
0x04	**ATSC (8 VSB)** – The virtual channel uses the 8-VSB modulation method conforming to the *ATSC Digital Television Standard,* ATSC Standard A/53 (1995).
0x05	**ATSC (16 VSB)** – The virtual channel uses the 16-VSB modulation method conforming to the *ATSC Digital Television Standard,* ATSC Standard A/53 (1995).
0x06-0x7F	[Reserved for future use]
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 analogue or digital transmission associated with this virtual channel, in Hz. For QAM-modulated signals, the given carrier_frequency represents the location of the digitally modulated carrier; for VSB-modulated signals, the given carrier_frequency represents the location of the pilot tone; for analogue signals, it represents the frequency of the picture carrier. The carrier_frequency field shall be disregarded for inactive channels.

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. For inactive channels, channel_TSID represents the ID of the Transport Stream that will carry the service when it becomes active. The Host may use the channel_TSID to verify that a TS acquired at the referenced carrier frequency is actually the desired multiplex. Analogue 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.[5] A value of 0xFFFF for channel_TSID shall be specified for analogue 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 analogue services, a value of 0xFFFF shall be specified for program_number. For inactive channels (those not currently present in the Transport Stream), program_number shall be set to zero. This number shall **not** be interpreted as pointing to a Program Map Table entry.

---

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

access_controlled: A 1-bit Boolean flag, when set, indicates that 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. Whether a hidden channel and its event may appear in EPG displays depends on the state of the hide_guide bit.

path_select: A 1-bit field that associates the virtual channel with a transmission path. Two paths are available as defined in Table B.32. For the cable transmission medium, path_select identifies which of two physical input cables carries the Transport Stream associated with this virtual channel.

Table B.32/J.94 – Path Select

path_select	Meaning
0	path 1
1	path 2

out_of_band: A Boolean flag that indicates, when set, that the virtual channel defined in this iteration of the "for" loop is carried on the cable on the Extended Channel interface carrying the tables defined in this protocol. When clear, the virtual channel is carried within a standard tuned multiplex at that frequency.

NOTE – A virtual channel carried on the out-of-band channel may be acquired by opening a flow between Host and POD to capture the PAT on PID 0. Processing the PAT will determine the PID associated with that service's PMT. Then, a flow can be opened to capture and process the PMT to determine the PIDs associated with elementary stream components of the service. Finally, a flow associated with the service's PID can be opened to capture service-related data.

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

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

service_type: A 6-bit enumerated type field that identifies the type of service carried in this virtual channel, based on Table B.33.

Table B.33/J.94 – Service Types

service_type	Meaning
0x00	[Reserved]
0x01	**analogue_television** – The virtual channel carries analogue 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 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.

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 ITU-T H.222.0 | ISO/IEC 13818-1 "MPEG-2 Systems" after processing the entire Long-form Virtual Channel Table section.

For inactive channels, the short_name, major_channel_number, and minor_channel_number fields reflect the name and channel number of the inactive channel, and may be used in construction of the program guide. The source_ID for inactive channels is used, as it is for active channels, to link the virtual channel to the program guide data. The service_type field and attribute flags reflect the characteristics of the channel that will be valid when it is active.

B.6.7 Rating Region Table (RRT)

The Rating Region Table carries rating information for multiple geographical regions. The RRT shall be associated on the POD-Host interface with PID value 0x1FFC, the SI_base PID.

Transmission of the RRT is required whenever any Transport Stream carries a service that includes a content_advisory_descriptor() in one of its Program Map Tables, or if a content_advisory_descriptor() appears in any transmitted AEIT. An instance of the RRT for each region referenced in any content_advisory_descriptor() shall be transmitted.

Each RRT instance, identified by rating_region (the eight 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.

Table B.34 describes the Rating Region Table.

Table B.34/J.94 – Rating Region Table section format

	Bits	Bytes	Format
rating_region_table_section () {			
table_ID	8	1	0xCA
section_syntax_indicator	1	2	'1'
private_indicator	1		'1'
Reserved	2		'11'
section_length	12		uimsbf
table_ID_extension{			
Reserved	8	1	0xFF
rating_region	8	1	uimsbf
}			
Reserved	2	1	'11'
version_number	5		uimsbf
current_next_indicator	1		'1'
section_number	8	1	uimsbf
last_section_number	8	1	uimsbf

Table B.34/J.94 – Rating Region Table section format *(concluded)*

	Bits	Bytes	Format
protocol_version	8	1	uimsbf
rating_region_name_length	8	1	uimsbf
rating_region_name_text()	var		
dimensions_defined	8	1	uimsbf
for(i=0; i<dimensions_defined;i++) {			
dimension_name_length	8	1	uimsbf
dimension_name_text()	var		
Reserved	3	1	'111'
graduated_scale	1		bslbf
values_defined	4		uimsbf
for (j=0;j<values_defined;j ++) {			
abbrev_rating_value_length	8	1	uimsbf
abbrev_rating_value_ text()	var		
rating_value_length	8	1	uimsbf
rating_value_ text()	var		
}			
}			
Reserved	6	2	'111111'
descriptors_length	10		uimsbf
for (i=0;i<N;i++) {			
descriptors()	var		
}			
CRC_32	32	4	rpchof
}			

table_ID: The table_ID of the Rating Region Table (RRT) shall be 0xCA.

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

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

section_length: 12-bit field specifying the number of remaining bytes in this section immediately following the section_length field up to the end of the section. The value of the section_length shall be no larger than 1021.

rating_region: An 8-bit unsigned integer number that defines the rating region to be associated with the text in this rating_region_table_section(). The value of this field is the identifier of this rating region, and thus this field may be used by the other tables (e.g. MGT) for referring to a specific rating region table. Values of rating_region are defined in Table B.35.

Table B.35/J.94 – 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. The rating_region_name_text() shall be formatted according to the Multiple String Structure (see B.8.2). 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". The dimension_name_text() shall be formatted according to the Multiple String Structure (see B.8.2). 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. "". The abbrev_rating_value_text() shall be formatted according to the Multiple String Structure (see B.8.2). 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. "". The rating_value_text() shall be formatted according to the Multiple String Structure (see B.8.2). 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 ITU-T H.222.0 | ISO/IEC 13818-1 "MPEG-2 Systems" after processing the entire Rating Region Table section.

B.6.8 Aggregate Event Information Tables (AEIT)

The Aggregate Event Information Table delivers event title and schedule information that may be used to support an Electronic Program Guide application. The transmission format allows instances of table sections for different time periods to be associated with common PID values. For use on the Extended Channel (out-of-band), reduction of the total number of PID values in use for SI data is important, because the POD module can typically support only a small number of concurrent data flows (each associated with one PID value).

Each AEIT instance describes event data for one three-hour time period. The start time for any AEIT is constrained to be one of the following eight UTC times: 00:00 (midnight), 03:00, 06:00, 09:00, 12:00 (noon), 15:00, 18:00, and 21:00.

The notation AEIT-*n* refers to the AEIT corresponding to timeslot *n*. Value 0 for *n* indicates the current timeslot, value 1 the next timeslot, etc. The same notational methods apply to AETT.

Except for AEIT-0, each AEIT instance shall include event data only for those events actually starting within the covered time period.[6] AEIT-0 shall also include event data for all events starting in a prior timeslot but continuing into the current timeslot. In addition, if the VCT entry for a particular source ID includes a time_shifted_service_descriptor(), AEIT-0 shall describe event data for active events on any channels referenced through the time_shifted_service_descriptor().

ETMs for events described in AEIT-0 shall be provided in AETT-0 on the PID associated with AEIT-0 until they are no longer referenced by AEIT-0.

Table B.36 defines the syntax of the Aggregate Event Information Table.

Table B.36/J.94 – Aggregate Event Information Table format

Syntax	Bits	Bytes	Format
aggregate_event_information_table_section () {			
table_ID	8	1	0xD6
section_syntax_indicator	1	2	'1'
private_indicator	1		'1'
Reserved	2		'11'
section_length	12		uimsbf
AEIT_subtype	8	1	uimsbf
MGT_tag	8	1	uimsbf
Reserved	2		'11'
version_number	5		uimsbf
current_next_indicator	1		'1'
section_number	8	1	uimsbf
last_section_number	8	1	uimsbf
if (AEIT_subtype == 0) {			
num_sources_in_section	8	1	uimsbf
for (j = 0; j< num_sources_in_section;j++) {			
source_ID	16	(2)	uimsbf
Num_events	8	(1)	uimsbf
for (j = 0; j< num_events;j++) {			
reserved	2	((2))	'11'
event_ID	14		uimsbf
start_time	32	((4))	uimsbf

[6] Although AEIT is similar in structure to the EIT in ATSC A/65, its properties differ from EIT in this regard.

Table B.36/J.94 – Aggregate Event Information Table format *(concluded)*

Syntax	Bits	Bytes	Format
reserved	2	((3))	'11'
ETM_present	2		bslbf
duration	20		uimsbf
title_length	8	((1))	uimsbf
title_text()	var		
reserved	4	((2))	'1111'
descriptors_length	12		
for (i=0;i<N;i++) {			
descriptor()			
}			
}			
}			
else			
reserved	n*8	n	
CRC_32	32	4	rpchof
}			

table_ID: The table_ID of the Aggregate Event Information Table shall be 0xD6.

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.

AEIT_subtype: This 8-bit field identifies the subtype of the AEIT. In the current protocol, only table subtype value 0x00 is defined. Host devices shall discard instances of the aggregate_event_information_table_section() in which an unknown AEIT_subtype is specified (currently, any value other than zero).

MGT_tag: An 8-bit field that ties this AEIT instance to the corresponding table_type in the MGT and to an AETT instance with the same value. The MGT_tag value for an AEIT instance for a given timeslot shall be one higher (modulo 256) than the instance for the preceding time period.

version_number: This 5-bit field is the version number of the AEIT instance. An instance is identified by the MGT_tag. The version number shall be incremented by 1 modulo 32 when any field in the AEIT instance changes. 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 AEIT sections; the AEIT 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.

num_sources_in_section: This 8-bit field gives the number of iterations of the "for" loop describing program schedule data.

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

num_events: Indicates the number of events to follow associated with the program source identified by source_ID. Value 0 indicates no events are defined for this source for the time period covered by the AEIT instance.

event_ID: This 14-bit field specifies the identification number of the event described. This number serves as a part of the event ETM_ID (identifier for event Extended Text Message). An assigned event_ID shall be unique at least within the scope of the instance of the AEIT in which it appears. Accordingly, as an example, the event associated with event_ID 0x0123 in AEIT-m shall be considered to be an event distinct from event_ID 0x0123 in AEIT-n, when m is not equal to n.

start_time: A 32-bit unsigned integer quantity representing the start time of this event as the number of seconds since 0000 Hours UTC, January 6th, 1980. If the GPS_UTC_offset delivered in the System Timetable is zero, start_time includes the correction for leap seconds. Otherwise, start_time can be converted to UTC by subtracting the GPS_UTC_offset.

ETM_present: This 2-bit field indicates the existence of an Extended Text Message (ETM) based on Table B.37.

Table B.37/J.94 – ETM_present

ETM_present	Meaning
0x00	No ETM
0x01	ETM present on this out-of-band Extended Channel
0x02-0x03	[Reserved for future use]

duration: Duration of this event in seconds.

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

title_text(): The event title in the format of a Multiple String Structure. title_text() shall be formatted according to the Multiple String Structure (see B.8.2).

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 ITU-T H.222.0 | ISO/IEC 13818-1 "MPEG-2 Systems" after processing the entire Aggregate Event Information Table section.

B.6.9 Aggregate Extended Text Tables (AETT)

The Aggregate Extended Text Table contains Extended Text Messages (ETM), which are used to provide detailed descriptions of events. An ETM is a multiple string data structure. 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 the allocated display space.

The transmission format of the AETT and its affiliated AEIT allows instances of AEIT/AETT table sections for different time slots to be associated with common PID values.

AETT-*n* shall be associated with the same PID value as AEIT-*n* for a given value of *n*.

The Aggregate Extended Text Table is carried in an MPEG-2 private section with table_ID 0xD7. An instance of the AETT includes one or more ETMs. Each description is distinguished by its unique 32-bit ETM_ID.

Table B.38 defines the syntax of the Aggregate Extended Text Table.

Table B.38/J.94 – Aggregate Extended Text Table format

Syntax	Bits	Bytes	Format
aggregate_extended_text_table_section () {			
table_ID	8	1	0xD7
section_syntax_indicator	1	2	'1'
private_indicator	1		'1'
Reserved	2		'11'
section_length	12		uimsbf
AETT_subtype	8	1	uimsbf
MGT_tag	8	1	uimsbf
Reserved	2	1	'11'
version_number	5		uimsbf
current_next_indicator	1		'1'
section_number	8	1	uimsbf
last_section_number	8	1	uimsbf
if (AETT_subtype == 0) {			
num_blocks_in_section	8	1	uimsbf
for (j = 0; j< num_blocks_in_section;j++) {			
ETM_ID	32	(4)	uimsbf
reserved	4	(2)	'1111'
extended_text_length	12		uimsbf
extended_text_message()	var		
}			
}			
Else			
reserved	n*8	n	
CRC_32	32	4	rpchof
}			

table_ID: The table_ID of the Aggregate Extended Text Table shall be 0xD7.

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.

AETT_subtype: This 8-bit field identifies the subtype of the AETT. In the current protocol, only table subtype value 0x00 is defined. Host devices shall discard instances of the aggregate_extended_text_table_section() in which an unknown AETT_subtype is specified (currently, any value other than zero).

MGT_tag: An 8-bit field that ties this AETT instance to the corresponding table_type in the MGT and to an AEIT instance with the same value. The MGT_tag value for an AETT instance for a given time period shall be one higher (modulo 256) than the instance for the preceding time period.

version_number: This 5-bit field is the version number of the AETT instance. An instance is uniquely identified by its MGT_tag. The version number shall be incremented by 1 modulo 32 when any field in the AETT instance changes. 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 AETT sections; the AETT 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.

num_blocks_in_section: This 8-bit field gives the number of iterations of the "for" loop describing ETM data.

ETM_ID: Unique 32-bit identifier of this Extended Text Message. This identifier is assigned by the rule shown in Table B.39.

Table B.39/J.94 – ETM ID

	MSB		**LSB**
Bit	**31 16**	**15 2**	**1 0**
event ETM_ID	source_ID	event_ID	1 0

extended_text_length: A 12-bit unsigned integer number that represents the length, in bytes, of the extended_text_message() field directly following.

extended_text_message(): The extended text message in the format of a Multiple String Structure (see B.8.2).

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 ITU-T H.222.0 | ISO/IEC 13818-1 "MPEG-2 Systems" after processing the entire Transport Stream AETT section.

B.7 Descriptors

This clause defines descriptors applicable for use with various table sections defined in this annex.

B.7.1 Descriptor usage

Table B.40 lists all descriptors, their tag numbers and associated table sections applicable to out-of-band SI transport. Asterisks mark the tables where the descriptors may appear. The range of descriptor tags defined or reserved by MPEG-2 includes those with tag values 0x3F or below, plus 0xFF.

Table B.40/J.94 – Descriptor usage

Descriptor name	**Tag**	**Table section**								
		PMT	NIT	NTT	S-VCT	STT	MGT	L-VCT	RRT	AEIT
Stuffing descriptor	0x80	*	*	*	*	*	*	*	*	*
AC-3 audio descriptor	0x81	*								*
Caption service descriptor	0x86	*								*
Content advisory descriptor	0x87	*								*
Revision detection descriptor	0x93		*	*	*					
Two part channel no. descriptor	0x94				*					
Channel properties descriptor	0x95				*					

Table B.40/J.94 – Descriptor usage *(concluded)*

Descriptor name	Tag	Table section								
		PMT	NIT	NTT	S-VCT	STT	MGT	L-VCT	RRT	AEIT
Daylight savings time descriptor	0x96					*				
Extended channel name descriptor	0xA0							*		
Time shifted service descriptor	0xA2							*		
Component name descriptor	0xA3	*								
User private descriptors	0xC0-0xFF		*	*	*	*	*	*	*	*

B.7.2 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 this annex.

B.7.3 AC-3 audio descriptor

The AC-3 audio descriptor, as defined in ATSC Standard A/52 (1995), and constrained in Annex B of ATSC Standard A/53 (1995), may be used in the PMT and/or in AEITs.

B.7.4 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 Caption Service Descriptor is shown in Table B.41.

Table B.41/J.94 – Caption Service Descriptor format

Syntax	Bits	Bytes	Format
caption_service_descriptor() {			
descriptor_tag	8	1	0x86
descriptor_length	8	1	uimsbf
Reserved	3	1	'111'
number_of_services	5		uimsbf
for (i=0;i<number_of_services;i++) {			
Language	8*3	(3)	uimsbf
cc_type	1	(1)	bslbf
Reserved	1		'1'
if (cc_type==line21) {			
reserved	5		'11111'
line21_field	1		bslbf
}			

Table B.41/J.94 – Caption Service Descriptor format *(concluded)*

Syntax	Bits	Bytes	Format
Else			
caption_service_number	6		uimsbf
easy_reader	1	(2)	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 Aggregate Event Information Table.

Each iteration of the "for" loop defines one closed caption service present as a sub-stream within the 9600 bit/s closed captioning stream. Each iteration provides the sub-stream's language, attributes, and (for advanced captions) the associated Service Number reference. Refer to EIA-708 Specification for Advanced Television Closed Captioning (ATVCC), 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 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 EIA-708 Specification for Advanced Television Closed Captioning (ATVCC). When the flag is clear, a line-21 closed caption service is present. For line 21 closed captions, the line21_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 EIA-708 Specification for Advanced Television Closed Captioning (ATVCC) 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 EIA-708 Specification for Advanced Television Closed Captioning (ATVCC).

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 EIA-708 Specification for Advanced Television Closed Captioning (ATVCC), 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.

B.7.5 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 B.42.

Table B.42/J.94 – Content Advisory Descriptor format

Syntax	Bits	Bytes	Format
content_advisory_descriptor() {			
descriptor_tag	8	1	0x87
descriptor_length	8	1	uimsbf
Reserved	2	1	'11'
rating_region_count	6		
for (i=0; i<rating_region_count; i++) {			
rating_region	8	1	uimsbf
rated_dimensions	8	1	uimsbf
for (j=0;j<rated_dimensions;j++) {			
rating_dimension_j	8	1	uimsbf
reserved	4	1	'1111'
rating_value	4		uimsbf
}			
rating_description_length	8	1	uimsbf
rating_description_text()	var		
}			
}			

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

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

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

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

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

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

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

rating_description_length: An 8-bit unsigned integer value in the range 0 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 B.8.2). 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().

The program source provider shall be the responsible party for insertion of correct content_advisory_descriptors in the Program Map Table (PMT). Also, the content_advisory_descriptors may be included in Aggregate Event Information Tables. If content_ advisory_descriptors are available both in AEIT and PMT, the PMT should be used first, then the AEITs.

B.7.6 Revision detection descriptor

The revision_detection_descriptor() is used to indicate whether new information is contained in the table section in which it appears.

Table B.43 describes the revision_detection_descriptor. This descriptor should be the first descriptor in the list to limit processing overhead.

Table B.43/J.94 – Revision Detection Descriptor format

	Bits	Bytes	Format
revision_detection_descriptor(){			
descriptor_tag	8	1	uimsbf value 0x93
descriptor_length	8	1	uimsbf
reserved	3	1	bslbf
table_version_number	5		uimsbf range 0-31
section_number	8	1	uimsbf range 0-255
last_section_number	8	1	uimsbf range 0-255
}			

descriptor_tag: An 8-bit unsigned integer number that identifies the descriptor as a revision_detection_descriptor(). The tag shall have the value 0x93.

descriptor_length: An 8-bit unsigned integer number that indicates the number of bytes to follow in the descriptor. At present, just three bytes are defined, but the length field shall be processed to allow new data to be added to the descriptor in the future.

table_version_number: This 5-bit unsigned integer in the range 0 to 31 identifies the version of the current table. This integer applies only to the table (or the section of it) currently transmitted. Other types of tables may have different version numbers. To indicate a change in a specific table, this integer is incremented by 1 modulo 32.

section_number: An 8-bit unsigned integer in the range 0 to 255 that identifies the current table section. Version numbers for all sections of a table must be the same. Note that section_number = 0 indicates the first section of a table.

last_section_number: An 8-bit unsigned integer in the range 0 to 255 that identifies the number of sections in a table. Note that if the last_section_number = 0, then there is only one section in this table.

B.7.7 Two-Part channel number descriptor

Table B.44 describes the two_part_channel_number_descriptor(). This descriptor may appear in the virtual_channel() record, contained in the VCM_structure; within the Short-form Virtual Channel Table section. The descriptor may be used by compatible Hosts to associate a two-part user channel number with any virtual channel. Some channels may have a two_part_channel_number_descriptor() while others do not.

NOTE – For the L-VCT, the 10-bit major/minor number fields can be coded to represent a one-part channel number. The one-part representation is not needed for the major/minor number fields in the two_part_channel_number_descriptor() in the S-VCT, because there is already a 12-bit one-part number on each channel in S-VCT. It would cause confusion to allow a second one-part number to be associated with a channel defined in S-VCT.

Table B.44/J.94 – Two-part Channel Number Descriptor format

	Bits	Bytes	Format
two_part_channel_number_descriptor(){			
descriptor_tag	8	1	uimsbf value 0x94
descriptor_length	8	1	uimsbf
Reserved	6	2	bslbf
major_channel_number	10		uimsbf range 0-999
Reserved	6	2	bslbf
minor_channel_number	10		uimsbf range 0-999
}			

descriptor_tag: An 8-bit unsigned integer number that identifies the descriptor as a two_part_channel_number_descriptor(). The tag shall have the value 0x94.

descriptor_length: An 8-bit unsigned integer number that indicates the number of bytes to follow in the descriptor. At present, just four bytes are defined, but the length field shall be processed to allow new data to be added to the descriptor in the future.

major_channel_number: A 10-bit unsigned integer in the range 0 to 999 that identifies the "major" channel number to be associated with the virtual channel.

minor_channel_number: A 10-bit unsigned integer in the range 0 to 999 that identifies the "minor" channel number to be associated with the virtual channel.

Hosts that support two-part channel numbering must support this descriptor. It is only mandatory for this descriptor to be sent in the instance where system support of two-part channel numbering is required. This means for virtual_channel() records where the Host does not receive the two-part channel number descriptor, that the Host is expected to use the virtual_channel_number described in the virtual_channel() record in B.6.3.2.

B.7.8 Channel properties descriptor

The channel_properties_descriptor() is defined to allow both forms of VCTs (S-VCT and L-VCT) carrying the same properties. Table B.45 describes the syntax for this descriptor. The descriptor may appear within a virtual_channel() record in the Short-form Virtual Channel Table.

Table B.45/J.94 – Channel Properties Descriptor format

	Bits	Bytes	Format
channel_properties_descriptor(){			
descriptor_tag	8	1	uimsbf value 0x95
descriptor_length	8	1	uimsbf
channel_TSID	16	2	uimsbf
reserved	6	1	'111111'
out_of_band_channel	1		uimsbf
access_controlled	1		uimbsf
hide_guide	1	1	bslbf
reserved	1		'1'
service_type	6		uimsbf
}			

descriptor_tag: An 8-bit unsigned integer number that identifies the descriptor as a channel_properties_descriptor(). The tag shall have the value 0x95.

descriptor_length: An 8-bit unsigned integer number that indicates the number of bytes to follow in the descriptor. At present, just four bytes are defined, but the length field shall be processed to allow new data to be added to the descriptor in the future.

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. For inactive channels, channel_TSID represents the ID of the Transport Stream that will carry the service when it becomes active. The Host may use the channel_TSID to verify that a TS acquired at the referenced carrier frequency is actually the desired multiplex. Analogue signals may have a TSID that is different from any MPEG-2 Transport Stream identifier, that is, it shall be truly unique if present. A value of 0xFFFF for channel_TSID shall be specified for situations where a valid TSID is not known (reserved as a wildcard capability).

out_of_band: A Boolean flag that indicates, when set, that the virtual channel associated with this descriptor is carried on the cable on the Extended Channel interface carrying the tables defined in this protocol. When clear, the virtual channel is carried within a standard tuned multiplex at that frequency.

access_controlled: A Boolean flag that indicates, when set, that events associated with this virtual channel may be access controlled. When the flag is zero, event access is not restricted.

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

service_type: A 6-bit enumerated type field that identifies the type of service carried in this virtual channel. Service type is coded according to Table B.33.

Hosts may use this descriptor to become aware of aspects of the channel. In the case where this descriptor is not received, the Host must tune the channel and self-discover these aspects of the channel. For example, if this descriptor is not sent, and the channel is access controlled, the Host must determine when it can obtain access permission (the same as if that bit in the descriptor were set). Similar rules can be applied for service type and channel_TSID.

B.7.9 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 B.46.

Table B.46/J.94 – Extended Channel Name Descriptor format

Syntax	Bits	Bytes	Format
extended_channel_name_descriptor() {			
descriptor_tag	8	1	0xA0
descriptor_length	8	1	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 B.8.2).

B.7.10 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.

NOTE – For the L-VCT, the 10-bit major/minor number fields can be coded to represent a one-part channel number. The one-part representation is not applicable for the major/minor number fields in the time_shifted_services_descriptor() because this descriptor is not applicable to S-VCT (see Table B.A.2). The major/minor number fields in the time_shifted_services_descriptor() are only used to match against fields in the L-VCT.

The bit stream syntax for the time_shifted_service_descriptor() is shown in Table B.47.

Table B.47/J.94 – Time-Shifted Service Descriptor format

Syntax	Bits	Bytes	Format
time_shifted_service_descriptor() {			
descriptor_tag	8	1	0xA2
descriptor_length	8	1	uimsbf
reserved	3	1	'111'
number_of_services	5		uimsbf
for (i=0;i<number_of_services;i++) {			
reserved	6	1	'111111'
time_shift	10	1	uimsbf
reserved	4	2	'1111'
major_channel_number	10		uimsbf
minor_channel_number	10	2	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.

B.7.11 Component name descriptor

Table B.48 defines the component_name_descriptor(), which serves to define an optional textual name tag for any component of the service.

Table B.48/J.94 – Component Name Descriptor format

Syntax	Bits	Bytes	Format
component_name_descriptor() {			
descriptor_tag	8	1	0xA3
descriptor_length	8	1	uimsbf
component_name_string()	var		
}			

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

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

component_name_string(): The name string in the format of a Multiple String Structure (see B.8.2).

B.7.12 Daylight savings time descriptor

This descriptor is defined for optional carriage in the System Timetable section (and in no other type of table). Hosts may use the data in the descriptor if present. If not present, *no indication is being provided as to whether daylight savings time is in effect or not.* In other words, the Host shall not infer that the lack of a descriptor means that daylight savings time is not currently in effect.

A description of the use of the daylight_savings_time_descriptor() is provided in Appendix B.III. The syntax is shown in Table B.49.

Table B.49/J.94 – Daylight Savings Time Descriptor format

Syntax	Bits	Bytes	Format
daylight_savings_time_descriptor() {			
descriptor_tag	8	1	uimsbf value 0x96
descriptor_length	8	1	uimsbf
DS_status	1	1	bslbf
reserved	2		'11'
DS_day_of_month	5		uimsbf
DS_hour	8	8	uimsbf
}			

descriptor_tag: This 8-bit unsigned integer shall have the value 0x96, identifying this descriptor as daylight_savings_time_descriptor.

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

DS_status: This bit indicates 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 United States.

B.7.13 User private descriptors

Privately defined descriptors are those with descriptor_tag in the range 0xC0 through 0xFF. They may be placed at any location where descriptors may be included within the table sections described in this Service Information annex. Ownership of one or more user private descriptors is indicated by the presence of an MPEG registration_descriptor() preceding the descriptor(s).

B.8 Text string coding

This clause describes the format of text strings in this Service Information annex. Two different formats are used in this annex. Text strings in the Network Text Table uses a format called Multilingual Text String (MTS), consisting of one or more mode-length-segment blocks. The MTS format is described in B.8.1. All other tables and descriptors use a data structure called Multiple String Structure, described in B.8.2. Tables B.50 and B.51 summarize these rules.

Table B.50/J.94 – Text String Coding Format in Tables

Table ID Value (hex)	Table	Coding	Reference
0xC3	Network Text Table (NTT)	MTS	B.8.1
0xCA	Rating Region Table (RRT)	MSS	B.8.2
0xD6	Aggregate Event Information Table (AEIT)	MSS	B.8.2
0xD7	Aggregate Extended Text Table (AETT)	MSS	B.8.2

Table B.51/J.94 – Text String Coding Format in Descriptors

Descriptor tag value (hex)	Descriptor	Coding	Reference
0x87	Content advisory descriptor	MSS	B.8.2
0xA0	Extended channel name descriptor	MSS	B.8.2
0xA3	Component name descriptor	MSS	B.8.2

B.8.1 Multilingual Text String (MTS) Format

The format of Multilingual Text Strings adheres to the following structure. Items in square brackets may be repeated one or more times:

<mode><length> [<mode><length>]

A string_length field always precedes the one or more instances of mode, length, segment. This field is described in each instance where multilingual text is used, and may be either 8- or 16-bits in length, as appropriate. The value of string_length represents the sum total of all mode, length, segment blocks comprising the multilingual text string to follow, and serves to indicate the end of the text string structure.

The multilingual text data structure is designed to accommodate the need to represent a text string composed of characters from a variety of alphabets, as well as ideographic characters. Whereas characters could be represented using 16- or 32-bit character codes (as does Unicode [ISO/IEC 10646-1]), that form is inefficient and wasteful of transmission bandwidth for strings composed primarily of alphabetic rather than ideographic characters. To accommodate the need to handle Chinese, Japanese, and Korean, modes are defined that allow 16-bit (double byte) character representations in standard formats.

References below to ISO/IEC 10646-1 (Unicode) shall be to the Basic Multilingual Plane (BMP) within that standard.

mode: An 8-bit value representing the text mode to be used to interpret characters in the segment to follow. See Table B.52 for definition. Mode bytes in the range zero through 0x3E select Unicode character code pages. Mode byte value 0x3F selects 16-bit Unicode character coding. Mode bytes in the range 0x40 through 0xFF represent selection of a format effector function such as *underline ON* or *new line*. If mode is in the range 0x40 to 0x9F, then the length/segment portion is omitted. Format effector codes in the range 0x40 through 0x9F involve no associated parametric data; hence the omission of the length/segment portion. Format effector codes in the range 0xA0 through 0xFF include one or more parameters specific to the particular format effector function.

Table B.52/J.94 – Mode Byte Encoding

Mode Byte	Meaning	Language(s) or script
0x00	Select ISO/IEC 10646-1 Page 0x00	ASCII, ISO Latin-1 (Roman)
0x01	Select ISO/IEC 10646-1 Page 0x01	European Latin (many)[a]
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[b]
0x07-0x08	Reserved	–
0x09	Select ISO/IEC 10646-1 Page 0x09	Devanagari[c], Bengali

Table B.52/J.94 – Mode Byte Encoding *(concluded)*

Mode Byte	Meaning	Language(s) or script
0x0A	Select ISO/IEC 10646-1 Page 0x0A	Punjabi, Gujarti
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	Select ISO/IEC 10646-1 Page 0x0F	Tibetan
0x10	Select ISO/IEC 10646-1 Page 0x10	Georgian
0x11-0x1F	Reserved	–
0x20	Select ISO/IEC 10646-1 Page 0x20	Miscellaneous[d]
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
0x40-0x9F	Format effector (single byte)	See Table B.41.
0xA0-0xFF	Format effector (with parameter[s])	–

a) When combined with page zero (ASCII and ISO Latin-1), covers Afrikaans, Breton, Basque, Catalan, Croatian, Czech, Danish, Dutch, Esperanto, Estonian, Faroese, Finnish, Flemish, Firsian, Greenlandic, Hungarian, Icelandic, Italian, Latin, Latvian, Lithuanian, Malay, Maltese, Norwegian, Polish, Portuguese, Provencal, Ghaeto-Romanic, Romanian, Romany, Slovak, Slovenian, Serbian, Spanish, Swedish, Turkish, and Welsh.

b) Also Persian, Urdu, Pashto, Sindhi, and Kurdish.

c) 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.

d) General punctuation, superscripts and subscripts, currency symbols, and other diacritics.

Table B.53 describes the format of the multilingual_text_string().

Table B.53/J.94 – Multilingual text string format

	Bits	Bytes	Format
multilingual_text_string(){			
For (i=0; i<N; i++) {			
Mode	8	(1)	uimsbf
if (mode < 0x3F) {			
eightbit_string_length	8	((1))	uimsbf
for (i=0; i<eightbit_string_length; I++) {			
eightbit_char	8	(((1)))	uimsbf
}			
} else if (mode==0x3F) {			
sixteenbit_string_length	8	((1))	uimsbf (even)
for (i=0; i<(sixteenbit_string_length); i+=2) {			
sixteenbit_char	16	(((2)))	uimsbf
}			
} else if (mode >= 0xA0) {			
format_effector_param_length	8	((1))	uimsbf
for (i=0; i<(format_effector_param_length); i++) {			
format_effector_data	8	(((1)))	
}			
}			
}			
}			

length: An 8-bit unsigned integer number representing the number of bytes in the segment to follow in this block.

segment: An array of bytes representing a character string formatted according to the mode byte.

B.8.1.1 Mode byte definition

The mode byte is used either to select an ISO/IEC 10646-1 code page from the BMP (exact mapping, or in the case of page zero, an extended mapping as defined herein), or to indicate that the text segment is coded in one of a number of standard double-byte formats. Table B.52 shows the encoding of the mode byte. Values in the zero to 0x33 range select ISO/IEC 10646-1 code pages.

Value 0x3F selects double-byte forms used with non-alphabetic script systems, where the segment consists of a sequence of 16-bit character codes according to the ISO/IEC 10646-1 standard. Byte ordering is high-order byte first (Motorola 680xx style), also known as *big-endian*.

B.8.1.2 Format effectors

Mode bytes in the 0x40 to 0xFF range are defined as format effectors. Table B.54 defines the encoding for currently defined single-byte values. Format effectors in the range 0x40 through 0x9F are self-contained, and do not have a length or data field following them. Format effectors in the range 0xA0 through 0xFF include a multi-byte parameter field. No multi-byte format effectors are currently defined.

Table B.54/J.94 – Format Effector Function Codes

Mode byte	Meaning
0x40-0x7F	Reserved
0x80	new line, left justify
0x81	new line, right justify
0x82	new line, center
0x83	italics ON
0x84	italics OFF
0x85	underline ON
0x86	underline OFF
0x87	bold ON
0x88	bold OFF
0x89-0x9F	Reserved

Line justification

Values 0x80, 0x81, and 0x82 signify the end of a line of displayed text. Value 0x80 indicates that the text is displayed left justified within an enclosing rectangular region (defined outside the scope of the text string). Value 0x81 indicates that the text is displayed right justified. Value 0x82 indicates that the text is centered on the line. The dimensions and location on the screen of the box into which text is placed is defined outside the scope of the text string itself.

Italics, underline, bold attributes

These format effectors toggle *italics*, underline, and **bold** display attributes. The italics, underline, and bold format effectors indicate the start or end of the associated formatting within a text string. Formatting extends through new lines. For example, to display three lines of bold text, only one instance of the *bold ON* format effector is required.

Processing of unknown or unsupported format effectors

Hosts must discard format effectors that are unknown, or known not to be supported within a specific Host model. If a parameter value carries an undefined value, that format effector is expected to be discarded.

B.8.1.3 Default attributes

Upon entry to a multilingual text string, all mode toggles (bold, underline, italics) shall be assumed "OFF".

B.8.1.4 Mode Zero

ISO/IEC 10646-1 page zero (U+0000 through U+00FF) includes ASCII in the lower half (U+0000 through U+007F), and Latin characters from ISO 8859-1, *Latin-1*, in U+0090 through U+00FF. This set of characters covers Danish, Dutch, Faroese, Finnish, French, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish and Swedish. Many other languages can be written with this set of letters, including Hawaiian, Indonesian/Malay, and Swahili.

Table B.55 shows encodings of page zero characters in the range 0x80 through 0x9F (these are undefined within ISO/IEC 10646-1).

Table B.55/J.94 – Encodings of columns 8 and 9 of mode zero latin character set

	8	9
0	<RESERVED>	<RESERVED>
1	<RESERVED>	<RESERVED>
2	<RESERVED>	<RESERVED>
3	<RESERVED>	<RESERVED>
4	<RESERVED>	<RESERVED>
5	<RESERVED>	<RESERVED>
6	<RESERVED>	<RESERVED>
7	<RESERVED>	<RESERVED>
8	<RESERVED>	U+2030 – <PER MILLE>
9	<RESERVED>	<RESERVED>
A	<RESERVED>	U+266A – <MUSICAL NOTE>
B	<RESERVED>	<RESERVED>
C	<RESERVED>	U+2190 – <LEFT ARROW>
D	<RESERVED>	U+2191 – <UP ARROW>
E	<RESERVED>	U+2192 – <RIGHT ARROW>
F	<RESERVED>	U+2193 – <DOWN ARROW>

B.8.1.5 Supported characters

Support for specific characters and languages depends upon the specific model of Standard-compatible Host. Not all Hosts support all defined character sets or character codes. Use of multilingual text must be predicated on the knowledge of limitations in character rendering inherent in different Host models for which text is available.

B.8.2 Multiple String Structure (MSS)

The Multiple String Structure 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 B.56.

Table B.56/J.94 – 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
}		
}		
}		

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

ISO_639_language_code: This 3-byte (24 bits) field, in conformance with ISO 639-2/B, specifies the language used for the ith string.

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

compression_type: This 8-bit field identifies the compression type for the jth segment. Allowed values for this field are shown in Table B.57.

Table B.57/J.94 – 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 of SCTE DVS 097, ATSC Standard A/65 (1997).
0x02	Huffman coding using standard encode/decode tables defined in Table C.6 and C.7 in Annex C of SCTE DVS 097, ATSC Standard A/65 (1997).
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 B.58 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. Hosts shall ignore string bytes associated with unknown or unsupported mode values.

Table B.58/J.94 – Modes

Mode	Meaning	Language(s) or script
0x00	Select ISO/IEC 10646-1 Page 0x00	ASCII, ISO Latin-1 (Roman)[a)]
0x01	Select ISO/IEC 10646-1 Page 0x01	European Latin (many)[b)]
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[c)]
0x07-0x08	Reserved	–
0x09	Select ISO/IEC 10646-1 Page 0x09	Devanagari[d)], 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	Select ISO/IEC 10646-1 Page 0x0F	Tibetan

Table B.58/J.94 – Modes *(concluded)*

Mode	Meaning	Language(s) or script
0x10	Select ISO/IEC 10646-1 Page 0x10	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	

a) 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.

b) 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.

c) Also Persian, Urdu, Pashto, Sindhi, and Kurdish.

d) 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.

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

compressed_string_byte[k]: The kth byte of the jth segment.

ANNEX B.A

Operational profiles for cable service information delivery

B.A.1 Operational profiles

This Annex B.A specifies Service Information tables that are required for delivery via an out-of-band channel on cable. Six profiles are described with required and optional data specified for out-of-band transport via cable. Adherence to these profile specifications is necessary for compliance with SCTE standard transport streams.

B.A.1.1 Profile 1 – Baseline

This Baseline Profile reflects a practice in cable where the Short-Form Virtual Channel Table, the Modulation Mode Subtable and the Carrier Definition Subtable are used for channel navigation.

B.A.1.2 Profile 2 – Revision detection

Profile 2 uses the same channel navigation mechanism as Profile 1 while adding a detection mechanism that facilitates revision handling of tables. The revision detection mechanism is applicable to the Network Information Table, Network Text Table, and S-VCT that are also used in Profile 1.

B.A.1.3 Profile 3 – Parental advisory

Profile 3 uses Profile 2 as the base and adds support for the Rating Region Table in order to be compliant with the FCC-mandated V-chip content advisory scheme. Since for the U.S. and its possessions, EIA-766 defines the contents of version 0 RRT, use of RRT is more applicable to outside of North America. The channel navigation mechanism is the same as in Profile 1.

B.A.1.4 Profile 4 – Standard electronic program guide data

Profile 4 uses Profile 3 as the base and further defines a standard format for delivery of Electronic Program Guide data by using the Aggregate Event Information Table and the Aggregate Extended Text Table. The Master Guide Table shall be supported to manage the AEITs, AETTs and other applicable tables from Profile 3. The same mechanism as in Profile 1 is used for channel navigation.

B.A.1.5 Profile 5 – Combination

Support for channel navigation based on L-VCT and MGT is added. Backward compatibility with systems operating within profiles 1 to 4 is maintained. Using Profile 5, a cable operator could have a mixture of devices requiring the S-VCT, NIT and NTT tables as well as ones requiring the long-form tables: i.e. L-VCT, MGT.

When using Profile 5, both the S-VCT and the L-VCT shall be present, and each shall describe all available services.

B.A.1.6 Profile 6 – PSIP Only

Profile 6 is based solely on long-form tables and is an extension of the terrestrial broadcasting mechanism. Channel navigation is based on the Long-form Virtual Channel Table. The AEIT and the optional AETT streams are used to provide EPG data.

B.A.2 Profile Definition Tables

In order to conform to this Service Information Annex B.A, a cable operator shall send a collection of tables that corresponds to one or more of the defined operational profiles defined in Table B.A.1 and Table B.A.2.

Table B.A.1/J.94 – Usage of Table Sections in Various Profiles

		Profile 1	**Profile 2**	**Profile 3**	**Profile 4**	**Profile 5**	**Profile 6**
Table Section	Table ID	Baseline	Revision Detection	Parental Advisory	Standard EPG Data	Combi-nation	PSIP only (Note 1)
Network Information Table	0xC2						
Carrier Definition Subtable		M	M	M	M	M	–
Modulation Mode Subtable		M	M	M	M	M	–
Network Text Table	0xC3						
Source Name Subtable		O	O	O	M	M	–
Short-form Virtual Channel Table	0xC4						
Virtual Channel Map		M	M	M	M	M	–
Defined Channels Map		M	M	M	M	M	–
Inverse Channel Map		O	O	O	O	O	–
System Timetable	0xC5	M	M	M	M	M	M
Master Guide Table	0xC7	–	–	(Note 2)	M	M	M
Rating Region Table	0xCA	–	–	(Note 3)	(Note 3)	(Note 3)	(Note 3)
Long-form Virtual Channel Table	0xC9	–	–	–	–	M	M
Aggregate Event Information Table	0xD6	–	–	–	M	M	M
Aggregate Extended Text Table	0xD7	–	–	–	O	O	O

M Mandatory (shall be present)

O Optional (may or may not be present)

– Not applicable (shall not be present)

NOTE 1 – Exception: System Timetable (table ID 0xC5 is used here instead of table ID 0xCD defined in PSIP) and other modifications.

NOTE 2 – Mandatory for outside of North America to describe any transmitted RRT. For region 0x01 (US and possessions), delivery of an RRT is optional, because this table is standardized in EIA-766.

NOTE 3 – Exception: delivery of the RRT corresponding to region 0x01 (US and possessions) is optional, because this table is standardized in EIA-766.

Table B.A.2/J.94 – Usage of Descriptors in Various Profiles

		Profile 1	**Profile 2**	**Profile 3**	**Profile 4**	**Profile 5**	**Profile 6**
Descriptor (and associated table)	Tag	Baseline	Revision Detection	Parental Advisory	Standard EPG Data	Combi-nation	PSIP only (Note 1)
AC-3 audio (PMT, AEIT)	0x81	–	–	–	O	O	O
Caption service (PMT, AEIT)	0x86	–	–	–	O	O	O
Content advisory (PMT, AEIT)	0x87	–	–	(Note 2)	(Note 2)	(Note 2)	(Note 2)
Revision detection (NIT,NTT, S-VCT)	0x93	–	M	M	M	M	–
Two-part channel number (S-VCT)	0x94	–	–	–	O	O	–
Channel properties (S-CT)	0x95	–	–	–	O	O	–
Daylight savings time (STT)	0x96	–	–	O	M	M	M
Extended channel name (L-VCT)	0xA0	–	–	–	–	O	O
Time-shifted service (L-CT)	0xA2	–	–	–	–	O	O
Component name (PMT)	0xA3	–	–	–	O	O	O

M Mandatory (shall be present)

O Optional (may or may not be present)

– Not applicable (shall not be present)

NOTE 1 – Exception: System Timetable (table ID 0xC5 is used here instead of table ID 0xCD defined in PSIP) and other modifications.

NOTE 2 – The content_advisory_descriptor() shall be present in the AEIT and PMT for a given program when Content Advisory data is available for that program. It is not required for programs for which Content Advisory data is not available.

B.A.3 Operational considerations for the use of profiles (Informative)

1) If devices deployed in a particular cable system require the S-VCT in Profiles 1-5 for navigation, cable operator's use of P6 will cause operational problems.

2) If devices in use require L-VCT for navigation, cable operator's use of Profiles 1-4 will cause operational problems.

3) To provide EPG data, cable-ready devices operating on a cable system conforming to Profiles 1, 2 or 3 must use alternative protocols and methods which are beyond the scope of this Annex B.A.

ANNEX B.B

Packet rates

B.B.1 Maximum cycle times

Table B.B.1 lists the maximum cycle time for Service Information table sections for out-of-band cable operation, when the indicated table is present.

Table B.B.1/J.94 – Maximum cycle time for the STT, MGT, S-VCT, L-VCT and RRT

Table Section	STT	MGT	S-VCT	L-VCT	RRT
Cycle time	1 min	500 msec	2 min	2 min	1 min

B.B.2 Maximum transmission rates

Table B.B.2 lists the maximum transmission rate for SI packet streams.

Table B.B.2/J.94 – Maximum rate for each packet stream

PID	SI_base PID	Any AEIT/AETT PID
Rate (bit/s)	150 000	150 000

B.B.3 Minimum transmission rates

Table B.B.3 lists the minimum transmission rate for SI packet streams. Minimum per-PID bit rates are required to ensure efficiency of recovery of EPG data covering the current time period (3 hours minimum) across the POD to Host interface, given the small number of PID values that can be used concurrently.

Table B.B.3/J.94 – Minimum rate for each packet stream

PID	AEIT-0,1/AETT-0,1 PID
Rate (bit/s)	10 000

ANNEX B.C

Standard Huffman tables for text compression

This Annex B.C 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. Clause B.C.2 defines standard Huffman encode and decode tables optimized for English-language text such as that typically found in program titles. Clause B.C.3 defines Huffman encode and decode tables optimized for English-language text such as that typically found in program descriptions. Hosts 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 B.C.5 and B.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.

B.C.1 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 in Table B.C.1 have special definitions:

Table B.C.1/J.94 – 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.

B.C.1.1 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-1*). 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.

B.C.1.2 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 B.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 B.C.2.

Decode tables have the following format:

Table B.C.2/J.94 – 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.

B.C.1.2.1 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 ith character's order-1 tree.

B.C.1.2.2 Order-1 decode trees

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

Decode trees have the format as shown in Table B.C.3:

Table B.C.3/J.94 – 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.

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.

B.C.2 Standard compression Type 1 Encode/Decode Tables

The following encode/decode tables (Tables B.C.4 and B.C.5) 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 B.C.4/J.94 – English-language Program Title Encode Table

Prior Symbol: 0 Symbol: 27 Code: 11001011
Prior Symbol: 0 Symbol: '$' Code: 1100101011
Prior Symbol: 0 Symbol: '2' Code: 011010010
Prior Symbol: 0 Symbol: '4' Code: 1100101010
Prior Symbol: 0 Symbol: '7' Code: 011010011
Prior Symbol: 0 Symbol: 'A' Code: 0111
Prior Symbol: 0 Symbol: 'B' Code: 1001
Prior Symbol: 0 Symbol: 'C' Code: 1011
Prior Symbol: 0 Symbol: 'D' Code: 11011
Prior Symbol: 0 Symbol: 'E' Code: 10001
Prior Symbol: 0 Symbol: 'F' Code: 11000
Prior Symbol: 0 Symbol: 'G' Code: 11100
Prior Symbol: 0 Symbol: 'H' Code: 11111
Prior Symbol: 0 Symbol: 'I' Code: 10000
Prior Symbol: 0 Symbol: 'J' Code: 01100
Prior Symbol: 0 Symbol: 'K' Code: 1100110
Prior Symbol: 0 Symbol: 'L' Code: 11101
Prior Symbol: 0 Symbol: 'M' Code: 1010
Prior Symbol: 0 Symbol: 'N' Code: 0011
Prior Symbol: 0 Symbol: 'O' Code: 011011
Prior Symbol: 0 Symbol: 'P' Code: 11110
Prior Symbol: 0 Symbol: 'Q' Code: 01101000
Prior Symbol: 0 Symbol: 'R' Code: 11010
Prior Symbol: 0 Symbol: 'S' Code: 000
Prior Symbol: 0 Symbol: 'T' Code: 010
Prior Symbol: 0 Symbol: 'U' Code: 0110101
Prior Symbol: 0 Symbol: 'V' Code: 1100111
Prior Symbol: 0 Symbol: 'W' Code: 0010
Prior Symbol: 0 Symbol: 'Y' Code: 1100100
Prior Symbol: 0 Symbol: 'Z' Code: 110010100
Prior Symbol: 1 Symbol: 27 Code: 1
Prior Symbol: 2 Symbol: 27 Code: 1
Prior Symbol: 3 Symbol: 27 Code: 1
Prior Symbol: 4 Symbol: 27 Code: 1
Prior Symbol: 5 Symbol: 27 Code: 1
Prior Symbol: 6 Symbol: 27 Code: 1
Prior Symbol: 7 Symbol: 27 Code: 1
Prior Symbol: 8 Symbol: 27 Code: 1
Prior Symbol: 9 Symbol: 27 Code: 1
Prior Symbol: 10 Symbol: 27 Code: 1
Prior Symbol: 11 Symbol: 27 Code: 1
Prior Symbol: 12 Symbol: 27 Code: 1
Prior Symbol: 13 Symbol: 27 Code: 1
Prior Symbol: 14 Symbol: 27 Code: 1
Prior Symbol: 15 Symbol: 27 Code: 1
Prior Symbol: 16 Symbol: 27 Code: 1
Prior Symbol: 17 Symbol: 27 Code: 1
Prior Symbol: 18 Symbol: 27 Code: 1
Prior Symbol: 19 Symbol: 27 Code: 1
Prior Symbol: 20 Symbol: 27 Code: 1
Prior Symbol: 21 Symbol: 27 Code: 1
Prior Symbol: 22 Symbol: 27 Code: 1
Prior Symbol: 23 Symbol: 27 Code: 1
Prior Symbol: 24 Symbol: 27 Code: 1
Prior Symbol: 25 Symbol: 27 Code: 1
Prior Symbol: 26 Symbol: 27 Code: 1
Prior Symbol: 27 Symbol: 27 Code: 1
Prior Symbol: 28 Symbol: 27 Code: 1
Prior Symbol: 29 Symbol: 27 Code: 1
Prior Symbol: 30 Symbol: 27 Code: 1
Prior Symbol: 31 Symbol: 27 Code: 1
Prior Symbol: ' ' Symbol: 27 Code: 10010100
Prior Symbol: ' ' Symbol: '&' Code: 010001
Prior Symbol: ' ' Symbol: '"' Code: 010000100
Prior Symbol: ' ' Symbol: '-' Code: 00000001
Prior Symbol: ' ' Symbol: '1' Code: 010000101
Prior Symbol: ' ' Symbol: '2' Code: 00000010
Prior Symbol: ' ' Symbol: '3' Code: 01000001
Prior Symbol: ' ' Symbol: '9' Code: 000000000
Prior Symbol: ' ' Symbol: 'A' Code: 10111

Prior Symbol: ' ' Symbol: 'B' Code: 0010
Prior Symbol: ' ' Symbol: 'C' Code: 1100
Prior Symbol: ' ' Symbol: 'D' Code: 11100
Prior Symbol: ' ' Symbol: 'E' Code: 011010
Prior Symbol: ' ' Symbol: 'F' Code: 10011
Prior Symbol: ' ' Symbol: 'G' Code: 00001
Prior Symbol: ' ' Symbol: 'H' Code: 10101
Prior Symbol: ' ' Symbol: 'I' Code: 111111
Prior Symbol: ' ' Symbol: 'J' Code: 111110
Prior Symbol: ' ' Symbol: 'K' Code: 010011
Prior Symbol: ' ' Symbol: 'L' Code: 11110
Prior Symbol: ' ' Symbol: 'M' Code: 0101
Prior Symbol: ' ' Symbol: 'N' Code: 10110
Prior Symbol: ' ' Symbol: 'O' Code: 011011
Prior Symbol: ' ' Symbol: 'P' Code: 11101
Prior Symbol: ' ' Symbol: 'Q' Code: 100100011
Prior Symbol: ' ' Symbol: 'R' Code: 10100
Prior Symbol: ' ' Symbol: 'S' Code: 1101
Prior Symbol: ' ' Symbol: 'T' Code: 1000
Prior Symbol: ' ' Symbol: 'U' Code: 1001001
Prior Symbol: ' ' Symbol: 'V' Code: 1001011
Prior Symbol: ' ' Symbol: 'W' Code: 0011
Prior Symbol: ' ' Symbol: 'X' Code: 0000000010
Prior Symbol: ' ' Symbol: 'Y' Code: 000001
Prior Symbol: ' ' Symbol: 'Z' Code: 00000011
Prior Symbol: ' ' Symbol: 'a' Code: 01100
Prior Symbol: ' ' Symbol: 'b' Code: 10010101
Prior Symbol: ' ' Symbol: 'c' Code: 01000000
Prior Symbol: ' ' Symbol: 'd' Code: 01000011
Prior Symbol: ' ' Symbol: 'e' Code: 0000000011
Prior Symbol: ' ' Symbol: 'f' Code: 10010000
Prior Symbol: ' ' Symbol: 'i' Code: 010010
Prior Symbol: ' ' Symbol: 'l' Code: 100100010
Prior Symbol: ' ' Symbol: 'o' Code: 0001
Prior Symbol: ' ' Symbol: 't' Code: 0111
Prior Symbol: '!' Symbol: 0 Code: 1
Prior Symbol: '!' Symbol: 27 Code: 01
Prior Symbol: '!' Symbol: ' ' Code: 00
Prior Symbol: '"' Symbol: 27 Code: 1
Prior Symbol: '#' Symbol: 27 Code: 1
Prior Symbol: '$' Symbol: 27 Code: 1
Prior Symbol: '$' Symbol: '1' Code: 0
Prior Symbol: '%' Symbol: 27 Code: 1
Prior Symbol: '&' Symbol: 27 Code: 0
Prior Symbol: '&' Symbol: ' ' Code: 1
Prior Symbol: ''' Symbol: 27 Code: 011
Prior Symbol: ''' Symbol: ' ' Code: 010
Prior Symbol: ''' Symbol: '9' Code: 0001
Prior Symbol: ''' Symbol: 'd' Code: 0000
Prior Symbol: ''' Symbol: 's' Code: 1
Prior Symbol: ''' Symbol: 't' Code: 001
Prior Symbol: '(' Symbol: 27 Code: 1
Prior Symbol: ')' Symbol: 27 Code: 1
Prior Symbol: '*' Symbol: 27 Code: 00
Prior Symbol: '*' Symbol: 'A' Code: 01
Prior Symbol: '*' Symbol: 'H' Code: 10
Prior Symbol: '*' Symbol: 'S' Code: 11
Prior Symbol: '+' Symbol: 27 Code: 1
Prior Symbol: ',' Symbol: 27 Code: 0
Prior Symbol: ',' Symbol: ' ' Code: 1
Prior Symbol: '-' Symbol: 27 Code: 01
Prior Symbol: '-' Symbol: ' ' Code: 111
Prior Symbol: '-' Symbol: '-' Code: 1101
Prior Symbol: '-' Symbol: '1' Code: 1000
Prior Symbol: '-' Symbol: 'A' Code: 001
Prior Symbol: '-' Symbol: 'M' Code: 000
Prior Symbol: '-' Symbol: 'R' Code: 1001
Prior Symbol: '-' Symbol: 'S' Code: 1010
Prior Symbol: '-' Symbol: 'T' Code: 1011
Prior Symbol: '-' Symbol: 'U' Code: 1100
Prior Symbol: '.' Symbol: 0 Code: 111
Prior Symbol: '.' Symbol: 27 Code: 101
Prior Symbol: '.' Symbol: ' ' Code: 0
Prior Symbol: '.' Symbol: '.' Code: 110
Prior Symbol: '.' Symbol: 'I' Code: 10010
Prior Symbol: '.' Symbol: 'S' Code: 1000
Prior Symbol: '.' Symbol: 'W' Code: 10011
Prior Symbol: '/' Symbol: 27 Code: 1
Prior Symbol: '0' Symbol: 0 Code: 01
Prior Symbol: '0' Symbol: 27 Code: 001
Prior Symbol: '0' Symbol: ' ' Code: 10
Prior Symbol: '0' Symbol: '-' Code: 000
Prior Symbol: '0' Symbol: '0' Code: 11
Prior Symbol: '1' Symbol: 0 Code: 010
Prior Symbol: '1' Symbol: 27 Code: 011
Prior Symbol: '1' Symbol: ' ' Code: 110
Prior Symbol: '1' Symbol: '0' Code: 111
Prior Symbol: '1' Symbol: '1' Code: 100

Prior Symbol: '1' Symbol: '2' Code: 101
Prior Symbol: '1' Symbol: '9' Code: 00
Prior Symbol: '2' Symbol: 0 Code: 11
Prior Symbol: '2' Symbol: 27 Code: 10
Prior Symbol: '2' Symbol: '0' Code: 01
Prior Symbol: '2' Symbol: '1' Code: 000
Prior Symbol: '2' Symbol: ':' Code: 001
Prior Symbol: '3' Symbol: 0 Code: 0
Prior Symbol: '3' Symbol: 27 Code: 11
Prior Symbol: '3' Symbol: '0' Code: 10
Prior Symbol: '4' Symbol: 27 Code: 0
Prior Symbol: '4' Symbol: '8' Code: 1
Prior Symbol: '5' Symbol: 27 Code: 1
Prior Symbol: '6' Symbol: 27 Code: 1
Prior Symbol: '7' Symbol: 27 Code: 0
Prior Symbol: '7' Symbol: '0' Code: 1
Prior Symbol: '8' Symbol: 27 Code: 0
Prior Symbol: '8' Symbol: ' ' Code: 1
Prior Symbol: '9' Symbol: 27 Code: 11
Prior Symbol: '9' Symbol: '0' Code: 01
Prior Symbol: '9' Symbol: '1' Code: 100
Prior Symbol: '9' Symbol: '3' Code: 101
Prior Symbol: '9' Symbol: '9' Code: 00
Prior Symbol: ':' Symbol: 27 Code: 0
Prior Symbol: ':' Symbol: ' ' Code: 1
Prior Symbol: ';' Symbol: 27 Code: 1
Prior Symbol: '<' Symbol: 27 Code: 1
Prior Symbol: '=' Symbol: 27 Code: 1
Prior Symbol: '>' Symbol: 27 Code: 1
Prior Symbol: '?' Symbol: 0 Code: 1
Prior Symbol: '?' Symbol: 27 Code: 0
Prior Symbol: '@' Symbol: 27 Code: 1
Prior Symbol: 'A' Symbol: 27 Code: 00010
Prior Symbol: 'A' Symbol: ' ' Code: 010
Prior Symbol: 'A' Symbol: '*' Code: 1101000
Prior Symbol: 'A' Symbol: '-' Code: 1101001
Prior Symbol: 'A' Symbol: '.' Code: 1101010
Prior Symbol: 'A' Symbol: 'B' Code: 110110
Prior Symbol: 'A' Symbol: 'b' Code: 110010
Prior Symbol: 'A' Symbol: 'c' Code: 01100
Prior Symbol: 'A' Symbol: 'd' Code: 001
Prior Symbol: 'A' Symbol: 'f' Code: 01101
Prior Symbol: 'A' Symbol: 'g' Code: 011110
Prior Symbol: 'A' Symbol: 'i' Code: 110011
Prior Symbol: 'A' Symbol: 'l' Code: 100
Prior Symbol: 'A' Symbol: 'm' Code: 111
Prior Symbol: 'A' Symbol: 'n' Code: 101
Prior Symbol: 'A' Symbol: 'p' Code: 110111
Prior Symbol: 'A' Symbol: 'r' Code: 0000
Prior Symbol: 'A' Symbol: 's' Code: 00011
Prior Symbol: 'A' Symbol: 't' Code: 011111
Prior Symbol: 'A' Symbol: 'u' Code: 11000
Prior Symbol: 'A' Symbol: 'v' Code: 1101011
Prior Symbol: 'A' Symbol: 'w' Code: 01110
Prior Symbol: 'B' Symbol: 27 Code: 00010
Prior Symbol: 'B' Symbol: 'A' Code: 000110
Prior Symbol: 'B' Symbol: 'C' Code: 0000
Prior Symbol: 'B' Symbol: 'S' Code: 000111
Prior Symbol: 'B' Symbol: 'a' Code: 111
Prior Symbol: 'B' Symbol: 'e' Code: 01
Prior Symbol: 'B' Symbol: 'i' Code: 1010
Prior Symbol: 'B' Symbol: 'l' Code: 1011
Prior Symbol: 'B' Symbol: 'o' Code: 110
Prior Symbol: 'B' Symbol: 'r' Code: 001
Prior Symbol: 'B' Symbol: 'u' Code: 100
Prior Symbol: 'C' Symbol: 27 Code: 00101
Prior Symbol: 'C' Symbol: ' ' Code: 10110
Prior Symbol: 'C' Symbol: 'A' Code: 0011100
Prior Symbol: 'C' Symbol: 'B' Code: 001111
Prior Symbol: 'C' Symbol: 'O' Code: 101110
Prior Symbol: 'C' Symbol: 'a' Code: 100
Prior Symbol: 'C' Symbol: 'e' Code: 101111
Prior Symbol: 'C' Symbol: 'h' Code: 01
Prior Symbol: 'C' Symbol: 'i' Code: 00110
Prior Symbol: 'C' Symbol: 'l' Code: 000
Prior Symbol: 'C' Symbol: 'o' Code: 11
Prior Symbol: 'C' Symbol: 'r' Code: 1010
Prior Symbol: 'C' Symbol: 'u' Code: 00100
Prior Symbol: 'C' Symbol: 'y' Code: 0011101
Prior Symbol: 'D' Symbol: 27 Code: 01001
Prior Symbol: 'D' Symbol: 'a' Code: 10
Prior Symbol: 'D' Symbol: 'e' Code: 111
Prior Symbol: 'D' Symbol: 'i' Code: 110
Prior Symbol: 'D' Symbol: 'o' Code: 00
Prior Symbol: 'D' Symbol: 'r' Code: 011
Prior Symbol: 'D' Symbol: 'u' Code: 0101
Prior Symbol: 'D' Symbol: 'y' Code: 01000
Prior Symbol: 'E' Symbol: 27 Code: 011
Prior Symbol: 'E' Symbol: 'C' Code: 1010

Prior Symbol: 'E' Symbol: 'a' Code: 111
Prior Symbol: 'E' Symbol: 'd' Code: 000
Prior Symbol: 'E' Symbol: 'l' Code: 1100
Prior Symbol: 'E' Symbol: 'm' Code: 0100
Prior Symbol: 'E' Symbol: 'n' Code: 1101
Prior Symbol: 'E' Symbol: 'q' Code: 101110
Prior Symbol: 'E' Symbol: 's' Code: 10110
Prior Symbol: 'E' Symbol: 'u' Code: 101111
Prior Symbol: 'E' Symbol: 'v' Code: 100
Prior Symbol: 'E' Symbol: 'x' Code: 001
Prior Symbol: 'E' Symbol: 'y' Code: 0101
Prior Symbol: 'F' Symbol: 27 Code: 011111
Prior Symbol: 'F' Symbol: ' ' Code: 011110
Prior Symbol: 'F' Symbol: 'L' Code: 01110
Prior Symbol: 'F' Symbol: 'a' Code: 10
Prior Symbol: 'F' Symbol: 'e' Code: 0110
Prior Symbol: 'F' Symbol: 'i' Code: 110
Prior Symbol: 'F' Symbol: 'l' Code: 000
Prior Symbol: 'F' Symbol: 'o' Code: 010
Prior Symbol: 'F' Symbol: 'r' Code: 111
Prior Symbol: 'F' Symbol: 'u' Code: 001
Prior Symbol: 'G' Symbol: 27 Code: 10110
Prior Symbol: 'G' Symbol: '.' Code: 101010
Prior Symbol: 'G' Symbol: 'A' Code: 101111
Prior Symbol: 'G' Symbol: 'a' Code: 1110
Prior Symbol: 'G' Symbol: 'e' Code: 110
Prior Symbol: 'G' Symbol: 'h' Code: 10100
Prior Symbol: 'G' Symbol: 'i' Code: 100
Prior Symbol: 'G' Symbol: 'l' Code: 101011
Prior Symbol: 'G' Symbol: 'o' Code: 01
Prior Symbol: 'G' Symbol: 'r' Code: 00
Prior Symbol: 'G' Symbol: 'u' Code: 1111
Prior Symbol: 'G' Symbol: 'y' Code: 101110
Prior Symbol: 'H' Symbol: 0 Code: 111010
Prior Symbol: 'H' Symbol: 27 Code: 111011
Prior Symbol: 'H' Symbol: 'a' Code: 110
Prior Symbol: 'H' Symbol: 'e' Code: 10
Prior Symbol: 'H' Symbol: 'i' Code: 1111
Prior Symbol: 'H' Symbol: 'o' Code: 0
Prior Symbol: 'H' Symbol: 'u' Code: 11100
Prior Symbol: 'I' Symbol: 0 Code: 1000
Prior Symbol: 'I' Symbol: 27 Code: 1001
Prior Symbol: 'I' Symbol: ' ' Code: 11110
Prior Symbol: 'I' Symbol: '.' Code: 111110
Prior Symbol: 'I' Symbol: ':' Code: 101110
Prior Symbol: 'I' Symbol: 'I' Code: 1100
Prior Symbol: 'I' Symbol: 'T' Code: 101111
Prior Symbol: 'I' Symbol: 'c' Code: 10110
Prior Symbol: 'I' Symbol: 'm' Code: 1010
Prior Symbol: 'I' Symbol: 'n' Code: 0
Prior Symbol: 'I' Symbol: 'r' Code: 111111
Prior Symbol: 'I' Symbol: 's' Code: 1101
Prior Symbol: 'I' Symbol: 't' Code: 1110
Prior Symbol: 'J' Symbol: 27 Code: 000
Prior Symbol: 'J' Symbol: 'a' Code: 01
Prior Symbol: 'J' Symbol: 'e' Code: 11
Prior Symbol: 'J' Symbol: 'o' Code: 10
Prior Symbol: 'J' Symbol: 'u' Code: 001
Prior Symbol: 'K' Symbol: 27 Code: 000
Prior Symbol: 'K' Symbol: 'a' Code: 0100
Prior Symbol: 'K' Symbol: 'e' Code: 001
Prior Symbol: 'K' Symbol: 'i' Code: 1
Prior Symbol: 'K' Symbol: 'n' Code: 0111
Prior Symbol: 'K' Symbol: 'o' Code: 0101
Prior Symbol: 'K' Symbol: 'u' Code: 0110
Prior Symbol: 'L' Symbol: 27 Code: 01001
Prior Symbol: 'L' Symbol: ' ' Code: 01000
Prior Symbol: 'L' Symbol: 'a' Code: 10
Prior Symbol: 'L' Symbol: 'e' Code: 011
Prior Symbol: 'L' Symbol: 'i' Code: 11
Prior Symbol: 'L' Symbol: 'o' Code: 00
Prior Symbol: 'L' Symbol: 'u' Code: 0101
Prior Symbol: 'M' Symbol: 27 Code: 1011111
Prior Symbol: 'M' Symbol: '*' Code: 10111100
Prior Symbol: 'M' Symbol: 'T' Code: 10111101
Prior Symbol: 'M' Symbol: 'a' Code: 11
Prior Symbol: 'M' Symbol: 'c' Code: 101110
Prior Symbol: 'M' Symbol: 'e' Code: 1010
Prior Symbol: 'M' Symbol: 'i' Code: 100
Prior Symbol: 'M' Symbol: 'o' Code: 00
Prior Symbol: 'M' Symbol: 'r' Code: 10110
Prior Symbol: 'M' Symbol: 'u' Code: 010
Prior Symbol: 'M' Symbol: 'y' Code: 011
Prior Symbol: 'N' Symbol: 27 Code: 1000
Prior Symbol: 'N' Symbol: ' ' Code: 110001
Prior Symbol: 'N' Symbol: 'B' Code: 1001
Prior Symbol: 'N' Symbol: 'F' Code: 110010
Prior Symbol: 'N' Symbol: 'N' Code: 110000
Prior Symbol: 'N' Symbol: 'a' Code: 1101

Prior Symbol: 'N' Symbol: 'e' Code: 0
Prior Symbol: 'N' Symbol: 'i' Code: 111
Prior Symbol: 'N' Symbol: 'o' Code: 101
Prior Symbol: 'N' Symbol: 'u' Code: 110011
Prior Symbol: 'O' Symbol: 27 Code: 010
Prior Symbol: 'O' Symbol: ' ' Code: 001
Prior Symbol: 'O' Symbol: 'd' Code: 01110
Prior Symbol: 'O' Symbol: 'f' Code: 11010
Prior Symbol: 'O' Symbol: 'l' Code: 1100
Prior Symbol: 'O' Symbol: 'n' Code: 10
Prior Symbol: 'O' Symbol: 'p' Code: 0001
Prior Symbol: 'O' Symbol: 'r' Code: 0110
Prior Symbol: 'O' Symbol: 's' Code: 01111
Prior Symbol: 'O' Symbol: 'u' Code: 111
Prior Symbol: 'O' Symbol: 'v' Code: 11011
Prior Symbol: 'O' Symbol: 'w' Code: 0000
Prior Symbol: 'P' Symbol: 27 Code: 111111
Prior Symbol: 'P' Symbol: ' ' Code: 1111100
Prior Symbol: 'P' Symbol: '.' Code: 011001
Prior Symbol: 'P' Symbol: 'G' Code: 111101
Prior Symbol: 'P' Symbol: 'R' Code: 111100
Prior Symbol: 'P' Symbol: 'a' Code: 00
Prior Symbol: 'P' Symbol: 'e' Code: 010
Prior Symbol: 'P' Symbol: 'i' Code: 0111
Prior Symbol: 'P' Symbol: 'l' Code: 1110
Prior Symbol: 'P' Symbol: 'o' Code: 110
Prior Symbol: 'P' Symbol: 'r' Code: 10
Prior Symbol: 'P' Symbol: 's' Code: 1111101
Prior Symbol: 'P' Symbol: 'u' Code: 01101
Prior Symbol: 'P' Symbol: 'y' Code: 011000
Prior Symbol: 'Q' Symbol: 27 Code: 00
Prior Symbol: 'Q' Symbol: 'V' Code: 01
Prior Symbol: 'Q' Symbol: 'u' Code: 1
Prior Symbol: 'R' Symbol: 27 Code: 10001
Prior Symbol: 'R' Symbol: 'a' Code: 101
Prior Symbol: 'R' Symbol: 'e' Code: 11
Prior Symbol: 'R' Symbol: 'h' Code: 10000
Prior Symbol: 'R' Symbol: 'i' Code: 00
Prior Symbol: 'R' Symbol: 'o' Code: 01
Prior Symbol: 'R' Symbol: 'u' Code: 1001
Prior Symbol: 'S' Symbol: 27 Code: 101110
Prior Symbol: 'S' Symbol: ' ' Code: 1110100
Prior Symbol: 'S' Symbol: '*' Code: 1011000
Prior Symbol: 'S' Symbol: '.' Code: 1011011
Prior Symbol: 'S' Symbol: 'a' Code: 1111
Prior Symbol: 'S' Symbol: 'c' Code: 11100
Prior Symbol: 'S' Symbol: 'e' Code: 000
Prior Symbol: 'S' Symbol: 'h' Code: 100
Prior Symbol: 'S' Symbol: 'i' Code: 1100
Prior Symbol: 'S' Symbol: 'k' Code: 101111
Prior Symbol: 'S' Symbol: 'l' Code: 1011001
Prior Symbol: 'S' Symbol: 'm' Code: 1110110
Prior Symbol: 'S' Symbol: 'n' Code: 1110111
Prior Symbol: 'S' Symbol: 'o' Code: 1010
Prior Symbol: 'S' Symbol: 'p' Code: 001
Prior Symbol: 'S' Symbol: 'q' Code: 1011010
Prior Symbol: 'S' Symbol: 't' Code: 01
Prior Symbol: 'S' Symbol: 'u' Code: 1101
Prior Symbol: 'S' Symbol: 'w' Code: 1110101
Prior Symbol: 'T' Symbol: 27 Code: 1111010
Prior Symbol: 'T' Symbol: '-' Code: 11110110
Prior Symbol: 'T' Symbol: 'N' Code: 11110111
Prior Symbol: 'T' Symbol: 'V' Code: 111100
Prior Symbol: 'T' Symbol: 'a' Code: 1010
Prior Symbol: 'T' Symbol: 'e' Code: 1011
Prior Symbol: 'T' Symbol: 'h' Code: 0
Prior Symbol: 'T' Symbol: 'i' Code: 1110
Prior Symbol: 'T' Symbol: 'o' Code: 110
Prior Symbol: 'T' Symbol: 'r' Code: 100
Prior Symbol: 'T' Symbol: 'u' Code: 111110
Prior Symbol: 'T' Symbol: 'w' Code: 111111
Prior Symbol: 'U' Symbol: 27 Code: 101
Prior Symbol: 'U' Symbol: '.' Code: 1001
Prior Symbol: 'U' Symbol: 'l' Code: 1000
Prior Symbol: 'U' Symbol: 'n' Code: 0
Prior Symbol: 'U' Symbol: 'p' Code: 11
Prior Symbol: 'V' Symbol: 0 Code: 000
Prior Symbol: 'V' Symbol: 27 Code: 0011
Prior Symbol: 'V' Symbol: ' ' Code: 01010
Prior Symbol: 'V' Symbol: 'C' Code: 01011
Prior Symbol: 'V' Symbol: 'a' Code: 011
Prior Symbol: 'V' Symbol: 'e' Code: 0100
Prior Symbol: 'V' Symbol: 'i' Code: 1
Prior Symbol: 'V' Symbol: 'o' Code: 0010
Prior Symbol: 'W' Symbol: 27 Code: 00011
Prior Symbol: 'W' Symbol: 'F' Code: 000100
Prior Symbol: 'W' Symbol: 'W' Code: 000101
Prior Symbol: 'W' Symbol: 'a' Code: 111

Prior Symbol: 'W' Symbol: 'e' Code: 110
Prior Symbol: 'W' Symbol: 'h' Code: 001
Prior Symbol: 'W' Symbol: 'i' Code: 01
Prior Symbol: 'W' Symbol: 'o' Code: 10
Prior Symbol: 'W' Symbol: 'r' Code: 0000
Prior Symbol: 'X' Symbol: 27 Code: 1
Prior Symbol: 'Y' Symbol: 27 Code: 001
Prior Symbol: 'Y' Symbol: 'a' Code: 000
Prior Symbol: 'Y' Symbol: 'e' Code: 01
Prior Symbol: 'Y' Symbol: 'o' Code: 1
Prior Symbol: 'Z' Symbol: 27 Code: 00
Prior Symbol: 'Z' Symbol: 'a' Code: 01
Prior Symbol: 'Z' Symbol: 'o' Code: 1
Prior Symbol: '[' Symbol: 27 Code: 1
Prior Symbol: '\' Symbol: 27 Code: 1
Prior Symbol: ']' Symbol: 27 Code: 1
Prior Symbol: '^' Symbol: 27 Code: 1
Prior Symbol: '_' Symbol: 27 Code: 1
Prior Symbol: '`' Symbol: 27 Code: 1
Prior Symbol: 'a' Symbol: 0 Code: 00010
Prior Symbol: 'a' Symbol: 27 Code: 1111010110
Prior Symbol: 'a' Symbol: ' ' Code: 10110
Prior Symbol: 'a' Symbol: '"' Code: 11110100
Prior Symbol: 'a' Symbol: ':' Code: 1111010111
Prior Symbol: 'a' Symbol: 'b' Code: 010010
Prior Symbol: 'a' Symbol: 'c' Code: 11111
Prior Symbol: 'a' Symbol: 'd' Code: 10100
Prior Symbol: 'a' Symbol: 'e' Code: 101011000
Prior Symbol: 'a' Symbol: 'f' Code: 10101101
Prior Symbol: 'a' Symbol: 'g' Code: 01000
Prior Symbol: 'a' Symbol: 'h' Code: 0100111
Prior Symbol: 'a' Symbol: 'i' Code: 10111
Prior Symbol: 'a' Symbol: 'j' Code: 101011001
Prior Symbol: 'a' Symbol: 'k' Code: 101010
Prior Symbol: 'a' Symbol: 'l' Code: 001
Prior Symbol: 'a' Symbol: 'm' Code: 0101
Prior Symbol: 'a' Symbol: 'n' Code: 110
Prior Symbol: 'a' Symbol: 'p' Code: 111100
Prior Symbol: 'a' Symbol: 'r' Code: 100
Prior Symbol: 'a' Symbol: 's' Code: 1110
Prior Symbol: 'a' Symbol: 't' Code: 011
Prior Symbol: 'a' Symbol: 'u' Code: 1111011
Prior Symbol: 'a' Symbol: 'v' Code: 00011
Prior Symbol: 'a' Symbol: 'w' Code: 1010111
Prior Symbol: 'a' Symbol: 'x' Code: 111101010
Prior Symbol: 'a' Symbol: 'y' Code: 0000
Prior Symbol: 'a' Symbol: 'z' Code: 0100110
Prior Symbol: 'b' Symbol: 0 Code: 11111
Prior Symbol: 'b' Symbol: 27 Code: 111101
Prior Symbol: 'b' Symbol: ' ' Code: 0110
Prior Symbol: 'b' Symbol: 'a' Code: 00
Prior Symbol: 'b' Symbol: 'b' Code: 01111
Prior Symbol: 'b' Symbol: 'e' Code: 1010
Prior Symbol: 'b' Symbol: 'i' Code: 1110
Prior Symbol: 'b' Symbol: 'l' Code: 010
Prior Symbol: 'b' Symbol: 'o' Code: 110
Prior Symbol: 'b' Symbol: 'r' Code: 1011
Prior Symbol: 'b' Symbol: 's' Code: 111100
Prior Symbol: 'b' Symbol: 'u' Code: 01110
Prior Symbol: 'b' Symbol: 'y' Code: 100
Prior Symbol: 'c' Symbol: 0 Code: 010110
Prior Symbol: 'c' Symbol: 27 Code: 1000011
Prior Symbol: 'c' Symbol: ' ' Code: 0100
Prior Symbol: 'c' Symbol: 'C' Code: 0010110
Prior Symbol: 'c' Symbol: 'G' Code: 1000010
Prior Symbol: 'c' Symbol: 'L' Code: 0010111
Prior Symbol: 'c' Symbol: 'a' Code: 011
Prior Symbol: 'c' Symbol: 'c' Code: 001010
Prior Symbol: 'c' Symbol: 'e' Code: 111
Prior Symbol: 'c' Symbol: 'h' Code: 101
Prior Symbol: 'c' Symbol: 'i' Code: 0011
Prior Symbol: 'c' Symbol: 'k' Code: 110
Prior Symbol: 'c' Symbol: 'l' Code: 010111
Prior Symbol: 'c' Symbol: 'o' Code: 1001
Prior Symbol: 'c' Symbol: 'r' Code: 10001
Prior Symbol: 'c' Symbol: 's' Code: 00100
Prior Symbol: 'c' Symbol: 't' Code: 000
Prior Symbol: 'c' Symbol: 'u' Code: 01010
Prior Symbol: 'c' Symbol: 'y' Code: 100000
Prior Symbol: 'd' Symbol: 0 Code: 011
Prior Symbol: 'd' Symbol: 27 Code: 101110
Prior Symbol: 'd' Symbol: ' ' Code: 11
Prior Symbol: 'd' Symbol: '.' Code: 101101110
Prior Symbol: 'd' Symbol: 'a' Code: 1010
Prior Symbol: 'd' Symbol: 'd' Code: 100000
Prior Symbol: 'd' Symbol: 'e' Code: 00
Prior Symbol: 'd' Symbol: 'g' Code: 100001
Prior Symbol: 'd' Symbol: 'i' Code: 1001

Prior Symbol: 'd' Symbol: 'l' Code: 1011010
Prior Symbol: 'd' Symbol: 'o' Code: 101111
Prior Symbol: 'd' Symbol: 'r' Code: 101100
Prior Symbol: 'd' Symbol: 's' Code: 0101
Prior Symbol: 'd' Symbol: 'u' Code: 101101111
Prior Symbol: 'd' Symbol: 'v' Code: 10001
Prior Symbol: 'd' Symbol: 'w' Code: 10110110
Prior Symbol: 'd' Symbol: 'y' Code: 0100
Prior Symbol: 'e' Symbol: 0 Code: 001
Prior Symbol: 'e' Symbol: 27 Code: 1010111100
Prior Symbol: 'e' Symbol: ' ' Code: 01
Prior Symbol: 'e' Symbol: '!' Code: 1010111101
Prior Symbol: 'e' Symbol: '"' Code: 10101100
Prior Symbol: 'e' Symbol: '-' Code: 1010111110
Prior Symbol: 'e' Symbol: ':' Code: 00010010
Prior Symbol: 'e' Symbol: 'a' Code: 1000
Prior Symbol: 'e' Symbol: 'b' Code: 10101101
Prior Symbol: 'e' Symbol: 'c' Code: 100111
Prior Symbol: 'e' Symbol: 'd' Code: 00011
Prior Symbol: 'e' Symbol: 'e' Code: 10100
Prior Symbol: 'e' Symbol: 'f' Code: 1001100
Prior Symbol: 'e' Symbol: 'g' Code: 1010100
Prior Symbol: 'e' Symbol: 'h' Code: 1010111111
Prior Symbol: 'e' Symbol: 'i' Code: 10101110
Prior Symbol: 'e' Symbol: 'j' Code: 000100000
Prior Symbol: 'e' Symbol: 'k' Code: 1010101
Prior Symbol: 'e' Symbol: 'l' Code: 10010
Prior Symbol: 'e' Symbol: 'm' Code: 1001101
Prior Symbol: 'e' Symbol: 'n' Code: 1110
Prior Symbol: 'e' Symbol: 'o' Code: 000101
Prior Symbol: 'e' Symbol: 'p' Code: 000001
Prior Symbol: 'e' Symbol: 'q' Code: 000100001
Prior Symbol: 'e' Symbol: 'r' Code: 110
Prior Symbol: 'e' Symbol: 's' Code: 1111
Prior Symbol: 'e' Symbol: 't' Code: 10110
Prior Symbol: 'e' Symbol: 'u' Code: 000100010
Prior Symbol: 'e' Symbol: 'v' Code: 000000
Prior Symbol: 'e' Symbol: 'w' Code: 10111
Prior Symbol: 'e' Symbol: 'x' Code: 00010011
Prior Symbol: 'e' Symbol: 'y' Code: 00001
Prior Symbol: 'e' Symbol: 'z' Code: 000100011
Prior Symbol: 'f' Symbol: 0 Code: 11100
Prior Symbol: 'f' Symbol: 27 Code: 1111001
Prior Symbol: 'f' Symbol: ' ' Code: 0
Prior Symbol: 'f' Symbol: 'a' Code: 11101
Prior Symbol: 'f' Symbol: 'e' Code: 110
Prior Symbol: 'f' Symbol: 'f' Code: 1011
Prior Symbol: 'f' Symbol: 'i' Code: 1001
Prior Symbol: 'f' Symbol: 'l' Code: 111101
Prior Symbol: 'f' Symbol: 'o' Code: 1010
Prior Symbol: 'f' Symbol: 'r' Code: 111111
Prior Symbol: 'f' Symbol: 's' Code: 111110
Prior Symbol: 'f' Symbol: 't' Code: 1000
Prior Symbol: 'f' Symbol: 'u' Code: 1111000
Prior Symbol: 'g' Symbol: 0 Code: 110
Prior Symbol: 'g' Symbol: 27 Code: 1110000
Prior Symbol: 'g' Symbol: ' ' Code: 01
Prior Symbol: 'g' Symbol: '"' Code: 1001100
Prior Symbol: 'g' Symbol: ':' Code: 11100010
Prior Symbol: 'g' Symbol: 'a' Code: 1000
Prior Symbol: 'g' Symbol: 'e' Code: 101
Prior Symbol: 'g' Symbol: 'g' Code: 1111010
Prior Symbol: 'g' Symbol: 'h' Code: 00
Prior Symbol: 'g' Symbol: 'i' Code: 11101
Prior Symbol: 'g' Symbol: 'l' Code: 1111011
Prior Symbol: 'g' Symbol: 'n' Code: 100111
Prior Symbol: 'g' Symbol: 'o' Code: 111001
Prior Symbol: 'g' Symbol: 'r' Code: 10010
Prior Symbol: 'g' Symbol: 's' Code: 11111
Prior Symbol: 'g' Symbol: 't' Code: 1001101
Prior Symbol: 'g' Symbol: 'u' Code: 111100
Prior Symbol: 'g' Symbol: 'y' Code: 11100011
Prior Symbol: 'h' Symbol: 0 Code: 11101
Prior Symbol: 'h' Symbol: 27 Code: 1110001
Prior Symbol: 'h' Symbol: ' ' Code: 1011
Prior Symbol: 'h' Symbol: 'a' Code: 1100
Prior Symbol: 'h' Symbol: 'b' Code: 11100110
Prior Symbol: 'h' Symbol: 'e' Code: 0
Prior Symbol: 'h' Symbol: 'i' Code: 100
Prior Symbol: 'h' Symbol: 'l' Code: 1110010
Prior Symbol: 'h' Symbol: 'n' Code: 101001
Prior Symbol: 'h' Symbol: 'o' Code: 1101
Prior Symbol: 'h' Symbol: 'r' Code: 10101
Prior Symbol: 'h' Symbol: 't' Code: 1111
Prior Symbol: 'h' Symbol: 'u' Code: 11100111
Prior Symbol: 'h' Symbol: 'w' Code: 1110000
Prior Symbol: 'h' Symbol: 'y' Code: 101000
Prior Symbol: 'i' Symbol: 0 Code: 00110101

Prior Symbol: 'i' Symbol: 27 Code: 00110110
Prior Symbol: 'i' Symbol: ' ' Code: 000100
Prior Symbol: 'i' Symbol: '!' Code: 001101000
Prior Symbol: 'i' Symbol: 'a' Code: 00011
Prior Symbol: 'i' Symbol: 'b' Code: 0011000
Prior Symbol: 'i' Symbol: 'c' Code: 1111
Prior Symbol: 'i' Symbol: 'd' Code: 0010
Prior Symbol: 'i' Symbol: 'e' Code: 1101
Prior Symbol: 'i' Symbol: 'f' Code: 00111
Prior Symbol: 'i' Symbol: 'g' Code: 1100
Prior Symbol: 'i' Symbol: 'i' Code: 00110010
Prior Symbol: 'i' Symbol: 'k' Code: 00110011
Prior Symbol: 'i' Symbol: 'l' Code: 0110
Prior Symbol: 'i' Symbol: 'm' Code: 11101
Prior Symbol: 'i' Symbol: 'n' Code: 10
Prior Symbol: 'i' Symbol: 'o' Code: 0100
Prior Symbol: 'i' Symbol: 'p' Code: 000101
Prior Symbol: 'i' Symbol: 'r' Code: 11100
Prior Symbol: 'i' Symbol: 's' Code: 0111
Prior Symbol: 'i' Symbol: 't' Code: 0101
Prior Symbol: 'i' Symbol: 'v' Code: 0000
Prior Symbol: 'i' Symbol: 'x' Code: 001101001
Prior Symbol: 'i' Symbol: 'z' Code: 00110111
Prior Symbol: 'j' Symbol: 27 Code: 10
Prior Symbol: 'j' Symbol: 'a' Code: 11
Prior Symbol: 'j' Symbol: 'o' Code: 0
Prior Symbol: 'k' Symbol: 0 Code: 01
Prior Symbol: 'k' Symbol: 27 Code: 00011
Prior Symbol: 'k' Symbol: ' ' Code: 111
Prior Symbol: 'k' Symbol: ':' Code: 00001
Prior Symbol: 'k' Symbol: 'T' Code: 000000
Prior Symbol: 'k' Symbol: 'a' Code: 001111
Prior Symbol: 'k' Symbol: 'e' Code: 10
Prior Symbol: 'k' Symbol: 'f' Code: 000100
Prior Symbol: 'k' Symbol: 'i' Code: 110
Prior Symbol: 'k' Symbol: 'l' Code: 000101
Prior Symbol: 'k' Symbol: 'o' Code: 000001
Prior Symbol: 'k' Symbol: 's' Code: 0010
Prior Symbol: 'k' Symbol: 'w' Code: 001110
Prior Symbol: 'k' Symbol: 'y' Code: 00110
Prior Symbol: 'l' Symbol: 0 Code: 1000
Prior Symbol: 'l' Symbol: 27 Code: 0111001
Prior Symbol: 'l' Symbol: ' ' Code: 010
Prior Symbol: 'l' Symbol: '"' Code: 01100010
Prior Symbol: 'l' Symbol: '-' Code: 11110011
Prior Symbol: 'l' Symbol: ':' Code: 01100011
Prior Symbol: 'l' Symbol: 'a' Code: 1110
Prior Symbol: 'l' Symbol: 'b' Code: 0110000
Prior Symbol: 'l' Symbol: 'c' Code: 01110000
Prior Symbol: 'l' Symbol: 'd' Code: 000
Prior Symbol: 'l' Symbol: 'e' Code: 110
Prior Symbol: 'l' Symbol: 'f' Code: 1111000
Prior Symbol: 'l' Symbol: 'i' Code: 001
Prior Symbol: 'l' Symbol: 'k' Code: 011001
Prior Symbol: 'l' Symbol: 'l' Code: 101
Prior Symbol: 'l' Symbol: 'm' Code: 1111010
Prior Symbol: 'l' Symbol: 'o' Code: 11111
Prior Symbol: 'l' Symbol: 'r' Code: 11110010
Prior Symbol: 'l' Symbol: 's' Code: 01101
Prior Symbol: 'l' Symbol: 't' Code: 011101
Prior Symbol: 'l' Symbol: 'u' Code: 01111
Prior Symbol: 'l' Symbol: 'v' Code: 1111011
Prior Symbol: 'l' Symbol: 'w' Code: 01110001
Prior Symbol: 'l' Symbol: 'y' Code: 1001
Prior Symbol: 'm' Symbol: 0 Code: 0100
Prior Symbol: 'm' Symbol: 27 Code: 010101
Prior Symbol: 'm' Symbol: ' ' Code: 001
Prior Symbol: 'm' Symbol: 'a' Code: 101
Prior Symbol: 'm' Symbol: 'b' Code: 0000
Prior Symbol: 'm' Symbol: 'e' Code: 11
Prior Symbol: 'm' Symbol: 'i' Code: 011
Prior Symbol: 'm' Symbol: 'm' Code: 0001
Prior Symbol: 'm' Symbol: 'o' Code: 1001
Prior Symbol: 'm' Symbol: 'p' Code: 1000
Prior Symbol: 'm' Symbol: 's' Code: 010111
Prior Symbol: 'm' Symbol: 'u' Code: 010110
Prior Symbol: 'm' Symbol: 'y' Code: 010100
Prior Symbol: 'n' Symbol: 0 Code: 000
Prior Symbol: 'n' Symbol: 27 Code: 01110011
Prior Symbol: 'n' Symbol: ' ' Code: 110
Prior Symbol: 'n' Symbol: '"' Code: 011101
Prior Symbol: 'n' Symbol: ':' Code: 1001010
Prior Symbol: 'n' Symbol: 'a' Code: 11100
Prior Symbol: 'n' Symbol: 'b' Code: 111010000
Prior Symbol: 'n' Symbol: 'c' Code: 01111
Prior Symbol: 'n' Symbol: 'd' Code: 001
Prior Symbol: 'n' Symbol: 'e' Code: 010
Prior Symbol: 'n' Symbol: 'f' Code: 1001011
Prior Symbol: 'n' Symbol: 'g' Code: 101

Prior Symbol: 'n' Symbol: 'h' Code: 111010101
Prior Symbol: 'n' Symbol: 'i' Code: 1000
Prior Symbol: 'n' Symbol: 'j' Code: 111010001
Prior Symbol: 'n' Symbol: 'k' Code: 1110110
Prior Symbol: 'n' Symbol: 'l' Code: 111010110
Prior Symbol: 'n' Symbol: 'm' Code: 111010111
Prior Symbol: 'n' Symbol: 'n' Code: 10011
Prior Symbol: 'n' Symbol: 'o' Code: 1110111
Prior Symbol: 'n' Symbol: 'r' Code: 111010100
Prior Symbol: 'n' Symbol: 's' Code: 0110
Prior Symbol: 'n' Symbol: 't' Code: 1111
Prior Symbol: 'n' Symbol: 'u' Code: 11101001
Prior Symbol: 'n' Symbol: 'v' Code: 0111000
Prior Symbol: 'n' Symbol: 'y' Code: 100100
Prior Symbol: 'n' Symbol: 'z' Code: 01110010
Prior Symbol: 'o' Symbol: 0 Code: 00101
Prior Symbol: 'o' Symbol: 27 Code: 01110001
Prior Symbol: 'o' Symbol: ' ' Code: 0101
Prior Symbol: 'o' Symbol: '"' Code: 01110000
Prior Symbol: 'o' Symbol: '.' Code: 0111011010
Prior Symbol: 'o' Symbol: '?' Code: 011101100
Prior Symbol: 'o' Symbol: 'a' Code: 1100010
Prior Symbol: 'o' Symbol: 'b' Code: 001001
Prior Symbol: 'o' Symbol: 'c' Code: 110000
Prior Symbol: 'o' Symbol: 'd' Code: 01111
Prior Symbol: 'o' Symbol: 'e' Code: 0111001
Prior Symbol: 'o' Symbol: 'f' Code: 1001
Prior Symbol: 'o' Symbol: 'g' Code: 00010
Prior Symbol: 'o' Symbol: 'h' Code: 0111010
Prior Symbol: 'o' Symbol: 'i' Code: 01110111
Prior Symbol: 'o' Symbol: 'k' Code: 1100011
Prior Symbol: 'o' Symbol: 'l' Code: 0100
Prior Symbol: 'o' Symbol: 'm' Code: 1000
Prior Symbol: 'o' Symbol: 'n' Code: 111
Prior Symbol: 'o' Symbol: 'o' Code: 0011
Prior Symbol: 'o' Symbol: 'p' Code: 01101
Prior Symbol: 'o' Symbol: 'r' Code: 101
Prior Symbol: 'o' Symbol: 's' Code: 11001
Prior Symbol: 'o' Symbol: 't' Code: 00011
Prior Symbol: 'o' Symbol: 'u' Code: 1101
Prior Symbol: 'o' Symbol: 'v' Code: 01100
Prior Symbol: 'o' Symbol: 'w' Code: 0000
Prior Symbol: 'o' Symbol: 'x' Code: 0010000
Prior Symbol: 'o' Symbol: 'y' Code: 0010001
Prior Symbol: 'o' Symbol: 'z' Code: 0111011011
Prior Symbol: 'p' Symbol: 0 Code: 1101
Prior Symbol: 'p' Symbol: 27 Code: 101110
Prior Symbol: 'p' Symbol: ' ' Code: 010
Prior Symbol: 'p' Symbol: '"' Code: 1100101
Prior Symbol: 'p' Symbol: 'a' Code: 1001
Prior Symbol: 'p' Symbol: 'd' Code: 101111
Prior Symbol: 'p' Symbol: 'e' Code: 111
Prior Symbol: 'p' Symbol: 'h' Code: 11000
Prior Symbol: 'p' Symbol: 'i' Code: 1010
Prior Symbol: 'p' Symbol: 'l' Code: 0110
Prior Symbol: 'p' Symbol: 'm' Code: 1100100
Prior Symbol: 'p' Symbol: 'o' Code: 00
Prior Symbol: 'p' Symbol: 'p' Code: 0111
Prior Symbol: 'p' Symbol: 'r' Code: 10001
Prior Symbol: 'p' Symbol: 's' Code: 10000
Prior Symbol: 'p' Symbol: 't' Code: 10110
Prior Symbol: 'p' Symbol: 'y' Code: 110011
Prior Symbol: 'q' Symbol: 27 Code: 0
Prior Symbol: 'q' Symbol: 'u' Code: 1
Prior Symbol: 'r' Symbol: 0 Code: 1001
Prior Symbol: 'r' Symbol: 27 Code: 01100101
Prior Symbol: 'r' Symbol: ' ' Code: 1111
Prior Symbol: 'r' Symbol: '"' Code: 0110011
Prior Symbol: 'r' Symbol: ',' Code: 110011101
Prior Symbol: 'r' Symbol: '.' Code: 0111100
Prior Symbol: 'r' Symbol: ':' Code: 110011100
Prior Symbol: 'r' Symbol: 'a' Code: 000
Prior Symbol: 'r' Symbol: 'b' Code: 01111101
Prior Symbol: 'r' Symbol: 'c' Code: 0111111
Prior Symbol: 'r' Symbol: 'd' Code: 11000
Prior Symbol: 'r' Symbol: 'e' Code: 101
Prior Symbol: 'r' Symbol: 'f' Code: 11001111
Prior Symbol: 'r' Symbol: 'g' Code: 0111101
Prior Symbol: 'r' Symbol: 'i' Code: 010
Prior Symbol: 'r' Symbol: 'k' Code: 110010
Prior Symbol: 'r' Symbol: 'l' Code: 0011
Prior Symbol: 'r' Symbol: 'm' Code: 011000
Prior Symbol: 'r' Symbol: 'n' Code: 01101
Prior Symbol: 'r' Symbol: 'o' Code: 1101
Prior Symbol: 'r' Symbol: 'p' Code: 01111100
Prior Symbol: 'r' Symbol: 'r' Code: 01110
Prior Symbol: 'r' Symbol: 's' Code: 1110
Prior Symbol: 'r' Symbol: 't' Code: 1000
Prior Symbol: 'r' Symbol: 'u' Code: 1100110

Prior Symbol: 'r' Symbol: 'v' Code: 01100100
Prior Symbol: 'r' Symbol: 'y' Code: 0010
Prior Symbol: 's' Symbol: 0 Code: 11
Prior Symbol: 's' Symbol: 27 Code: 0010011
Prior Symbol: 's' Symbol: ' ' Code: 01
Prior Symbol: 's' Symbol: '"' Code: 001011010
Prior Symbol: 's' Symbol: ',' Code: 001011011
Prior Symbol: 's' Symbol: '.' Code: 00100101
Prior Symbol: 's' Symbol: ':' Code: 0000001
Prior Symbol: 's' Symbol: '?' Code: 001011100
Prior Symbol: 's' Symbol: 'C' Code: 001011101
Prior Symbol: 's' Symbol: 'H' Code: 001011110
Prior Symbol: 's' Symbol: 'a' Code: 101010
Prior Symbol: 's' Symbol: 'c' Code: 101011
Prior Symbol: 's' Symbol: 'd' Code: 001011111
Prior Symbol: 's' Symbol: 'e' Code: 1011
Prior Symbol: 's' Symbol: 'f' Code: 00000000
Prior Symbol: 's' Symbol: 'h' Code: 00001
Prior Symbol: 's' Symbol: 'i' Code: 0011
Prior Symbol: 's' Symbol: 'k' Code: 000001
Prior Symbol: 's' Symbol: 'l' Code: 00101010
Prior Symbol: 's' Symbol: 'm' Code: 00000001
Prior Symbol: 's' Symbol: 'n' Code: 00101011
Prior Symbol: 's' Symbol: 'o' Code: 10100
Prior Symbol: 's' Symbol: 'p' Code: 001000
Prior Symbol: 's' Symbol: 'r' Code: 00100100
Prior Symbol: 's' Symbol: 's' Code: 0001
Prior Symbol: 's' Symbol: 't' Code: 100
Prior Symbol: 's' Symbol: 'u' Code: 0010100
Prior Symbol: 's' Symbol: 'y' Code: 00101100
Prior Symbol: 't' Symbol: 0 Code: 010
Prior Symbol: 't' Symbol: 27 Code: 11000010
Prior Symbol: 't' Symbol: ' ' Code: 101
Prior Symbol: 't' Symbol: '"' Code: 11000011
Prior Symbol: 't' Symbol: ':' Code: 110110000
Prior Symbol: 't' Symbol: '?' Code: 110110001
Prior Symbol: 't' Symbol: 'a' Code: 0000
Prior Symbol: 't' Symbol: 'b' Code: 100000
Prior Symbol: 't' Symbol: 'c' Code: 1101101
Prior Symbol: 't' Symbol: 'd' Code: 11000000
Prior Symbol: 't' Symbol: 'e' Code: 011
Prior Symbol: 't' Symbol: 'h' Code: 111
Prior Symbol: 't' Symbol: 'i' Code: 001
Prior Symbol: 't' Symbol: 'l' Code: 10001
Prior Symbol: 't' Symbol: 'm' Code: 100001
Prior Symbol: 't' Symbol: 'n' Code: 11011001
Prior Symbol: 't' Symbol: 'o' Code: 1001
Prior Symbol: 't' Symbol: 'r' Code: 11010
Prior Symbol: 't' Symbol: 's' Code: 0001
Prior Symbol: 't' Symbol: 't' Code: 110111
Prior Symbol: 't' Symbol: 'u' Code: 11001
Prior Symbol: 't' Symbol: 'w' Code: 11000001
Prior Symbol: 't' Symbol: 'y' Code: 110001
Prior Symbol: 'u' Symbol: 0 Code: 0011110
Prior Symbol: 'u' Symbol: 27 Code: 000100
Prior Symbol: 'u' Symbol: ' ' Code: 001110
Prior Symbol: 'u' Symbol: 'a' Code: 00110
Prior Symbol: 'u' Symbol: 'b' Code: 10011
Prior Symbol: 'u' Symbol: 'c' Code: 11100
Prior Symbol: 'u' Symbol: 'd' Code: 10000
Prior Symbol: 'u' Symbol: 'e' Code: 0010
Prior Symbol: 'u' Symbol: 'f' Code: 0011111
Prior Symbol: 'u' Symbol: 'g' Code: 11101
Prior Symbol: 'u' Symbol: 'i' Code: 00011
Prior Symbol: 'u' Symbol: 'k' Code: 0001010
Prior Symbol: 'u' Symbol: 'l' Code: 0000
Prior Symbol: 'u' Symbol: 'm' Code: 10010
Prior Symbol: 'u' Symbol: 'n' Code: 110
Prior Symbol: 'u' Symbol: 'p' Code: 10001
Prior Symbol: 'u' Symbol: 'r' Code: 01
Prior Symbol: 'u' Symbol: 's' Code: 101
Prior Symbol: 'u' Symbol: 't' Code: 1111
Prior Symbol: 'u' Symbol: 'z' Code: 0001011
Prior Symbol: 'v' Symbol: 27 Code: 0010
Prior Symbol: 'v' Symbol: 'a' Code: 000
Prior Symbol: 'v' Symbol: 'e' Code: 1
Prior Symbol: 'v' Symbol: 'i' Code: 01
Prior Symbol: 'v' Symbol: 'o' Code: 00111
Prior Symbol: 'v' Symbol: 's' Code: 00110
Prior Symbol: 'w' Symbol: 0 Code: 001
Prior Symbol: 'w' Symbol: 27 Code: 01010
Prior Symbol: 'w' Symbol: ' ' Code: 011
Prior Symbol: 'w' Symbol: '"' Code: 010010
Prior Symbol: 'w' Symbol: 'a' Code: 000
Prior Symbol: 'w' Symbol: 'b' Code: 010011
Prior Symbol: 'w' Symbol: 'c' Code: 010111
Prior Symbol: 'w' Symbol: 'e' Code: 1111
Prior Symbol: 'w' Symbol: 'i' Code: 1100

Prior Symbol: 'w' Symbol: 'l' Code: 010110
Prior Symbol: 'w' Symbol: 'n' Code: 1110
Prior Symbol: 'w' Symbol: 'o' Code: 1101
Prior Symbol: 'w' Symbol: 'r' Code: 01000
Prior Symbol: 'w' Symbol: 's' Code: 10
Prior Symbol: 'x' Symbol: 0 Code: 110
Prior Symbol: 'x' Symbol: 27 Code: 1010
Prior Symbol: 'x' Symbol: ' ' Code: 1011
Prior Symbol: 'x' Symbol: 'a' Code: 000
Prior Symbol: 'x' Symbol: 'e' Code: 001
Prior Symbol: 'x' Symbol: 'i' Code: 100
Prior Symbol: 'x' Symbol: 'p' Code: 111
Prior Symbol: 'x' Symbol: 't' Code: 01
Prior Symbol: 'y' Symbol: 0 Code: 10
Prior Symbol: 'y' Symbol: 27 Code: 111110
Prior Symbol: 'y' Symbol: ' ' Code: 0
Prior Symbol: 'y' Symbol: '!' Code: 1101101
Prior Symbol: 'y' Symbol: '" Code: 110101
Prior Symbol: 'y' Symbol: '-' Code: 11110101
Prior Symbol: 'y' Symbol: 'a' Code: 1101110
Prior Symbol: 'y' Symbol: 'b' Code: 1111011
Prior Symbol: 'y' Symbol: 'c' Code: 11110100
Prior Symbol: 'y' Symbol: 'd' Code: 1100000
Prior Symbol: 'y' Symbol: 'e' Code: 11001
Prior Symbol: 'y' Symbol: 'i' Code: 1100001
Prior Symbol: 'y' Symbol: 'l' Code: 111111
Prior Symbol: 'y' Symbol: 'm' Code: 1101111
Prior Symbol: 'y' Symbol: 'n' Code: 1100010
Prior Symbol: 'y' Symbol: 'o' Code: 1100011
Prior Symbol: 'y' Symbol: 'p' Code: 1101000
Prior Symbol: 'y' Symbol: 's' Code: 1110
Prior Symbol: 'y' Symbol: 't' Code: 1101001
Prior Symbol: 'y' Symbol: 'v' Code: 1101100
Prior Symbol: 'y' Symbol: 'w' Code: 111100
Prior Symbol: 'z' Symbol: 0 Code: 110
Prior Symbol: 'z' Symbol: 27 Code: 100
Prior Symbol: 'z' Symbol: ' ' Code: 000
Prior Symbol: 'z' Symbol: 'a' Code: 01
Prior Symbol: 'z' Symbol: 'e' Code: 1010
Prior Symbol: 'z' Symbol: 'i' Code: 111
Prior Symbol: 'z' Symbol: 'y' Code: 001
Prior Symbol: 'z' Symbol: 'z' Code: 1011
Prior Symbol: '{' Symbol: 27 Code: 1
Prior Symbol: '|' Symbol: 27 Code: 1
Prior Symbol: '}' Symbol: 27 Code: 1
Prior Symbol: '~' Symbol: 27 Code: 1
Prior Symbol: 127 Symbol: 27 Code: 1

Table B.C.5/J.94 – English-language Program Title Decode Table

	42 1	86 1	129 82	173 230
	43 98	87 240	130 2	174 3
0 1	44 1	88 1	131 84	175 244
1 0	45 100	89 242	132 2	176 4
2 1	46 1	90 1	133 126	177 4
3 58	47 102	91 244	134 2	178 4
4 1	48 1	92 2	135 146	179 6
5 60	49 104	93 6	136 2	180 4
6 1	50 1	94 2	137 172	181 12
7 62	51 106	95 18	138 2	182 4
8 1	52 1	96 2	139 186	183 16
9 64	53 108	97 20	140 2	184 4
10 1	54 1	98 2	141 210	185 18
11 66	55 110	99 28	142 2	186 4
12 1	56 1	100 2	143 228	187 20
13 68	57 112	101 40	144 2	188 4
14 1	58 1	102 2	145 250	189 22
15 70	59 114	103 48	146 3	190 4
16 1	60 1	104 2	147 6	191 24
17 72	61 116	105 52	148 3	192 4
18 1	62 1	106 2	149 30	193 26
19 74	63 118	107 54	150 3	194 4
20 1	64 1	108 2	151 38	195 28
21 76	65 120	109 56	152 3	196 4
22 1	66 1	110 2	153 50	197 82
23 78	67 206	111 58	154 3	198 4
24 1	68 1	112 2	155 62	199 106
25 80	69 210	113 60	156 3	200 4
26 1	70 1	114 2	157 82	201 142
27 82	71 212	115 62	158 3	202 4
28 1	72 1	116 2	159 100	203 174
29 84	73 214	117 70	160 3	204 4
30 1	74 1	118 2	161 122	205 238
31 86	75 216	119 72	162 3	206 5
32 1	76 1	120 2	163 148	207 6
33 88	77 218	121 74	164 3	208 5
34 1	78 1	122 2	165 152	209 40
35 90	79 220	123 76	166 3	210 5
36 1	80 1	124 2	167 164	211 68
37 92	81 230	125 78	168 3	212 5
38 1	82 1	126 2	169 200	213 114
39 94	83 232	127 80	170 3	214 5
40 1	84 1	128 2	171 222	215 118
41 96	85 234		172 3	216 5

217 144
218 5
219 190
220 5
221 214
222 6
223 10
224 6
225 68
226 6
227 100
228 6
229 102
230 6
231 154
232 6
233 208
234 6
235 252
236 7
237 34
238 7
239 44
240 7
241 70
242 7
243 84
244 7
245 124
246 7
247 138
248 7
249 140
250 7
251 142
252 7
253 144
254 7
255 146
256 27
257 28
258 180
259 164
260 178
261 183
262 218
263 1
264 209
265 2
266 3
267 155
268 4
269 213
270 217
271 5
272 203
273 214
274 6
275 207
276 7
277 8
278 202
279 9
280 201
281 197
282 198
283 10
284 210
285 196
286 199
287 204
288 208
289 200
290 215
291 206
292 11
293 193
294 12
295 194
296 205
297 195
298 13
299 14
300 15
301 16
302 211
303 17
304 212
305 18
306 19
307 20
308 21
309 22
310 23
311 24
312 25
313 26
314 155
315 155
316 155
317 155
318 155
319 155
320 155
321 155
322 155
323 155
324 155
325 155
326 155
327 155
328 155
329 155
330 155
331 155
332 155
333 155
334 155
335 155
336 155
337 155
338 155
339 155
340 155
341 155
342 155
343 155
344 155
345 155
346 155
347 155
348 155
349 155
350 155
351 155
352 155
353 155
354 155
355 155
356 155
357 155
358 155
359 155
360 155
361 155
362 155
363 155
364 155
365 155
366 155
367 155
368 155
369 155
370 155
371 155
372 155
373 155
374 155
375 155
376 41
377 42
378 216
379 229
380 185
381 1
382 167
383 177
384 236
385 209
386 2
387 173
388 178
389 218
390 227
391 179
392 3
393 228
394 230
395 4
396 155
397 226
398 5
399 6
400 7
401 8
402 9
403 213
404 10
405 214
406 11
407 217
408 12
409 166
410 233
411 203
412 197
413 207
414 13
415 14
416 202
417 201
418 15
419 199
420 16
421 17
422 225
423 18
424 19
425 198
426 210
427 200
428 206
429 193
430 196
431 208
432 204
433 20
434 21
435 239
436 194
437 215
438 22
439 205
440 23
441 244
442 212
443 24
444 25
445 26
446 195
447 211
448 27
449 28
450 29
451 30

452 31
453 32
454 33
455 34
456 35
457 36
458 37
459 38
460 39
461 40
462 1
463 128
464 160
465 155
466 155
467 155
468 155
469 155
470 177
471 155
472 155
473 155
474 155
475 160
476 4
477 243
478 228
479 185
480 1
481 244
482 160
483 155
484 2
485 3
486 155
487 155
488 155
489 155
490 1
491 2
492 155
493 193
494 200
495 211
496 155
497 155
498 155
499 160
500 7
501 8
502 177
503 210
504 211
505 212
506 213
507 173
508 205
509 193
510 1
511 2
512 3
513 160
514 4
515 155
516 5
517 6
518 160
519 5
520 201
521 215
522 211
523 1
524 2
525 155
526 174
527 128
528 3
529 4
530 155
531 155
532 2
533 3
534 173
535 155
536 1
537 128
538 160
539 176
540 4
541 5
542 128
543 155
544 177
545 178
546 160
547 176
548 185
549 1
550 2
551 3
552 2
553 3
554 177
555 186
556 1
557 176
558 155
559 128
560 128
561 1
562 176
563 155
564 155
565 184
566 155
567 155
568 155
569 155
570 155
571 176
572 155
573 160
574 2
575 3
576 177
577 179
578 185
579 176
580 1
581 155
582 155
583 160
584 155
585 155
586 155
587 155
588 155
589 155
590 155
591 155
592 155
593 128
594 155
595 155
596 19
597 20
598 170
599 173
600 174
601 246
602 231
603 244
604 226
605 233
606 1
607 2
608 194
609 240
610 155
611 243
612 227
613 230
614 247
615 3
616 245
617 4
618 5
619 6
620 242
621 7
622 8
623 9
624 10
625 11
626 12
627 228
628 160
629 13
630 236
631 238
632 14
633 237
634 15
635 16
636 17
637 18
638 8
639 9
640 193
641 211
642 155
643 1
644 195
645 2
646 233
647 236
648 3
649 242
650 245
651 4
652 239
653 225
654 5
655 229
656 6
657 7
658 11
659 12
660 193
661 249
662 1
663 194
664 207
665 229
666 245
667 155
668 233
669 2
670 160
671 3
672 4
673 5
674 242
675 6
676 236
677 7
678 225
679 8
680 9
681 232
682 10
683 239
684 5
685 6
686 249

687 155
688 1
689 245
690 2
691 242
692 233
693 229
694 239
695 3
696 225
697 4
698 10
699 11
700 241
701 245
702 243
703 1
704 237
705 249
706 195
707 2
708 236
709 238
710 228
711 248
712 3
713 155
714 246
715 4
716 5
717 225
718 6
719 7
720 8
721 9
722 7
723 8
724 160
725 155
726 204
727 1
728 229
729 2
730 236
731 245
732 239
733 3
734 233
735 242
736 4
737 5
738 225
739 6
740 9
741 10
742 174
743 236
744 249
745 193
746 232
747 1
748 155
749 2
750 3
751 4
752 225
753 245
754 233
755 5
756 229
757 6
758 242
759 239
760 7
761 8
762 239
763 5
764 128
765 155
766 245
767 1
768 2
769 233
770 225
771 3
772 229
773 4
774 238
775 11
776 186
777 212
778 174
779 242
780 227
781 1
782 160
783 2
784 128
785 155
786 237
787 3
788 201
789 243
790 244
791 4
792 5
793 6
794 7
795 8
796 9
797 10
798 2
799 3
800 155
801 245
802 1
803 225
804 239
805 229
806 5
807 233
808 225
809 239
810 245
811 238
812 155
813 229
814 1
815 2
816 3
817 4
818 4
819 5
820 160
821 155
822 1
823 245
824 2
825 229
826 239
827 3
828 225
829 233
830 8
831 9
832 170
833 212
834 1
835 155
836 227
837 2
838 242
839 3
840 229
841 4
842 245
843 249
844 233
845 5
846 239
847 6
848 7
849 225
850 229
851 8
852 206
853 160
854 198
855 245
856 1
857 2
858 155
859 194
860 3
861 225
862 4
863 239
864 5
865 233
866 6
867 7
868 9
869 10
870 228
871 243
872 230
873 246
874 247
875 240
876 242
877 1
878 236
879 2
880 3
881 160
882 155
883 4
884 5
885 245
886 6
887 7
888 238
889 8
890 11
891 12
892 160
893 243
894 249
895 174
896 210
897 199
898 1
899 155
900 2
901 245
902 3
903 4
904 5
905 233
906 236
907 6
908 229
909 7
910 239
911 8
912 225
913 9
914 242
915 10
916 1
917 245
918 155
919 214
920 4
921 5

922 232
923 155
924 1
925 245
926 2
927 225
928 233
929 239
930 3
931 229
932 16
933 17
934 170
935 236
936 241
937 174
938 160
939 247
940 237
941 238
942 1
943 2
944 155
945 235
946 3
947 4
948 5
949 6
950 227
951 7
952 239
953 8
954 233
955 245
956 9
957 225
958 229
959 240
960 232
961 10
962 11
963 12
964 13
965 244
966 14
967 15
968 232
969 10
970 173
971 206
972 155
973 1
974 214
975 2
976 245
977 247
978 3
979 4
980 225
981 229
982 233
983 5
984 242
985 6
986 239
987 7
988 8
989 9
990 238
991 3
992 236
993 174
994 1
995 155
996 2
997 240
998 6
999 233
1000 160
1001 195
1002 239
1003 155
1004 229
1005 1
1006 128
1007 2
1008 3
1009 225
1010 4
1011 5
1012 6
1013 7
1014 198
1015 215
1016 1
1017 155
1018 242
1019 2
1020 3
1021 232
1022 229
1023 225
1024 4
1025 233
1026 239
1027 5
1028 155
1029 155
1030 2
1031 239
1032 225
1033 155
1034 1
1035 229
1036 1
1037 239
1038 155
1039 225
1040 155
1041 155
1042 155
1043 155
1044 155
1045 155
1046 155
1047 155
1048 155
1049 155
1050 155
1051 155
1052 25
1053 26
1054 155
1055 186
1056 229
1057 234
1058 248
1059 1
1060 2
1061 230
1062 167
1063 3
1064 250
1065 232
1066 4
1067 247
1068 5
1069 245
1070 226
1071 6
1072 235
1073 7
1074 240
1075 8
1076 128
1077 246
1078 231
1079 9
1080 228
1081 10
1082 160
1083 233
1084 11
1085 227
1086 249
1087 12
1088 13
1089 237
1090 14
1091 15
1092 243
1093 16
1094 17
1095 236
1096 18
1097 244
1098 242
1099 19
1100 238
1101 20
1102 21
1103 22
1104 23
1105 24
1106 10
1107 11
1108 243
1109 155
1110 245
1111 226
1112 1
1113 128
1114 160
1115 2
1116 229
1117 242
1118 233
1119 3
1120 236
1121 4
1122 249
1123 5
1124 239
1125 6
1126 225
1127 7
1128 8
1129 9
1130 16
1131 17
1132 195
1133 204
1134 199
1135 155
1136 227
1137 1
1138 128
1139 236
1140 249
1141 2
1142 243
1143 3
1144 245
1145 4
1146 5
1147 242
1148 6
1149 233
1150 160
1151 7
1152 8
1153 239
1154 244
1155 9
1156 10

1157 225	1204 234	1251 243	1298 3	1345 10
1158 11	1205 241	1252 23	1299 239	1346 11
1159 232	1206 245	1253 128	1300 245	1347 12
1160 235	1207 250	1254 24	1301 4	1348 21
1161 229	1208 1	1255 25	1302 242	1349 22
1162 12	1209 2	1256 242	1303 5	1350 161
1163 13	1210 3	1257 26	1304 6	1351 248
1164 14	1211 4	1258 27	1305 233	1352 233
1165 15	1212 186	1259 160	1306 7	1353 235
1166 14	1213 248	1260 28	1307 243	1354 1
1167 15	1214 167	1261 29	1308 225	1355 128
1168 174	1215 226	1262 160	1309 8	1356 155
1169 245	1216 233	1263 11	1310 9	1357 250
1170 247	1217 5	1264 245	1311 10	1358 226
1171 1	1218 6	1265 155	1312 11	1359 2
1172 236	1219 7	1266 1	1313 229	1360 3
1173 2	1220 230	1267 236	1314 128	1361 4
1174 228	1221 237	1268 243	1315 12	1362 160
1175 231	1222 231	1269 242	1316 232	1363 240
1176 242	1223 235	1270 128	1317 160	1364 5
1177 3	1224 8	1271 225	1318 13	1365 6
1178 155	1225 9	1272 2	1319 14	1366 7
1179 239	1226 246	1273 3	1320 229	1367 225
1180 4	1227 240	1274 244	1321 13	1368 8
1181 246	1228 10	1275 233	1322 226	1369 230
1182 5	1229 239	1276 239	1323 245	1370 242
1183 6	1230 11	1277 230	1324 247	1371 237
1184 249	1231 227	1278 4	1325 155	1372 246
1185 243	1232 12	1279 5	1326 236	1373 9
1186 7	1233 13	1280 6	1327 1	1374 228
1187 233	1234 14	1281 7	1328 249	1375 10
1188 225	1235 249	1282 229	1329 238	1376 239
1189 8	1236 15	1283 8	1330 2	1377 244
1190 9	1237 228	1284 9	1331 3	1378 236
1191 128	1238 236	1285 10	1332 4	1379 243
1192 10	1239 16	1286 15	1333 242	1380 231
1193 11	1240 229	1287 16	1334 5	1381 229
1194 229	1241 17	1288 186	1335 128	1382 11
1195 12	1242 244	1289 249	1336 6	1383 227
1196 13	1243 247	1290 167	1337 160	1384 12
1197 160	1244 18	1291 244	1338 225	1385 13
1198 30	1245 19	1292 155	1339 239	1386 14
1199 31	1246 225	1293 1	1340 7	1387 15
1200 155	1247 20	1294 231	1341 244	1388 16
1201 161	1248 21	1295 236	1342 233	1389 17
1202 173	1249 22	1296 2	1343 8	1390 18
1203 232	1250 238	1297 238	1344 9	1391 19

1392 238
1393 20
1394 239
1395 1
1396 155
1397 225
1398 11
1399 12
1400 212
1401 239
1402 230
1403 236
1404 247
1405 225
1406 1
1407 186
1408 2
1409 155
1410 249
1411 3
1412 4
1413 5
1414 243
1415 6
1416 7
1417 8
1418 233
1419 160
1420 9
1421 128
1422 229
1423 10
1424 21
1425 22
1426 167
1427 186
1428 227
1429 247
1430 242
1431 173
1432 226
1433 1
1434 2
1435 155
1436 230
1437 3
1438 237
1439 246
1440 4
1441 235
1442 5
1443 244
1444 6
1445 7
1446 8
1447 243
1448 9
1449 245
1450 10
1451 239
1452 11
1453 12
1454 128
1455 249
1456 225
1457 13
1458 228
1459 233
1460 160
1461 14
1462 15
1463 236
1464 229
1465 16
1466 17
1467 18
1468 19
1469 20
1470 10
1471 11
1472 249
1473 155
1474 245
1475 243
1476 1
1477 2
1478 226
1479 237
1480 128
1481 3
1482 240
1483 239
1484 4
1485 160
1486 5
1487 233
1488 6
1489 225
1490 7
1491 8
1492 9
1493 229
1494 24
1495 25
1496 226
1497 234
1498 242
1499 232
1500 236
1501 237
1502 250
1503 155
1504 1
1505 245
1506 2
1507 3
1508 246
1509 4
1510 186
1511 230
1512 5
1513 6
1514 235
1515 239
1516 7
1517 167
1518 249
1519 8
1520 9
1521 10
1522 11
1523 227
1524 12
1525 238
1526 225
1527 13
1528 243
1529 14
1530 233
1531 15
1532 16
1533 244
1534 128
1535 228
1536 229
1537 17
1538 18
1539 231
1540 160
1541 19
1542 20
1543 21
1544 22
1545 23
1546 27
1547 28
1548 174
1549 250
1550 191
1551 1
1552 167
1553 155
1554 2
1555 233
1556 248
1557 249
1558 3
1559 229
1560 232
1561 4
1562 225
1563 235
1564 5
1565 226
1566 6
1567 7
1568 227
1569 8
1570 231
1571 244
1572 9
1573 128
1574 246
1575 240
1576 10
1577 228
1578 11
1579 243
1580 247
1581 12
1582 13
1583 239
1584 236
1585 160
1586 14
1587 15
1588 237
1589 230
1590 16
1591 245
1592 17
1593 18
1594 19
1595 20
1596 21
1597 242
1598 22
1599 238
1600 23
1601 24
1602 25
1603 26
1604 14
1605 15
1606 237
1607 167
1608 155
1609 228
1610 1
1611 249
1612 243
1613 242
1614 244
1615 2
1616 232
1617 3
1618 236
1619 240
1620 4
1621 225
1622 233
1623 5
1624 6
1625 128
1626 160

1627 7
1628 8
1629 9
1630 10
1631 229
1632 239
1633 11
1634 12
1635 13
1636 155
1637 245
1638 24
1639 25
1640 186
1641 172
1642 246
1643 155
1644 240
1645 226
1646 1
1647 230
1648 2
1649 167
1650 174
1651 231
1652 3
1653 227
1654 245
1655 4
1656 237
1657 5
1658 6
1659 7
1660 235
1661 8
1662 9
1663 238
1664 242
1665 10
1666 228
1667 11
1668 249
1669 236
1670 12
1671 13
1672 244
1673 128
1674 14
1675 239
1676 243
1677 160
1678 225
1679 15
1680 233
1681 16
1682 17
1683 229
1684 18
1685 19
1686 20
1687 21
1688 22
1689 23
1690 25
1691 26
1692 167
1693 172
1694 191
1695 195
1696 200
1697 228
1698 230
1699 237
1700 242
1701 174
1702 236
1703 238
1704 249
1705 1
1706 2
1707 3
1708 4
1709 186
1710 5
1711 155
1712 245
1713 6
1714 7
1715 8
1716 9
1717 235
1718 240
1719 10
1720 11
1721 12
1722 225
1723 227
1724 13
1725 232
1726 14
1727 15
1728 239
1729 16
1730 17
1731 243
1732 18
1733 233
1734 19
1735 229
1736 20
1737 21
1738 244
1739 22
1740 23
1741 160
1742 24
1743 128
1744 20
1745 21
1746 186
1747 191
1748 228
1749 247
1750 155
1751 167
1752 1
1753 238
1754 2
1755 3
1756 4
1757 227
1758 226
1759 237
1760 5
1761 249
1762 6
1763 244
1764 7
1765 236
1766 8
1767 245
1768 242
1769 9
1770 225
1771 243
1772 10
1773 239
1774 11
1775 12
1776 13
1777 233
1778 128
1779 229
1780 14
1781 160
1782 15
1783 232
1784 16
1785 17
1786 18
1787 19
1788 17
1789 18
1790 235
1791 250
1792 128
1793 230
1794 155
1795 1
1796 160
1797 2
1798 3
1799 233
1800 225
1801 4
1802 228
1803 240
1804 237
1805 226
1806 227
1807 231
1808 236
1809 5
1810 229
1811 6
1812 7
1813 8
1814 9
1815 244
1816 10
1817 11
1818 12
1819 243
1820 238
1821 13
1822 14
1823 242
1824 15
1825 16
1826 4
1827 229
1828 243
1829 239
1830 155
1831 1
1832 225
1833 2
1834 3
1835 233
1836 11
1837 12
1838 167
1839 226
1840 236
1841 227
1842 242
1843 1
1844 155
1845 2
1846 3
1847 4
1848 233
1849 239
1850 238
1851 229
1852 225
1853 128
1854 5
1855 160
1856 6
1857 7
1858 8
1859 9
1860 243
1861 10

1862 5
1863 6
1864 155
1865 160
1866 225
1867 229
1868 233
1869 1
1870 128
1871 240
1872 2
1873 244
1874 3
1875 4
1876 160
1877 19
1878 227
1879 173
1880 228
1881 233
1882 238
1883 239
1884 240
1885 244
1886 246
1887 161
1888 225
1889 237
1890 1
1891 226
1892 2
1893 3
1894 4
1895 167
1896 5
1897 6
1898 247
1899 7
1900 155
1901 236
1902 8
1903 229
1904 9
1905 10
1906 11
1907 12
1908 13
1909 14
1910 243
1911 15
1912 16
1913 17
1914 128
1915 18
1916 5
1917 6
1918 229
1919 250
1920 160
1921 249
1922 155
1923 1
1924 128
1925 233
1926 2
1927 225
1928 3
1929 4
1930 155
1931 155
1932 155
1933 155
1934 155
1935 155
1936 155
1937 155
1938 155
1939 155

B.C.3 Standard compression Type 2 Huffman Encode/Decode tables

The following encode/decode tables (Tables B.C.6 and B.C.7) 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 B.C.6/J.94 – English-language Program Description Encode Table

Prior Symbol: 0 Symbol: 27 Code: 1110000
Prior Symbol: 0 Symbol: "" Code: 111001
Prior Symbol: 0 Symbol: 'A' Code: 010
Prior Symbol: 0 Symbol: 'B' Code: 0011
Prior Symbol: 0 Symbol: 'C' Code: 0111
Prior Symbol: 0 Symbol: 'D' Code: 11101
Prior Symbol: 0 Symbol: 'E' Code: 10010
Prior Symbol: 0 Symbol: 'F' Code: 10110
Prior Symbol: 0 Symbol: 'G' Code: 011011
Prior Symbol: 0 Symbol: 'H' Code: 10111
Prior Symbol: 0 Symbol: 'I' Code: 011000
Prior Symbol: 0 Symbol: 'J' Code: 1100
Prior Symbol: 0 Symbol: 'K' Code: 00101
Prior Symbol: 0 Symbol: 'L' Code: 10011
Prior Symbol: 0 Symbol: 'M' Code: 1111
Prior Symbol: 0 Symbol: 'N' Code: 00100
Prior Symbol: 0 Symbol: 'O' Code: 011001
Prior Symbol: 0 Symbol: 'P' Code: 000
Prior Symbol: 0 Symbol: 'R' Code: 1000
Prior Symbol: 0 Symbol: 'S' Code: 1010
Prior Symbol: 0 Symbol: 'T' Code: 1101
Prior Symbol: 0 Symbol: 'V' Code: 1110001
Prior Symbol: 0 Symbol: 'W' Code: 011010
Prior Symbol: 1 Symbol: 27 Code: 1
Prior Symbol: 2 Symbol: 27 Code: 1
Prior Symbol: 3 Symbol: 27 Code: 1
Prior Symbol: 4 Symbol: 27 Code: 1
Prior Symbol: 5 Symbol: 27 Code: 1
Prior Symbol: 6 Symbol: 27 Code: 1
Prior Symbol: 7 Symbol: 27 Code: 1
Prior Symbol: 8 Symbol: 27 Code: 1
Prior Symbol: 9 Symbol: 27 Code: 1
Prior Symbol: 10 Symbol: 27 Code: 1
Prior Symbol: 11 Symbol: 27 Code: 1
Prior Symbol: 12 Symbol: 27 Code: 1
Prior Symbol: 13 Symbol: 27 Code: 1
Prior Symbol: 14 Symbol: 27 Code: 1
Prior Symbol: 15 Symbol: 27 Code: 1
Prior Symbol: 16 Symbol: 27 Code: 1
Prior Symbol: 17 Symbol: 27 Code: 1
Prior Symbol: 18 Symbol: 27 Code: 1
Prior Symbol: 19 Symbol: 27 Code: 1
Prior Symbol: 20 Symbol: 27 Code: 1
Prior Symbol: 21 Symbol: 27 Code: 1
Prior Symbol: 22 Symbol: 27 Code: 1
Prior Symbol: 23 Symbol: 27 Code: 1
Prior Symbol: 24 Symbol: 27 Code: 1
Prior Symbol: 25 Symbol: 27 Code: 1
Prior Symbol: 26 Symbol: 27 Code: 1
Prior Symbol: 27 Symbol: 27 Code: 1
Prior Symbol: 28 Symbol: 27 Code: 1
Prior Symbol: 29 Symbol: 27 Code: 1
Prior Symbol: 30 Symbol: 27 Code: 1
Prior Symbol: 31 Symbol: 27 Code: 1
Prior Symbol: ' ' Symbol: 27 Code: 101000001
Prior Symbol: ' ' Symbol: "" Code: 111111010
Prior Symbol: ' ' Symbol: '(' Code: 1111111100
Prior Symbol: ' ' Symbol: '-' Code: 11111111110
Prior Symbol: ' ' Symbol: '/' Code: 11111111111
Prior Symbol: ' ' Symbol: '1' Code: 0101011
Prior Symbol: ' ' Symbol: '2' Code: 0100010
Prior Symbol: ' ' Symbol: '3' Code: 1111111101
Prior Symbol: ' ' Symbol: '4' Code: 110010100
Prior Symbol: ' ' Symbol: '5' Code: 1111111110
Prior Symbol: ' ' Symbol: '7' Code: 1010000000
Prior Symbol: ' ' Symbol: 'A' Code: 10010
Prior Symbol: ' ' Symbol: 'B' Code: 010100
Prior Symbol: ' ' Symbol: 'C' Code: 111100
Prior Symbol: ' ' Symbol: 'D' Code: 1111010
Prior Symbol: ' ' Symbol: 'E' Code: 0100011

Prior Symbol: ' ' Symbol: 'F' Code: 0101010
Prior Symbol: ' ' Symbol: 'G' Code: 000010
Prior Symbol: ' ' Symbol: 'H' Code: 1111011
Prior Symbol: ' ' Symbol: 'I' Code: 11001011
Prior Symbol: ' ' Symbol: 'J' Code: 000011
Prior Symbol: ' ' Symbol: 'K' Code: 1100100
Prior Symbol: ' ' Symbol: 'L' Code: 010110
Prior Symbol: ' ' Symbol: 'M' Code: 101001
Prior Symbol: ' ' Symbol: 'N' Code: 001100
Prior Symbol: ' ' Symbol: 'O' Code: 10100001
Prior Symbol: ' ' Symbol: 'P' Code: 001101
Prior Symbol: ' ' Symbol: 'R' Code: 1111100
Prior Symbol: ' ' Symbol: 'S' Code: 01001
Prior Symbol: ' ' Symbol: 'T' Code: 1100110
Prior Symbol: ' ' Symbol: 'U' Code: 111111011
Prior Symbol: ' ' Symbol: 'V' Code: 111111100
Prior Symbol: ' ' Symbol: 'W' Code: 010000
Prior Symbol: ' ' Symbol: 'Y' Code: 111111101
Prior Symbol: ' ' Symbol: 'Z' Code: 1010000001
Prior Symbol: ' ' Symbol: 'a' Code: 011
Prior Symbol: ' ' Symbol: 'b' Code: 10111
Prior Symbol: ' ' Symbol: 'c' Code: 10011
Prior Symbol: ' ' Symbol: 'd' Code: 10000
Prior Symbol: ' ' Symbol: 'e' Code: 100010
Prior Symbol: ' ' Symbol: 'f' Code: 11101
Prior Symbol: ' ' Symbol: 'g' Code: 100011
Prior Symbol: ' ' Symbol: 'h' Code: 0001
Prior Symbol: ' ' Symbol: 'i' Code: 10101
Prior Symbol: ' ' Symbol: 'j' Code: 11001111
Prior Symbol: ' ' Symbol: 'k' Code: 11111010
Prior Symbol: ' ' Symbol: 'l' Code: 010111
Prior Symbol: ' ' Symbol: 'm' Code: 00000
Prior Symbol: ' ' Symbol: 'n' Code: 1010001
Prior Symbol: ' ' Symbol: 'o' Code: 0010
Prior Symbol: ' ' Symbol: 'p' Code: 10110
Prior Symbol: ' ' Symbol: 'q' Code: 110010101
Prior Symbol: ' ' Symbol: 'r' Code: 00111
Prior Symbol: ' ' Symbol: 's' Code: 11100
Prior Symbol: ' ' Symbol: 't' Code: 1101
Prior Symbol: ' ' Symbol: 'u' Code: 11111011
Prior Symbol: ' ' Symbol: 'v' Code: 11111100
Prior Symbol: ' ' Symbol: 'w' Code: 11000
Prior Symbol: ' ' Symbol: 'y' Code: 11001110
Prior Symbol: '!' Symbol: 27 Code: 1
Prior Symbol: '"' Symbol: 0 Code: 000
Prior Symbol: '"' Symbol: 27 Code: 10
Prior Symbol: '"' Symbol: ' ' Code: 11
Prior Symbol: '"' Symbol: '.' Code: 001
Prior Symbol: '"' Symbol: 'H' Code: 010
Prior Symbol: '"' Symbol: 'T' Code: 011
Prior Symbol: '#' Symbol: 27 Code: 1
Prior Symbol: '$' Symbol: 27 Code: 1
Prior Symbol: '%' Symbol: 27 Code: 1
Prior Symbol: '&' Symbol: 27 Code: 1
Prior Symbol: ''' Symbol: 27 Code: 00
Prior Symbol: ''' Symbol: ' ' Code: 010
Prior Symbol: ''' Symbol: 's' Code: 1
Prior Symbol: ''' Symbol: 't' Code: 011
Prior Symbol: '(' Symbol: 27 Code: 1
Prior Symbol: ')' Symbol: 27 Code: 1
Prior Symbol: ')' Symbol: ',' Code: 0
Prior Symbol: '*' Symbol: 27 Code: 1
Prior Symbol: '+' Symbol: 27 Code: 1
Prior Symbol: ',' Symbol: 27 Code: 00
Prior Symbol: ',' Symbol: ' ' Code: 1
Prior Symbol: ',' Symbol: '"' Code: 01
Prior Symbol: '-' Symbol: 27 Code: 10
Prior Symbol: '-' Symbol: ' ' Code: 1110
Prior Symbol: '-' Symbol: 'a' Code: 000
Prior Symbol: '-' Symbol: 'b' Code: 0010
Prior Symbol: '-' Symbol: 'c' Code: 110
Prior Symbol: '-' Symbol: 'd' Code: 0011
Prior Symbol: '-' Symbol: 'e' Code: 0100
Prior Symbol: '-' Symbol: 'f' Code: 0101
Prior Symbol: '-' Symbol: 'r' Code: 1111
Prior Symbol: '-' Symbol: 's' Code: 011
Prior Symbol: '.' Symbol: 0 Code: 1
Prior Symbol: '.' Symbol: 27 Code: 000
Prior Symbol: '.' Symbol: ' ' Code: 01
Prior Symbol: '.' Symbol: '"' Code: 0010
Prior Symbol: '.' Symbol: 'J' Code: 00110
Prior Symbol: '.' Symbol: 'S' Code: 00111
Prior Symbol: '/' Symbol: 27 Code: 0
Prior Symbol: '/' Symbol: ' ' Code: 1
Prior Symbol: '0' Symbol: 27 Code: 100
Prior Symbol: '0' Symbol: ' ' Code: 111
Prior Symbol: '0' Symbol: '0' Code: 00
Prior Symbol: '0' Symbol: '7' Code: 101

Prior Symbol: '0' Symbol: 's' Code: 01
Prior Symbol: '0' Symbol: 't' Code: 110
Prior Symbol: '1' Symbol: 27 Code: 111
Prior Symbol: '1' Symbol: ' ' Code: 10
Prior Symbol: '1' Symbol: '8' Code: 110
Prior Symbol: '1' Symbol: '9' Code: 0
Prior Symbol: '2' Symbol: 27 Code: 101
Prior Symbol: '2' Symbol: ' ' Code: 11
Prior Symbol: '2' Symbol: '.' Code: 0
Prior Symbol: '2' Symbol: '6' Code: 100
Prior Symbol: '3' Symbol: 27 Code: 10
Prior Symbol: '3' Symbol: ' ' Code: 0
Prior Symbol: '3' Symbol: '0' Code: 11
Prior Symbol: '4' Symbol: 27 Code: 10
Prior Symbol: '4' Symbol: ' ' Code: 11
Prior Symbol: '4' Symbol: '.' Code: 0
Prior Symbol: '5' Symbol: 27 Code: 11
Prior Symbol: '5' Symbol: ' ' Code: 10
Prior Symbol: '5' Symbol: '.' Code: 0
Prior Symbol: '6' Symbol: 27 Code: 1
Prior Symbol: '7' Symbol: 27 Code: 0
Prior Symbol: '7' Symbol: ',' Code: 10
Prior Symbol: '7' Symbol: '.' Code: 11
Prior Symbol: '8' Symbol: 27 Code: 1
Prior Symbol: '9' Symbol: 27 Code: 110
Prior Symbol: '9' Symbol: ' ' Code: 111
Prior Symbol: '9' Symbol: '5' Code: 00
Prior Symbol: '9' Symbol: '6' Code: 01
Prior Symbol: '9' Symbol: '8' Code: 10
Prior Symbol: ':' Symbol: 27 Code: 0
Prior Symbol: ':' Symbol: ' ' Code: 1
Prior Symbol: ';' Symbol: 27 Code: 0
Prior Symbol: ';' Symbol: ' ' Code: 1
Prior Symbol: '<' Symbol: 27 Code: 1
Prior Symbol: '=' Symbol: 27 Code: 1
Prior Symbol: '>' Symbol: 27 Code: 1
Prior Symbol: '?' Symbol: 27 Code: 0
Prior Symbol: '?' Symbol: ' ' Code: 1
Prior Symbol: '@' Symbol: 27 Code: 1
Prior Symbol: 'A' Symbol: 27 Code: 10010
Prior Symbol: 'A' Symbol: ' ' Code: 11
Prior Symbol: 'A' Symbol: 'd' Code: 10011
Prior Symbol: 'A' Symbol: 'f' Code: 101000
Prior Symbol: 'A' Symbol: 'l' Code: 00
Prior Symbol: 'A' Symbol: 'm' Code: 10101
Prior Symbol: 'A' Symbol: 'n' Code: 01
Prior Symbol: 'A' Symbol: 'r' Code: 1011
Prior Symbol: 'A' Symbol: 's' Code: 10000
Prior Symbol: 'A' Symbol: 't' Code: 10001
Prior Symbol: 'A' Symbol: 'u' Code: 101001
Prior Symbol: 'B' Symbol: 27 Code: 10010
Prior Symbol: 'B' Symbol: 'a' Code: 101
Prior Symbol: 'B' Symbol: 'e' Code: 111
Prior Symbol: 'B' Symbol: 'i' Code: 00
Prior Symbol: 'B' Symbol: 'l' Code: 10011
Prior Symbol: 'B' Symbol: 'o' Code: 110
Prior Symbol: 'B' Symbol: 'r' Code: 01
Prior Symbol: 'B' Symbol: 'u' Code: 1000
Prior Symbol: 'C' Symbol: 27 Code: 01110
Prior Symbol: 'C' Symbol: 'a' Code: 00
Prior Symbol: 'C' Symbol: 'h' Code: 10
Prior Symbol: 'C' Symbol: 'i' Code: 01111
Prior Symbol: 'C' Symbol: 'l' Code: 110
Prior Symbol: 'C' Symbol: 'o' Code: 111
Prior Symbol: 'C' Symbol: 'r' Code: 0101
Prior Symbol: 'C' Symbol: 'u' Code: 0110
Prior Symbol: 'C' Symbol: 'y' Code: 0100
Prior Symbol: 'D' Symbol: 27 Code: 1111
Prior Symbol: 'D' Symbol: 'a' Code: 01
Prior Symbol: 'D' Symbol: 'e' Code: 100
Prior Symbol: 'D' Symbol: 'i' Code: 00
Prior Symbol: 'D' Symbol: 'o' Code: 101
Prior Symbol: 'D' Symbol: 'r' Code: 1101
Prior Symbol: 'D' Symbol: 'u' Code: 1110
Prior Symbol: 'D' Symbol: 'y' Code: 1100
Prior Symbol: 'E' Symbol: 27 Code: 10
Prior Symbol: 'E' Symbol: 'a' Code: 0110
Prior Symbol: 'E' Symbol: 'd' Code: 000
Prior Symbol: 'E' Symbol: 'i' Code: 0111
Prior Symbol: 'E' Symbol: 'l' Code: 001
Prior Symbol: 'E' Symbol: 'n' Code: 1100
Prior Symbol: 'E' Symbol: 'r' Code: 111
Prior Symbol: 'E' Symbol: 's' Code: 010
Prior Symbol: 'E' Symbol: 'v' Code: 1101
Prior Symbol: 'F' Symbol: 27 Code: 00
Prior Symbol: 'F' Symbol: 'e' Code: 100
Prior Symbol: 'F' Symbol: 'l' Code: 101
Prior Symbol: 'F' Symbol: 'o' Code: 01

Prior Symbol: 'F' Symbol: 'r' Code: 11
Prior Symbol: 'G' Symbol: 27 Code: 000
Prior Symbol: 'G' Symbol: 'a' Code: 110
Prior Symbol: 'G' Symbol: 'e' Code: 01
Prior Symbol: 'G' Symbol: 'i' Code: 100
Prior Symbol: 'G' Symbol: 'l' Code: 001
Prior Symbol: 'G' Symbol: 'o' Code: 1011
Prior Symbol: 'G' Symbol: 'r' Code: 111
Prior Symbol: 'G' Symbol: 'u' Code: 1010
Prior Symbol: 'H' Symbol: 27 Code: 010
Prior Symbol: 'H' Symbol: 'a' Code: 00
Prior Symbol: 'H' Symbol: 'e' Code: 011
Prior Symbol: 'H' Symbol: 'i' Code: 110
Prior Symbol: 'H' Symbol: 'o' Code: 10
Prior Symbol: 'H' Symbol: 'u' Code: 111
Prior Symbol: 'I' Symbol: 27 Code: 011
Prior Symbol: 'I' Symbol: ' ' Code: 000
Prior Symbol: 'I' Symbol: '.' Code: 100
Prior Symbol: 'I' Symbol: 'l' Code: 001
Prior Symbol: 'I' Symbol: 'n' Code: 11
Prior Symbol: 'I' Symbol: 'r' Code: 101
Prior Symbol: 'I' Symbol: 's' Code: 010
Prior Symbol: 'J' Symbol: 27 Code: 1000
Prior Symbol: 'J' Symbol: '.' Code: 1001
Prior Symbol: 'J' Symbol: 'a' Code: 111
Prior Symbol: 'J' Symbol: 'e' Code: 1101
Prior Symbol: 'J' Symbol: 'i' Code: 1100
Prior Symbol: 'J' Symbol: 'o' Code: 0
Prior Symbol: 'J' Symbol: 'u' Code: 101
Prior Symbol: 'K' Symbol: 27 Code: 111
Prior Symbol: 'K' Symbol: 'a' Code: 100
Prior Symbol: 'K' Symbol: 'e' Code: 0
Prior Symbol: 'K' Symbol: 'i' Code: 101
Prior Symbol: 'K' Symbol: 'r' Code: 110
Prior Symbol: 'L' Symbol: 27 Code: 0110
Prior Symbol: 'L' Symbol: 'a' Code: 11
Prior Symbol: 'L' Symbol: 'e' Code: 00
Prior Symbol: 'L' Symbol: 'i' Code: 0111
Prior Symbol: 'L' Symbol: 'o' Code: 10
Prior Symbol: 'L' Symbol: 'u' Code: 010
Prior Symbol: 'M' Symbol: 27 Code: 11010
Prior Symbol: 'M' Symbol: 'a' Code: 0
Prior Symbol: 'M' Symbol: 'c' Code: 11011
Prior Symbol: 'M' Symbol: 'e' Code: 1111
Prior Symbol: 'M' Symbol: 'i' Code: 10
Prior Symbol: 'M' Symbol: 'o' Code: 1100
Prior Symbol: 'M' Symbol: 'u' Code: 1110
Prior Symbol: 'N' Symbol: 27 Code: 1100
Prior Symbol: 'N' Symbol: 'a' Code: 111
Prior Symbol: 'N' Symbol: 'e' Code: 0
Prior Symbol: 'N' Symbol: 'i' Code: 1101
Prior Symbol: 'N' Symbol: 'o' Code: 10
Prior Symbol: 'O' Symbol: 27 Code: 10
Prior Symbol: 'O' Symbol: "'" Code: 010
Prior Symbol: 'O' Symbol: 'l' Code: 110
Prior Symbol: 'O' Symbol: 'n' Code: 011
Prior Symbol: 'O' Symbol: 'r' Code: 111
Prior Symbol: 'O' Symbol: 's' Code: 00
Prior Symbol: 'P' Symbol: 27 Code: 10010
Prior Symbol: 'P' Symbol: 'a' Code: 0
Prior Symbol: 'P' Symbol: 'e' Code: 111
Prior Symbol: 'P' Symbol: 'h' Code: 10011
Prior Symbol: 'P' Symbol: 'i' Code: 1000
Prior Symbol: 'P' Symbol: 'l' Code: 1101
Prior Symbol: 'P' Symbol: 'o' Code: 101
Prior Symbol: 'P' Symbol: 'r' Code: 1100
Prior Symbol: 'Q' Symbol: 27 Code: 1
Prior Symbol: 'R' Symbol: 27 Code: 0000
Prior Symbol: 'R' Symbol: '.' Code: 0001
Prior Symbol: 'R' Symbol: 'a' Code: 01
Prior Symbol: 'R' Symbol: 'e' Code: 10
Prior Symbol: 'R' Symbol: 'i' Code: 001
Prior Symbol: 'R' Symbol: 'o' Code: 11
Prior Symbol: 'S' Symbol: 27 Code: 1011
Prior Symbol: 'S' Symbol: '.' Code: 0001
Prior Symbol: 'S' Symbol: 'a' Code: 100
Prior Symbol: 'S' Symbol: 'c' Code: 0010
Prior Symbol: 'S' Symbol: 'e' Code: 1110
Prior Symbol: 'S' Symbol: 'h' Code: 110
Prior Symbol: 'S' Symbol: 'i' Code: 0011
Prior Symbol: 'S' Symbol: 'o' Code: 1111
Prior Symbol: 'S' Symbol: 't' Code: 01
Prior Symbol: 'S' Symbol: 'u' Code: 1010
Prior Symbol: 'S' Symbol: 'v' Code: 00000
Prior Symbol: 'S' Symbol: 'y' Code: 00001
Prior Symbol: 'T' Symbol: 27 Code: 1010
Prior Symbol: 'T' Symbol: 'V' Code: 1000
Prior Symbol: 'T' Symbol: 'a' Code: 1001

Prior Symbol: 'T' Symbol: 'e' Code: 11010
Prior Symbol: 'T' Symbol: 'h' Code: 0
Prior Symbol: 'T' Symbol: 'i' Code: 1011
Prior Symbol: 'T' Symbol: 'o' Code: 111
Prior Symbol: 'T' Symbol: 'r' Code: 1100
Prior Symbol: 'T' Symbol: 'w' Code: 11011
Prior Symbol: 'U' Symbol: 27 Code: 10
Prior Symbol: 'U' Symbol: '.' Code: 0
Prior Symbol: 'U' Symbol: 'n' Code: 11
Prior Symbol: 'V' Symbol: 27 Code: 111
Prior Symbol: 'V' Symbol: ' ' Code: 10
Prior Symbol: 'V' Symbol: 'e' Code: 110
Prior Symbol: 'V' Symbol: 'i' Code: 0
Prior Symbol: 'W' Symbol: 27 Code: 010
Prior Symbol: 'W' Symbol: 'a' Code: 111
Prior Symbol: 'W' Symbol: 'e' Code: 110
Prior Symbol: 'W' Symbol: 'h' Code: 011
Prior Symbol: 'W' Symbol: 'i' Code: 10
Prior Symbol: 'W' Symbol: 'o' Code: 00
Prior Symbol: 'X' Symbol: 27 Code: 1
Prior Symbol: 'Y' Symbol: 27 Code: 0
Prior Symbol: 'Y' Symbol: 'o' Code: 1
Prior Symbol: 'Z' Symbol: 27 Code: 1
Prior Symbol: '[' Symbol: 27 Code: 1
Prior Symbol: '\' Symbol: 27 Code: 1
Prior Symbol: ']' Symbol: 27 Code: 1
Prior Symbol: '^' Symbol: 27 Code: 1
Prior Symbol: '_' Symbol: 27 Code: 1
Prior Symbol: '`' Symbol: 27 Code: 1
Prior Symbol: 'a' Symbol: 27 Code: 111001101
Prior Symbol: 'a' Symbol: ' ' Code: 101
Prior Symbol: 'a' Symbol: ''' Code: 111001110
Prior Symbol: 'a' Symbol: '.' Code: 1110010
Prior Symbol: 'a' Symbol: 'b' Code: 001011
Prior Symbol: 'a' Symbol: 'c' Code: 11001
Prior Symbol: 'a' Symbol: 'd' Code: 00111
Prior Symbol: 'a' Symbol: 'e' Code: 0011001
Prior Symbol: 'a' Symbol: 'f' Code: 001010
Prior Symbol: 'a' Symbol: 'g' Code: 00100
Prior Symbol: 'a' Symbol: 'h' Code: 001100010
Prior Symbol: 'a' Symbol: 'i' Code: 111000
Prior Symbol: 'a' Symbol: 'k' Code: 110000
Prior Symbol: 'a' Symbol: 'l' Code: 1101
Prior Symbol: 'a' Symbol: 'm' Code: 11101
Prior Symbol: 'a' Symbol: 'n' Code: 01
Prior Symbol: 'a' Symbol: 'o' Code: 001100011
Prior Symbol: 'a' Symbol: 'p' Code: 00000
Prior Symbol: 'a' Symbol: 'r' Code: 100
Prior Symbol: 'a' Symbol: 's' Code: 0001
Prior Symbol: 'a' Symbol: 't' Code: 1111
Prior Symbol: 'a' Symbol: 'u' Code: 110001
Prior Symbol: 'a' Symbol: 'v' Code: 001101
Prior Symbol: 'a' Symbol: 'w' Code: 111001111
Prior Symbol: 'a' Symbol: 'x' Code: 111001100
Prior Symbol: 'a' Symbol: 'y' Code: 00001
Prior Symbol: 'a' Symbol: 'z' Code: 00110000
Prior Symbol: 'b' Symbol: 27 Code: 101000
Prior Symbol: 'b' Symbol: ' ' Code: 0101
Prior Symbol: 'b' Symbol: '.' Code: 101001
Prior Symbol: 'b' Symbol: 'a' Code: 100
Prior Symbol: 'b' Symbol: 'b' Code: 101010
Prior Symbol: 'b' Symbol: 'd' Code: 1010110
Prior Symbol: 'b' Symbol: 'e' Code: 00
Prior Symbol: 'b' Symbol: 'i' Code: 1011
Prior Symbol: 'b' Symbol: 'l' Code: 0100
Prior Symbol: 'b' Symbol: 'o' Code: 110
Prior Symbol: 'b' Symbol: 'r' Code: 1110
Prior Symbol: 'b' Symbol: 's' Code: 1010111
Prior Symbol: 'b' Symbol: 'u' Code: 1111
Prior Symbol: 'b' Symbol: 'y' Code: 011
Prior Symbol: 'c' Symbol: 27 Code: 00010
Prior Symbol: 'c' Symbol: ' ' Code: 10000
Prior Symbol: 'c' Symbol: ',' Code: 010000
Prior Symbol: 'c' Symbol: '.' Code: 0100011
Prior Symbol: 'c' Symbol: 'D' Code: 0100110
Prior Symbol: 'c' Symbol: 'a' Code: 110
Prior Symbol: 'c' Symbol: 'c' Code: 010010
Prior Symbol: 'c' Symbol: 'e' Code: 011
Prior Symbol: 'c' Symbol: 'h' Code: 111
Prior Symbol: 'c' Symbol: 'i' Code: 0101
Prior Symbol: 'c' Symbol: 'k' Code: 1001
Prior Symbol: 'c' Symbol: 'l' Code: 10001
Prior Symbol: 'c' Symbol: 'o' Code: 101
Prior Symbol: 'c' Symbol: 'q' Code: 0100010
Prior Symbol: 'c' Symbol: 'r' Code: 00011
Prior Symbol: 'c' Symbol: 't' Code: 001
Prior Symbol: 'c' Symbol: 'u' Code: 0000
Prior Symbol: 'c' Symbol: 'y' Code: 0100111

Prior Symbol: 'd' Symbol: 27 Code: 1010001
Prior Symbol: 'd' Symbol: ' ' Code: 11
Prior Symbol: 'd' Symbol: '"' Code: 01111010
Prior Symbol: 'd' Symbol: ',' Code: 101011
Prior Symbol: 'd' Symbol: '.' Code: 0100
Prior Symbol: 'd' Symbol: ';' Code: 01111011
Prior Symbol: 'd' Symbol: 'a' Code: 1000
Prior Symbol: 'd' Symbol: 'd' Code: 01010
Prior Symbol: 'd' Symbol: 'e' Code: 00
Prior Symbol: 'd' Symbol: 'f' Code: 10100000
Prior Symbol: 'd' Symbol: 'g' Code: 10101011
Prior Symbol: 'd' Symbol: 'i' Code: 1011
Prior Symbol: 'd' Symbol: 'l' Code: 011111
Prior Symbol: 'd' Symbol: 'm' Code: 10100001
Prior Symbol: 'd' Symbol: 'n' Code: 1010100
Prior Symbol: 'd' Symbol: 'o' Code: 0110
Prior Symbol: 'd' Symbol: 'r' Code: 01110
Prior Symbol: 'd' Symbol: 's' Code: 1001
Prior Symbol: 'd' Symbol: 'u' Code: 101001
Prior Symbol: 'd' Symbol: 'v' Code: 0111100
Prior Symbol: 'd' Symbol: 'w' Code: 10101010
Prior Symbol: 'd' Symbol: 'y' Code: 01011
Prior Symbol: 'e' Symbol: 27 Code: 101110011
Prior Symbol: 'e' Symbol: ' ' Code: 111
Prior Symbol: 'e' Symbol: '"' Code: 10111010
Prior Symbol: 'e' Symbol: ')' Code: 100110000
Prior Symbol: 'e' Symbol: ',' Code: 000111
Prior Symbol: 'e' Symbol: '-' Code: 10011001
Prior Symbol: 'e' Symbol: '.' Code: 00110
Prior Symbol: 'e' Symbol: ';' Code: 10011010
Prior Symbol: 'e' Symbol: 'a' Code: 1000
Prior Symbol: 'e' Symbol: 'b' Code: 0001100
Prior Symbol: 'e' Symbol: 'c' Code: 10010
Prior Symbol: 'e' Symbol: 'd' Code: 0000
Prior Symbol: 'e' Symbol: 'e' Code: 10100
Prior Symbol: 'e' Symbol: 'f' Code: 10111011
Prior Symbol: 'e' Symbol: 'g' Code: 0001101
Prior Symbol: 'e' Symbol: 'h' Code: 100110001
Prior Symbol: 'e' Symbol: 'i' Code: 000100
Prior Symbol: 'e' Symbol: 'k' Code: 10011011
Prior Symbol: 'e' Symbol: 'l' Code: 0010
Prior Symbol: 'e' Symbol: 'm' Code: 100111
Prior Symbol: 'e' Symbol: 'n' Code: 010
Prior Symbol: 'e' Symbol: 'o' Code: 001110
Prior Symbol: 'e' Symbol: 'p' Code: 001111
Prior Symbol: 'e' Symbol: 'r' Code: 110
Prior Symbol: 'e' Symbol: 's' Code: 011
Prior Symbol: 'e' Symbol: 't' Code: 10101
Prior Symbol: 'e' Symbol: 'u' Code: 101110010
Prior Symbol: 'e' Symbol: 'v' Code: 101100
Prior Symbol: 'e' Symbol: 'w' Code: 101111
Prior Symbol: 'e' Symbol: 'x' Code: 000101
Prior Symbol: 'e' Symbol: 'y' Code: 101101
Prior Symbol: 'e' Symbol: 'z' Code: 10111000
Prior Symbol: 'f' Symbol: 27 Code: 1110111
Prior Symbol: 'f' Symbol: ' ' Code: 10
Prior Symbol: 'f' Symbol: '.' Code: 1110110
Prior Symbol: 'f' Symbol: 'a' Code: 1111
Prior Symbol: 'f' Symbol: 'e' Code: 000
Prior Symbol: 'f' Symbol: 'f' Code: 0101
Prior Symbol: 'f' Symbol: 'i' Code: 001
Prior Symbol: 'f' Symbol: 'l' Code: 111010
Prior Symbol: 'f' Symbol: 'o' Code: 110
Prior Symbol: 'f' Symbol: 'r' Code: 011
Prior Symbol: 'f' Symbol: 't' Code: 0100
Prior Symbol: 'f' Symbol: 'u' Code: 11100
Prior Symbol: 'g' Symbol: 27 Code: 1111010
Prior Symbol: 'g' Symbol: ' ' Code: 10
Prior Symbol: 'g' Symbol: '"' Code: 1111011
Prior Symbol: 'g' Symbol: ',' Code: 111110
Prior Symbol: 'g' Symbol: '-' Code: 0101010
Prior Symbol: 'g' Symbol: '.' Code: 01011
Prior Symbol: 'g' Symbol: 'a' Code: 1110
Prior Symbol: 'g' Symbol: 'e' Code: 00
Prior Symbol: 'g' Symbol: 'g' Code: 0101011
Prior Symbol: 'g' Symbol: 'h' Code: 011
Prior Symbol: 'g' Symbol: 'i' Code: 1101
Prior Symbol: 'g' Symbol: 'l' Code: 111100
Prior Symbol: 'g' Symbol: 'o' Code: 0100
Prior Symbol: 'g' Symbol: 'r' Code: 111111
Prior Symbol: 'g' Symbol: 's' Code: 11000
Prior Symbol: 'g' Symbol: 'u' Code: 11001
Prior Symbol: 'g' Symbol: 'y' Code: 010100
Prior Symbol: 'h' Symbol: 27 Code: 1011100
Prior Symbol: 'h' Symbol: ' ' Code: 100
Prior Symbol: 'h' Symbol: '"' Code: 10101000
Prior Symbol: 'h' Symbol: ',' Code: 10101001
Prior Symbol: 'h' Symbol: '-' Code: 10101011

Prior Symbol: 'h' Symbol: '.' Code: 101001
Prior Symbol: 'h' Symbol: 'a' Code: 011
Prior Symbol: 'h' Symbol: 'e' Code: 11
Prior Symbol: 'h' Symbol: 'i' Code: 00
Prior Symbol: 'h' Symbol: 'n' Code: 101011
Prior Symbol: 'h' Symbol: 'o' Code: 010
Prior Symbol: 'h' Symbol: 'r' Code: 101111
Prior Symbol: 'h' Symbol: 's' Code: 10101010
Prior Symbol: 'h' Symbol: 't' Code: 10110
Prior Symbol: 'h' Symbol: 'u' Code: 101000
Prior Symbol: 'h' Symbol: 'y' Code: 1011101
Prior Symbol: 'i' Symbol: 27 Code: 00011101
Prior Symbol: 'i' Symbol: ' ' Code: 0001111
Prior Symbol: 'i' Symbol: ',' Code: 100110100
Prior Symbol: 'i' Symbol: '.' Code: 10011000
Prior Symbol: 'i' Symbol: 'a' Code: 11010
Prior Symbol: 'i' Symbol: 'b' Code: 100110101
Prior Symbol: 'i' Symbol: 'c' Code: 1111
Prior Symbol: 'i' Symbol: 'd' Code: 10000
Prior Symbol: 'i' Symbol: 'e' Code: 1110
Prior Symbol: 'i' Symbol: 'f' Code: 100111
Prior Symbol: 'i' Symbol: 'g' Code: 10010
Prior Symbol: 'i' Symbol: 'k' Code: 10011011
Prior Symbol: 'i' Symbol: 'l' Code: 1100
Prior Symbol: 'i' Symbol: 'm' Code: 10001
Prior Symbol: 'i' Symbol: 'n' Code: 01
Prior Symbol: 'i' Symbol: 'o' Code: 11011
Prior Symbol: 'i' Symbol: 'p' Code: 000110
Prior Symbol: 'i' Symbol: 'r' Code: 0000
Prior Symbol: 'i' Symbol: 's' Code: 101
Prior Symbol: 'i' Symbol: 't' Code: 001
Prior Symbol: 'i' Symbol: 'v' Code: 00010
Prior Symbol: 'i' Symbol: 'x' Code: 00011100
Prior Symbol: 'i' Symbol: 'z' Code: 10011001
Prior Symbol: 'j' Symbol: 27 Code: 000
Prior Symbol: 'j' Symbol: 'a' Code: 001
Prior Symbol: 'j' Symbol: 'e' Code: 010
Prior Symbol: 'j' Symbol: 'o' Code: 1
Prior Symbol: 'j' Symbol: 'u' Code: 011
Prior Symbol: 'k' Symbol: 27 Code: 0000
Prior Symbol: 'k' Symbol: ' ' Code: 01
Prior Symbol: 'k' Symbol: '"' Code: 10000
Prior Symbol: 'k' Symbol: ',' Code: 10011
Prior Symbol: 'k' Symbol: '.' Code: 0001
Prior Symbol: 'k' Symbol: 'e' Code: 11
Prior Symbol: 'k' Symbol: 'i' Code: 101
Prior Symbol: 'k' Symbol: 'l' Code: 100100
Prior Symbol: 'k' Symbol: 'n' Code: 10001
Prior Symbol: 'k' Symbol: 's' Code: 001
Prior Symbol: 'k' Symbol: 'y' Code: 100101
Prior Symbol: 'l' Symbol: 27 Code: 0011100
Prior Symbol: 'l' Symbol: ' ' Code: 110
Prior Symbol: 'l' Symbol: '"' Code: 00111100
Prior Symbol: 'l' Symbol: ',' Code: 001101
Prior Symbol: 'l' Symbol: '-' Code: 00111101
Prior Symbol: 'l' Symbol: '.' Code: 00100
Prior Symbol: 'l' Symbol: 'a' Code: 000
Prior Symbol: 'l' Symbol: 'b' Code: 0011101
Prior Symbol: 'l' Symbol: 'c' Code: 00111111
Prior Symbol: 'l' Symbol: 'd' Code: 10111
Prior Symbol: 'l' Symbol: 'e' Code: 111
Prior Symbol: 'l' Symbol: 'f' Code: 010110
Prior Symbol: 'l' Symbol: 'i' Code: 011
Prior Symbol: 'l' Symbol: 'k' Code: 10110110
Prior Symbol: 'l' Symbol: 'l' Code: 100
Prior Symbol: 'l' Symbol: 'm' Code: 010111
Prior Symbol: 'l' Symbol: 'n' Code: 00111110
Prior Symbol: 'l' Symbol: 'o' Code: 1010
Prior Symbol: 'l' Symbol: 'p' Code: 00101
Prior Symbol: 'l' Symbol: 'r' Code: 10110111
Prior Symbol: 'l' Symbol: 's' Code: 01010
Prior Symbol: 'l' Symbol: 't' Code: 001100
Prior Symbol: 'l' Symbol: 'u' Code: 1011010
Prior Symbol: 'l' Symbol: 'v' Code: 101100
Prior Symbol: 'l' Symbol: 'y' Code: 0100
Prior Symbol: 'm' Symbol: 27 Code: 101010
Prior Symbol: 'm' Symbol: ' ' Code: 111
Prior Symbol: 'm' Symbol: '"' Code: 1010110
Prior Symbol: 'm' Symbol: '.' Code: 110101
Prior Symbol: 'm' Symbol: ';' Code: 1010111
Prior Symbol: 'm' Symbol: 'a' Code: 00
Prior Symbol: 'm' Symbol: 'b' Code: 10100
Prior Symbol: 'm' Symbol: 'e' Code: 01
Prior Symbol: 'm' Symbol: 'i' Code: 1100
Prior Symbol: 'm' Symbol: 'm' Code: 10110
Prior Symbol: 'm' Symbol: 'o' Code: 1000
Prior Symbol: 'm' Symbol: 'p' Code: 1001
Prior Symbol: 'm' Symbol: 's' Code: 10111

Prior Symbol: 'm' Symbol: 'u' Code: 11011
Prior Symbol: 'm' Symbol: 'y' Code: 110100
Prior Symbol: 'n' Symbol: 27 Code: 0100000
Prior Symbol: 'n' Symbol: ' ' Code: 10
Prior Symbol: 'n' Symbol: "'" Code: 0100011
Prior Symbol: 'n' Symbol: ',' Code: 111100
Prior Symbol: 'n' Symbol: '-' Code: 011011010
Prior Symbol: 'n' Symbol: '.' Code: 01100
Prior Symbol: 'n' Symbol: ';' Code: 011011011
Prior Symbol: 'n' Symbol: 'a' Code: 11111
Prior Symbol: 'n' Symbol: 'b' Code: 011011100
Prior Symbol: 'n' Symbol: 'c' Code: 01001
Prior Symbol: 'n' Symbol: 'd' Code: 110
Prior Symbol: 'n' Symbol: 'e' Code: 001
Prior Symbol: 'n' Symbol: 'f' Code: 01000101
Prior Symbol: 'n' Symbol: 'g' Code: 000
Prior Symbol: 'n' Symbol: 'i' Code: 01111
Prior Symbol: 'n' Symbol: 'j' Code: 011011101
Prior Symbol: 'n' Symbol: 'k' Code: 1111010
Prior Symbol: 'n' Symbol: 'l' Code: 01101100
Prior Symbol: 'n' Symbol: 'm' Code: 011011110
Prior Symbol: 'n' Symbol: 'n' Code: 01110
Prior Symbol: 'n' Symbol: 'o' Code: 1111011
Prior Symbol: 'n' Symbol: 'r' Code: 011011111
Prior Symbol: 'n' Symbol: 's' Code: 0101
Prior Symbol: 'n' Symbol: 't' Code: 1110
Prior Symbol: 'n' Symbol: 'u' Code: 0100001
Prior Symbol: 'n' Symbol: 'v' Code: 0110100
Prior Symbol: 'n' Symbol: 'y' Code: 0110101
Prior Symbol: 'n' Symbol: 'z' Code: 01000100
Prior Symbol: 'o' Symbol: 27 Code: 101010011
Prior Symbol: 'o' Symbol: ' ' Code: 001
Prior Symbol: 'o' Symbol: ',' Code: 01001111
Prior Symbol: 'o' Symbol: '-' Code: 01001110
Prior Symbol: 'o' Symbol: '.' Code: 0100110
Prior Symbol: 'o' Symbol: 'B' Code: 101010010
Prior Symbol: 'o' Symbol: 'a' Code: 100001
Prior Symbol: 'o' Symbol: 'b' Code: 110111
Prior Symbol: 'o' Symbol: 'c' Code: 100000
Prior Symbol: 'o' Symbol: 'd' Code: 110101
Prior Symbol: 'o' Symbol: 'e' Code: 1010101
Prior Symbol: 'o' Symbol: 'f' Code: 000
Prior Symbol: 'o' Symbol: 'g' Code: 1101000
Prior Symbol: 'o' Symbol: 'h' Code: 1101001
Prior Symbol: 'o' Symbol: 'i' Code: 1101101
Prior Symbol: 'o' Symbol: 'k' Code: 010010
Prior Symbol: 'o' Symbol: 'l' Code: 0101
Prior Symbol: 'o' Symbol: 'm' Code: 1100
Prior Symbol: 'o' Symbol: 'n' Code: 111
Prior Symbol: 'o' Symbol: 'o' Code: 10100
Prior Symbol: 'o' Symbol: 'p' Code: 01000
Prior Symbol: 'o' Symbol: 'r' Code: 011
Prior Symbol: 'o' Symbol: 's' Code: 10001
Prior Symbol: 'o' Symbol: 't' Code: 10010
Prior Symbol: 'o' Symbol: 'u' Code: 1011
Prior Symbol: 'o' Symbol: 'v' Code: 101011
Prior Symbol: 'o' Symbol: 'w' Code: 10011
Prior Symbol: 'o' Symbol: 'x' Code: 10101000
Prior Symbol: 'o' Symbol: 'y' Code: 1101100
Prior Symbol: 'p' Symbol: 27 Code: 011011
Prior Symbol: 'p' Symbol: ' ' Code: 000
Prior Symbol: 'p' Symbol: '-' Code: 1010010
Prior Symbol: 'p' Symbol: '.' Code: 101000
Prior Symbol: 'p' Symbol: 'a' Code: 001
Prior Symbol: 'p' Symbol: 'e' Code: 110
Prior Symbol: 'p' Symbol: 'h' Code: 1111
Prior Symbol: 'p' Symbol: 'i' Code: 1011
Prior Symbol: 'p' Symbol: 'l' Code: 010
Prior Symbol: 'p' Symbol: 'm' Code: 1010011
Prior Symbol: 'p' Symbol: 'o' Code: 0111
Prior Symbol: 'p' Symbol: 'p' Code: 11101
Prior Symbol: 'p' Symbol: 'r' Code: 100
Prior Symbol: 'p' Symbol: 's' Code: 01100
Prior Symbol: 'p' Symbol: 't' Code: 11100
Prior Symbol: 'p' Symbol: 'u' Code: 10101
Prior Symbol: 'p' Symbol: 'y' Code: 011010
Prior Symbol: 'q' Symbol: 27 Code: 0
Prior Symbol: 'q' Symbol: 'u' Code: 1
Prior Symbol: 'r' Symbol: 27 Code: 10011111
Prior Symbol: 'r' Symbol: ' ' Code: 111
Prior Symbol: 'r' Symbol: "'" Code: 1001110
Prior Symbol: 'r' Symbol: ')' Code: 100111100
Prior Symbol: 'r' Symbol: ',' Code: 100100
Prior Symbol: 'r' Symbol: '-' Code: 11001100
Prior Symbol: 'r' Symbol: '.' Code: 10001
Prior Symbol: 'r' Symbol: ';' Code: 100111101
Prior Symbol: 'r' Symbol: 'a' Code: 1101
Prior Symbol: 'r' Symbol: 'b' Code: 11001101

Prior Symbol: 'r' Symbol: 'c' Code: 100001
Prior Symbol: 'r' Symbol: 'd' Code: 11000
Prior Symbol: 'r' Symbol: 'e' Code: 101
Prior Symbol: 'r' Symbol: 'f' Code: 110011111
Prior Symbol: 'r' Symbol: 'g' Code: 100101
Prior Symbol: 'r' Symbol: 'i' Code: 010
Prior Symbol: 'r' Symbol: 'k' Code: 110010
Prior Symbol: 'r' Symbol: 'l' Code: 00100
Prior Symbol: 'r' Symbol: 'm' Code: 00101
Prior Symbol: 'r' Symbol: 'n' Code: 01100
Prior Symbol: 'r' Symbol: 'o' Code: 000
Prior Symbol: 'r' Symbol: 'p' Code: 11001110
Prior Symbol: 'r' Symbol: 'r' Code: 100110
Prior Symbol: 'r' Symbol: 's' Code: 0111
Prior Symbol: 'r' Symbol: 't' Code: 0011
Prior Symbol: 'r' Symbol: 'u' Code: 100000
Prior Symbol: 'r' Symbol: 'v' Code: 110011110
Prior Symbol: 'r' Symbol: 'y' Code: 01101
Prior Symbol: 's' Symbol: 27 Code: 10011100
Prior Symbol: 's' Symbol: ' ' Code: 0
Prior Symbol: 's' Symbol: '"' Code: 100111100
Prior Symbol: 's' Symbol: ''' Code: 100111101
Prior Symbol: 's' Symbol: ',' Code: 111011
Prior Symbol: 's' Symbol: '.' Code: 1000
Prior Symbol: 's' Symbol: ';' Code: 11101011
Prior Symbol: 's' Symbol: 'a' Code: 110011
Prior Symbol: 's' Symbol: 'b' Code: 100111110
Prior Symbol: 's' Symbol: 'c' Code: 10010
Prior Symbol: 's' Symbol: 'e' Code: 1101
Prior Symbol: 's' Symbol: 'h' Code: 11000
Prior Symbol: 's' Symbol: 'i' Code: 11100
Prior Symbol: 's' Symbol: 'k' Code: 100111111
Prior Symbol: 's' Symbol: 'l' Code: 1110100
Prior Symbol: 's' Symbol: 'm' Code: 111010100
Prior Symbol: 's' Symbol: 'n' Code: 111010101
Prior Symbol: 's' Symbol: 'o' Code: 11110
Prior Symbol: 's' Symbol: 'p' Code: 1001101
Prior Symbol: 's' Symbol: 's' Code: 11111
Prior Symbol: 's' Symbol: 't' Code: 101
Prior Symbol: 's' Symbol: 'u' Code: 110010
Prior Symbol: 's' Symbol: 'w' Code: 10011101
Prior Symbol: 's' Symbol: 'y' Code: 1001100
Prior Symbol: 't' Symbol: 27 Code: 11000011
Prior Symbol: 't' Symbol: ' ' Code: 111
Prior Symbol: 't' Symbol: '"' Code: 11000100
Prior Symbol: 't' Symbol: ',' Code: 0111100
Prior Symbol: 't' Symbol: '-' Code: 01111110
Prior Symbol: 't' Symbol: '.' Code: 01101
Prior Symbol: 't' Symbol: ';' Code: 110000100
Prior Symbol: 't' Symbol: 'a' Code: 0100
Prior Symbol: 't' Symbol: 'b' Code: 110000101
Prior Symbol: 't' Symbol: 'c' Code: 11000101
Prior Symbol: 't' Symbol: 'e' Code: 101
Prior Symbol: 't' Symbol: 'h' Code: 00
Prior Symbol: 't' Symbol: 'i' Code: 1101
Prior Symbol: 't' Symbol: 'l' Code: 0111101
Prior Symbol: 't' Symbol: 'm' Code: 01111111
Prior Symbol: 't' Symbol: 'n' Code: 0111110
Prior Symbol: 't' Symbol: 'o' Code: 100
Prior Symbol: 't' Symbol: 'r' Code: 11001
Prior Symbol: 't' Symbol: 's' Code: 0101
Prior Symbol: 't' Symbol: 't' Code: 01100
Prior Symbol: 't' Symbol: 'u' Code: 01110
Prior Symbol: 't' Symbol: 'w' Code: 1100000
Prior Symbol: 't' Symbol: 'y' Code: 1100011
Prior Symbol: 'u' Symbol: 27 Code: 1001100
Prior Symbol: 'u' Symbol: ' ' Code: 100000
Prior Symbol: 'u' Symbol: 'a' Code: 100111
Prior Symbol: 'u' Symbol: 'b' Code: 100001
Prior Symbol: 'u' Symbol: 'c' Code: 10001
Prior Symbol: 'u' Symbol: 'd' Code: 11100
Prior Symbol: 'u' Symbol: 'e' Code: 11101
Prior Symbol: 'u' Symbol: 'g' Code: 11110
Prior Symbol: 'u' Symbol: 'i' Code: 10010
Prior Symbol: 'u' Symbol: 'k' Code: 1001101
Prior Symbol: 'u' Symbol: 'l' Code: 0100
Prior Symbol: 'u' Symbol: 'm' Code: 111111
Prior Symbol: 'u' Symbol: 'n' Code: 110
Prior Symbol: 'u' Symbol: 'o' Code: 11111010
Prior Symbol: 'u' Symbol: 'p' Code: 0101
Prior Symbol: 'u' Symbol: 'r' Code: 00
Prior Symbol: 'u' Symbol: 's' Code: 011
Prior Symbol: 'u' Symbol: 't' Code: 101
Prior Symbol: 'u' Symbol: 'v' Code: 11111011
Prior Symbol: 'u' Symbol: 'y' Code: 1111100
Prior Symbol: 'v' Symbol: 27 Code: 00010
Prior Symbol: 'v' Symbol: 'a' Code: 001
Prior Symbol: 'v' Symbol: 'e' Code: 1

Prior Symbol: 'v' Symbol: 'i' Code: 01
Prior Symbol: 'v' Symbol: 'o' Code: 0000
Prior Symbol: 'v' Symbol: 's' Code: 000110
Prior Symbol: 'v' Symbol: 'y' Code: 000111
Prior Symbol: 'w' Symbol: 27 Code: 011101
Prior Symbol: 'w' Symbol: ' ' Code: 001
Prior Symbol: 'w' Symbol: '.' Code: 011100
Prior Symbol: 'w' Symbol: 'a' Code: 010
Prior Symbol: 'w' Symbol: 'e' Code: 1110
Prior Symbol: 'w' Symbol: 'h' Code: 000
Prior Symbol: 'w' Symbol: 'i' Code: 10
Prior Symbol: 'w' Symbol: 'l' Code: 011110
Prior Symbol: 'w' Symbol: 'm' Code: 011111
Prior Symbol: 'w' Symbol: 'n' Code: 11111
Prior Symbol: 'w' Symbol: 'o' Code: 110
Prior Symbol: 'w' Symbol: 'r' Code: 0110
Prior Symbol: 'w' Symbol: 's' Code: 11110
Prior Symbol: 'x' Symbol: 27 Code: 10
Prior Symbol: 'x' Symbol: ' ' Code: 0110
Prior Symbol: 'x' Symbol: ',' Code: 0111
Prior Symbol: 'x' Symbol: '-' Code: 1100
Prior Symbol: 'x' Symbol: 'a' Code: 111
Prior Symbol: 'x' Symbol: 'e' Code: 00
Prior Symbol: 'x' Symbol: 'i' Code: 010
Prior Symbol: 'x' Symbol: 't' Code: 1101
Prior Symbol: 'y' Symbol: 27 Code: 01010
Prior Symbol: 'y' Symbol: ' ' Code: 1
Prior Symbol: 'y' Symbol: '"' Code: 010010
Prior Symbol: 'y' Symbol: ',' Code: 0001
Prior Symbol: 'y' Symbol: '.' Code: 0111
Prior Symbol: 'y' Symbol: ';' Code: 011001
Prior Symbol: 'y' Symbol: '?' Code: 0100110
Prior Symbol: 'y' Symbol: 'a' Code: 0100111
Prior Symbol: 'y' Symbol: 'b' Code: 0110000
Prior Symbol: 'y' Symbol: 'd' Code: 000001
Prior Symbol: 'y' Symbol: 'e' Code: 0010
Prior Symbol: 'y' Symbol: 'f' Code: 0110001
Prior Symbol: 'y' Symbol: 'i' Code: 000010
Prior Symbol: 'y' Symbol: 'l' Code: 01000
Prior Symbol: 'y' Symbol: 'm' Code: 000000
Prior Symbol: 'y' Symbol: 'n' Code: 01011
Prior Symbol: 'y' Symbol: 'o' Code: 01101
Prior Symbol: 'y' Symbol: 's' Code: 0011
Prior Symbol: 'y' Symbol: 'w' Code: 000011
Prior Symbol: 'z' Symbol: 27 Code: 100
Prior Symbol: 'z' Symbol: ' ' Code: 1110
Prior Symbol: 'z' Symbol: '.' Code: 1111
Prior Symbol: 'z' Symbol: 'a' Code: 000
Prior Symbol: 'z' Symbol: 'e' Code: 001
Prior Symbol: 'z' Symbol: 'i' Code: 110
Prior Symbol: 'z' Symbol: 'l' Code: 010
Prior Symbol: 'z' Symbol: 'o' Code: 101
Prior Symbol: 'z' Symbol: 'z' Code: 011
Prior Symbol: '{' Symbol: 27 Code: 1
Prior Symbol: '|' Symbol: 27 Code: 1
Prior Symbol: '}' Symbol: 27 Code: 1
Prior Symbol: '~' Symbol: 27 Code: 1
Prior Symbol: 127 Symbol: 27 Code: 1

Table B.C.7/J.94 – English-language Program Description Decode Table

0 1	42 1	84 1	126 2	168 3
1 0	43 84	85 252	127 94	169 74
2 1	44 1	86 1	128 2	170 3
3 44	45 86	87 254	129 96	171 90
4 1	46 1	88 2	130 2	172 3
5 46	47 88	89 0	131 98	173 94
6 1	48 1	90 2	132 2	174 3
7 48	49 90	91 4	133 118	175 100
8 1	50 1	92 2	134 2	176 3
9 50	51 92	93 22	135 132	177 110
10 1	52 1	94 2	136 2	178 3
11 52	53 94	95 32	137 148	179 112
12 1	54 1	96 2	138 2	180 3
13 54	55 96	97 34	139 162	181 114
14 1	56 1	98 2	140 2	182 3
15 56	57 98	99 44	141 178	183 116
16 1	58 1	100 2	142 2	184 3
17 58	59 100	101 50	143 186	185 118
18 1	60 1	102 2	144 2	186 3
19 60	61 102	103 56	145 200	187 120
20 1	62 1	104 2	146 2	188 3
21 62	63 104	105 60	147 210	189 122
22 1	64 1	106 2	148 2	190 3
23 64	65 106	107 64	149 222	191 124
24 1	66 1	108 2	150 2	192 3
25 66	67 222	109 68	151 234	193 126
26 1	68 1	110 2	152 2	194 3
27 68	69 224	111 70	153 242	195 128
28 1	70 1	112 2	154 2	196 3
29 70	71 234	113 74	155 252	197 180
30 1	72 1	114 2	156 3	198 3
31 72	73 236	115 76	157 8	199 206
32 1	74 1	116 2	158 3	200 3
33 74	75 238	117 84	159 16	201 240
34 1	76 1	118 2	160 3	202 4
35 76	77 240	119 86	161 26	203 26
36 1	78 1	120 2	162 3	204 4
37 78	79 242	121 88	163 40	205 88
38 1	80 1	122 2	164 3	206 4
39 80	81 248	123 90	165 42	207 110
40 1	82 1	124 2	166 3	208 4
41 82	83 250	125 92	167 52	209 142

210 4
211 172
212 4
213 216
214 4
215 224
216 4
217 244
218 5
219 36
220 5
221 64
222 5
223 118
224 5
225 174
226 5
227 206
228 5
229 208
230 6
231 6
232 6
233 52
234 6
235 96
236 6
237 134
238 6
239 146
240 6
241 170
242 6
243 184
244 6
245 220
246 6
247 236 248 6
249 238
250 6
251 240
252 6
253 242
254 6
255 244
256 20
257 21
258 155
259 214
260 201
261 207
262 215
263 199
264 1
265 162
266 206
267 203
268 2
269 3
270 197
271 204
272 198
273 200
274 4
275 196
276 5
277 194
278 6
279 195
280 210
281 7
282 211
283 8
284 202
285 212
286 9
287 205
288 208
289 10
290 193
291 11
292 12
293 13
294 14
295 15
296 16
297 17
298 18
299 19
300 155
301 155
302 155
303 155
304 155
305 155
306 155
307 155
308 155
309 155
310 155
311 155
312 155
313 155
314 155
315 155
316 155
317 155
318 155
319 155
320 155
321 155
322 155
323 155
324 155
325 155
326 155
327 155
328 155
329 155
330 155
331 155
332 155
333 155
334 155
335 155
336 155
337 155
338 155
339 155
340 155
341 155
342 155
343 155
344 155
345 155
346 155
347 155
348 155
349 155
350 155
351 155
352 155
353 155
354 155
355 155
356 155
357 155
358 155
359 155
360 155
361 155
362 56
363 57
364 173
365 175
366 183
367 218
368 168
369 179
370 181
371 1
372 2
373 155
374 180
375 241
376 162
377 213
378 214
379 217
380 3
381 4
382 5
383 207
384 6
385 201
386 249
387 234
388 235
389 245
390 246
391 7
392 8
393 9
394 178
395 197
396 198
397 177
398 10
399 238
400 203
401 11
402 212
403 12
404 196
405 200
406 210
407 13
408 14
409 15
410 199
411 202
412 206
413 208
414 215
415 16
416 194
417 17
418 204
419 236
420 229
421 231
422 18
423 205
424 19
425 20
426 195
427 21
428 22
429 23
430 237
431 24
432 25
433 242
434 26
435 211
436 27
437 28
438 228
439 29
440 193

441 227
442 30
443 233
444 240
445 226
446 247
447 31
448 243
449 230
450 32
451 33
452 34
453 232
454 239
455 35
456 36
457 37
458 38
459 39
460 40
461 41
462 42
463 244
464 43
465 44
466 45
467 46
468 47
469 225
470 48
471 49
472 50
473 51
474 52
475 53
476 54
477 55
478 155
479 155
480 3
481 4
482 128
483 174
484 200
485 212
486 1
487 2
488 155
489 160
490 155
491 155
492 155
493 155
494 155
495 155
496 155
497 155
498 2
499 243
500 160
501 244
502 155
503 1
504 155
505 155
506 172
507 155
508 155
509 155
510 155
511 155
512 1
513 160
514 155
515 162
516 7
517 8
518 226
519 228
520 229
521 230
522 160
523 242
524 225
525 1
526 2
527 243
528 227
529 3
530 4
531 5
532 155
533 6
534 4
535 128
536 202
537 211
538 162
539 1
540 155
541 2
542 3
543 160
544 155
545 160
546 3
547 4
548 155
549 183
550 244
551 160
552 176
553 243
554 1
555 2
556 185
557 2
558 184
559 155
560 160
561 1
562 174
563 2
564 182
565 155
566 1
567 160
568 160
569 1
570 155
571 176
572 174
573 1
574 155
575 160
576 174
577 1
578 160
579 155
580 155
581 155
582 155
583 1
584 172
585 174
586 155
587 155
588 2
589 3
590 155
591 160
592 181
593 182
594 184
595 1
596 155
597 160
598 155
599 160
600 155
601 155
602 155
603 155
604 155
605 155
606 155
607 160
608 155
609 155
610 8
611 9
612 230
613 245
614 243
615 244
616 155
617 228
618 1
619 237
620 2
621 3
622 4
623 242
624 5
625 6
626 236
627 238
628 7
629 160
630 5
631 6
632 155
633 236
634 245
635 1
636 2
637 225
638 239
639 229
640 233
641 242
642 3
643 4
644 6
645 7
646 155
647 233
648 249
649 242
650 245
651 1
652 2
653 3
654 236
655 239
656 225
657 4
658 232
659 5
660 5
661 6
662 249
663 242
664 245
665 155
666 229
667 239
668 1
669 2
670 233

671 225
672 3
673 4
674 6
675 7
676 225
677 233
678 238
679 246
680 228
681 236
682 243
683 1
684 2
685 242
686 3
687 4
688 155
689 5
690 2
691 3
692 229
693 236
694 155
695 239
696 1
697 242
698 5
699 6
700 245
701 239
702 155
703 236
704 233
705 1
706 225
707 242
708 2
709 229
710 3
711 4
712 3
713 4
714 155
715 229
716 233
717 245
718 225
719 1
720 239
721 2
722 4
723 5
724 160
725 201
726 243
727 155
728 174
729 242
730 1
731 2
732 3
733 238
734 239
735 5
736 155
737 174
738 233
739 229
740 1
741 245
742 2
743 225
744 3
745 4
746 229
747 3
748 225
749 233
750 242
751 155
752 1
753 2
754 3
755 4
756 155
757 233
758 245
759 1
760 229
761 2
762 239
763 225
764 225
765 5
766 155
767 227
768 239
769 1
770 245
771 229
772 2
773 3
774 233
775 4
776 229
777 3
778 155
779 233
780 1
781 225
782 239
783 2
784 3
785 4
786 167
787 238
788 236
789 242
790 243
791 1
792 155
793 2
794 225
795 6
796 155
797 232
798 233
799 1
800 242
801 236
802 2
803 239
804 3
805 229
806 4
807 5
808 155
809 155
810 3
811 4
812 155
813 174
814 1
815 233
816 2
817 225
818 229
819 239
820 9
821 10
822 246
823 249
824 1
825 174
826 227
827 233
828 245
829 155
830 229
831 239
832 2
833 3
834 225
835 4
836 232
837 5
838 6
839 244
840 7
841 8
842 232
843 7
844 229
845 247
846 214
847 225
848 155
849 233
850 242
851 1
852 2
853 3
854 4
855 239
856 5
857 6
858 174
859 1
860 155
861 238
862 233
863 2
864 229
865 155
866 160
867 1
868 3
869 4
870 155
871 232
872 229
873 225
874 239
875 1
876 233
877 2
878 155
879 155
880 155
881 239
882 155
883 155
884 155
885 155
886 155
887 155
888 155
889 155
890 155
891 155
892 155
893 155
894 155
895 155
896 24
897 25
898 232
899 239
900 248

901 155
902 167
903 247
904 250
905 1
906 2
907 3
908 4
909 229
910 174
911 5
912 230
913 226
914 6
915 246
916 235
917 245
918 233
919 7
920 240
921 249
922 231
923 8
924 9
925 228
926 10
927 227
928 11
929 237
930 12
931 243
932 13
933 14
934 15
935 236
936 16
937 244
938 17
939 18
940 242
941 160
942 19
943 20
944 21
945 238
946 22
947 23
948 11
949 12
950 228
951 243
952 155
953 174
954 226
955 1
956 2
957 3
958 236
959 160
960 4
961 233
962 242
963 245
964 5
965 249
966 225
967 6
968 239
969 7
970 229
971 8
972 9
973 10
974 15
975 16
976 241
977 174
978 196
979 249
980 172
981 1
982 227
983 2
984 155
985 242
986 3
987 4
988 160
989 236
990 245
991 5
992 6
993 233
994 7
995 235
996 8
997 244
998 9
999 229
1000 10
1001 239
1002 225
1003 232
1004 11
1005 12
1006 13
1007 14
1008 19
1009 20
1010 167
1011 187
1012 230
1013 237
1014 247
1015 231
1016 246
1017 1
1018 2
1019 155
1020 238
1021 3
1022 4
1023 236
1024 5
1025 245
1026 6
1027 172
1028 228
1029 249
1030 242
1031 7
1032 8
1033 9
1034 174
1035 10
1036 239
1037 11
1038 225
1039 243
1040 12
1041 233
1042 13
1043 14
1044 15
1045 16
1046 229
1047 17
1048 18
1049 160
1050 29
1051 30
1052 169
1053 232
1054 245
1055 155
1056 1
1057 173
1058 187
1059 235
1060 250
1061 2
1062 167
1063 230
1064 226
1065 231
1066 3
1067 4
1068 5
1069 6
1070 233
1071 248
1072 7
1073 172
1074 239
1075 240
1076 8
1077 237
1078 246
1079 249
1080 9
1081 247
1082 10
1083 11
1084 174
1085 12
1086 227
1087 13
1088 229
1089 244
1090 14
1091 15
1092 228
1093 16
1094 236
1095 17
1096 225
1097 18
1098 19
1099 20
1100 21
1101 22
1102 238
1103 243
1104 23
1105 24
1106 242
1107 160
1108 25
1109 26
1110 27
1111 28
1112 9
1113 10
1114 174
1115 155
1116 236
1117 1
1118 245
1119 2
1120 244
1121 230
1122 3
1123 225
1124 229
1125 233
1126 4
1127 242
1128 239
1129 5
1130 6

1131 7	1177 174	1223 9	1269 23	1315 21
1132 160	1178 3	1224 10	1270 167	1316 12
1133 8	1179 238	1225 11	1271 173	1317 13
1134 14	1180 4	1226 236	1272 238	1318 167
1135 15	1181 242	1227 12	1273 227	1319 187
1136 173	1182 5	1228 229	1274 235	1320 155
1137 231	1183 6	1229 227	1275 242	1321 1
1138 155	1184 244	1230 13	1276 155	1322 249
1139 167	1185 7	1231 244	1277 226	1323 174
1140 249	1186 8	1232 14	1278 1	1324 226
1141 1	1187 9	1233 243	1279 2	1325 2
1142 236	1188 239	1234 15	1280 245	1326 237
1143 2	1189 225	1235 16	1281 3	1327 243
1144 172	1190 160	1236 17	1282 244	1328 3
1145 242	1191 10	1237 238	1283 172	1329 245
1146 3	1192 233	1238 18	1284 4	1330 239
1147 174	1193 11	1239 19	1285 5	1331 240
1148 243	1194 12	1240 3	1286 230	1332 4
1149 245	1195 229	1241 239	1287 237	1333 5
1150 4	1196 20	1242 155	1288 246	1334 233
1151 5	1197 21	1243 225	1289 6	1335 6
1152 239	1198 172	1244 229	1290 174	1336 7
1153 6	1199 226	1245 245	1291 240	1337 8
1154 7	1200 248	1246 1	1292 7	1338 9
1155 233	1201 155	1247 2	1293 8	1339 160
1156 225	1202 174	1248 8	1294 243	1340 225
1157 8	1203 250	1249 9	1295 9	1341 229
1158 9	1204 1	1250 236	1296 10	1342 10
1159 232	1205 235	1251 249	1297 228	1343 11
1160 10	1206 2	1252 167	1298 11	1344 25
1161 11	1207 160	1253 238	1299 12	1345 26
1162 229	1208 3	1254 1	1300 249	1346 173
1163 12	1209 4	1255 172	1301 13	1347 187
1164 160	1210 240	1256 155	1302 239	1348 226
1165 13	1211 5	1257 174	1303 14	1349 234
1166 13	1212 6	1258 2	1304 225	1350 237
1167 14	1213 230	1259 3	1305 15	1351 242
1168 167	1214 246	1260 4	1306 16	1352 250
1169 172	1215 7	1261 243	1307 233	1353 230
1170 243	1216 228	1262 5	1308 236	1354 236
1171 173	1217 237	1263 233	1309 17	1355 1
1172 1	1218 231	1264 6	1310 160	1356 2
1173 2	1219 8	1265 160	1311 229	1357 3
1174 155	1220 225	1266 7	1312 18	1358 155
1175 249	1221 239	1267 229	1313 19	1359 245
1176 245	1222 242	1268 22	1314 20	1360 4

1361 167
1362 246
1363 249
1364 5
1365 6
1366 235
1367 239
1368 7
1369 8
1370 9
1371 10
1372 172
1373 11
1374 12
1375 227
1376 174
1377 13
1378 238
1379 233
1380 14
1381 225
1382 15
1383 243
1384 16
1385 17
1386 244
1387 18
1388 231
1389 229
1390 19
1391 20
1392 228
1393 21
1394 22
1395 23
1396 160
1397 24
1398 26
1399 27
1400 194
1401 155
1402 173
1403 172
1404 248
1405 1
1406 174
1407 2
1408 3
1409 229
1410 231
1411 232
1412 249
1413 233
1414 235
1415 4
1416 227
1417 225
1418 5
1419 246
1420 6
1421 228
1422 7
1423 226
1424 240
1425 8
1426 9
1427 243
1428 244
1429 247
1430 239
1431 10
1432 11
1433 12
1434 13
1435 236
1436 14
1437 15
1438 16
1439 245
1440 237
1441 17
1442 230
1443 160
1444 18
1445 242
1446 19
1447 20
1448 21
1449 238
1450 22
1451 23
1452 24
1453 25
1454 14
1455 15
1456 173
1457 237
1458 249
1459 155
1460 174
1461 1
1462 243
1463 2
1464 3
1465 245
1466 244
1467 240
1468 4
1469 239
1470 5
1471 233
1472 6
1473 232
1474 160
1475 225
1476 236
1477 7
1478 242
1479 8
1480 229
1481 9
1482 10
1483 11
1484 12
1485 13
1486 155
1487 245
1488 25
1489 26
1490 169
1491 187
1492 246
1493 230
1494 1
1495 155
1496 173
1497 226
1498 240
1499 2
1500 167
1501 3
1502 4
1503 5
1504 245
1505 227
1506 172
1507 231
1508 242
1509 6
1510 235
1511 7
1512 236
1513 237
1514 238
1515 249
1516 8
1517 174
1518 9
1519 10
1520 228
1521 11
1522 12
1523 244
1524 13
1525 243
1526 14
1527 15
1528 16
1529 225
1530 239
1531 17
1532 233
1533 18
1534 19
1535 229
1536 20
1537 160
1538 21
1539 22
1540 23
1541 24
1542 160
1543 22
1544 162
1545 167
1546 226
1547 235
1548 237
1549 238
1550 155
1551 247
1552 1
1553 2
1554 3
1555 187
1556 249
1557 240
1558 4
1559 5
1560 236
1561 6
1562 7
1563 8
1564 245
1565 225
1566 9
1567 172
1568 227
1569 10
1570 232
1571 11
1572 233
1573 12
1574 239
1575 243
1576 174
1577 13
1578 14
1579 229
1580 15
1581 16
1582 17
1583 244
1584 18
1585 19
1586 20
1587 21
1588 20
1589 21
1590 187

1591 226
1592 173
1593 237
1594 1
1595 155
1596 167
1597 227
1598 172
1599 236
1600 238
1601 2
1602 247
1603 3
1604 4
1605 249
1606 5
1607 6
1608 7
1609 8
1610 244
1611 174
1612 245
1613 9
1614 10
1615 242
1616 225
1617 243
1618 11
1619 12
1620 13
1621 233
1622 14
1623 15
1624 239
1625 229
1626 16
1627 160
1628 232
1629 17
1630 18
1631 19
1632 17
1633 18
1634 239
1635 246
1636 155
1637 235
1638 249
1639 1
1640 160
1641 226
1642 2
1643 225
1644 3
1645 237
1646 4
1647 227
1648 233
1649 5
1650 228
1651 229
1652 231
1653 6
1654 236
1655 240
1656 7
1657 8
1658 9
1659 10
1660 11
1661 243
1662 12
1663 244
1664 238
1665 13
1666 242
1667 14
1668 15
1669 16
1670 5
1671 229
1672 243
1673 249
1674 155
1675 1
1676 239
1677 2
1678 3
1679 225
1680 4
1681 233
1682 10
1683 11
1684 174
1685 155
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

APPENDIX B.I

Implementation recommendations

B.I.1 Implications for retail digital cable-ready devices

Given that a cable operator could choose to deliver SI tables according to any of the profiles defined in Annex B.A on any given hub, digital cable-ready devices offered for retail sale should be able to accept a Short-form Virtual Channel Table for basic navigation if the Long-form Virtual Channel is not provided. It should also accept the Long-form Virtual Channel Table if the Short-form table is not provided.

B.I.2 Channel number handling

Host devices are expected to support navigation based on virtual channel records associated with two-part channel numbers. If an S-VCT virtual channel record includes a two_part_channel_number_descriptor(), the Host is expected to use it, and to disregard the 12-bit virtual_channel_number field in the same virtual_channel() record.

If a two_part_channel_number_descriptor() is not present in the record-level descriptors loop of a particular S-VCT virtual channel record, the Host is expected to use the virtual_channel_number field in the virtual_channel() record, (see Table B.20) as the channel number reference.

Both numbering schemes may co-exist in a channel map, but each individual channel must be considered labelled with either a one-part or a two-part number.

B.I.3 Processing of dynamic changes to service information

The Host is expected to monitor SI data on a continuous basis, and react to changes dynamically. For example, an update to an S-VCT or L-VCT may indicate that the definition of the currently acquired virtual channel has changed. The change could involve, for example, association of the channel with a different MPEG-2 program_number within a Transport Stream on a different carrier frequency. In response to such a change, the Host is expected to tune to and acquire the service as redefined.

For some types of changes, the Host is not expected to respond in a visible way. For example, the name of the current event may change, but the new name would be visible as the response to a regular user action to show the event name on-screen or in a program guide display.

B.I.4 AEITs may include event information for inaccessible channels

In the out-of-band system, depending on the data delivery methods employed by the cable headend and POD module, there may be occasions where AEITs are broadcast for which some set-top boxes do not have corresponding virtual channel assignments. In these cases, the Host is expected to discard portions of the AEITs corresponding to source_ID values not present in the Virtual Channel Table (short- or long-form).

For example, the AEIT may include data describing the program schedule for a service identified with source_ID value 0x0123. Let's say the Virtual Channel Table does not include a channel associated with source_ID 0x0123. When constructing a program guide display, the channel name, number and physical location associated with events tied to source_ID 0x0123 will not be available. Therefore, the events described in the AEIT data for this channel are inaccessible, and the AEIT records for this source_ID should be discarded.

B.I.5 Splice flag processing

The S-VCT includes a flag called splice. Hosts supporting application of virtual channel changes tied to video splice point timing are expected to execute the change after two seconds following the activation_time, in the absence of a video splice point prior to that time.

Support of the splice timing function is optional in Hosts. A Host not supporting the splice timing feature is expected to apply the data delivered in the VCM_structure() at the indicated activation time (i.e. the splice flag may be simply disregarded).

APPENDIX B.II

Service Information overview and guide

B.II.1 Table hierarchy

Figures B.II.1 through B.II.5 describe the relationships between SI tables for Profiles 1 through 6 in a simplified form. A mandatory table is shown in solid box. An optional table is shown in dotted box. An italicized name indicates a sub-table or a map carried within the table.

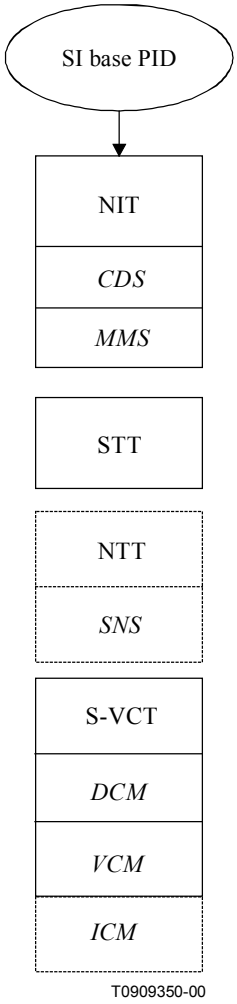

Figure B.II.1/J.94 – Hierarchy of Table Sections – Profiles 1 and 2

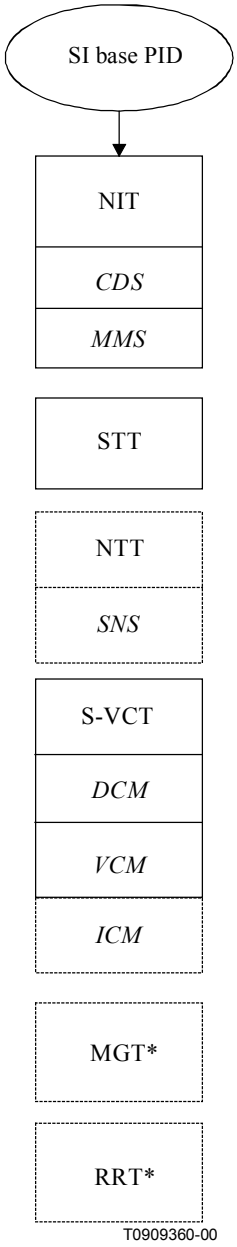

Figure B.II.2/J.94 – Hierarchy of Table Sections – Profile 3

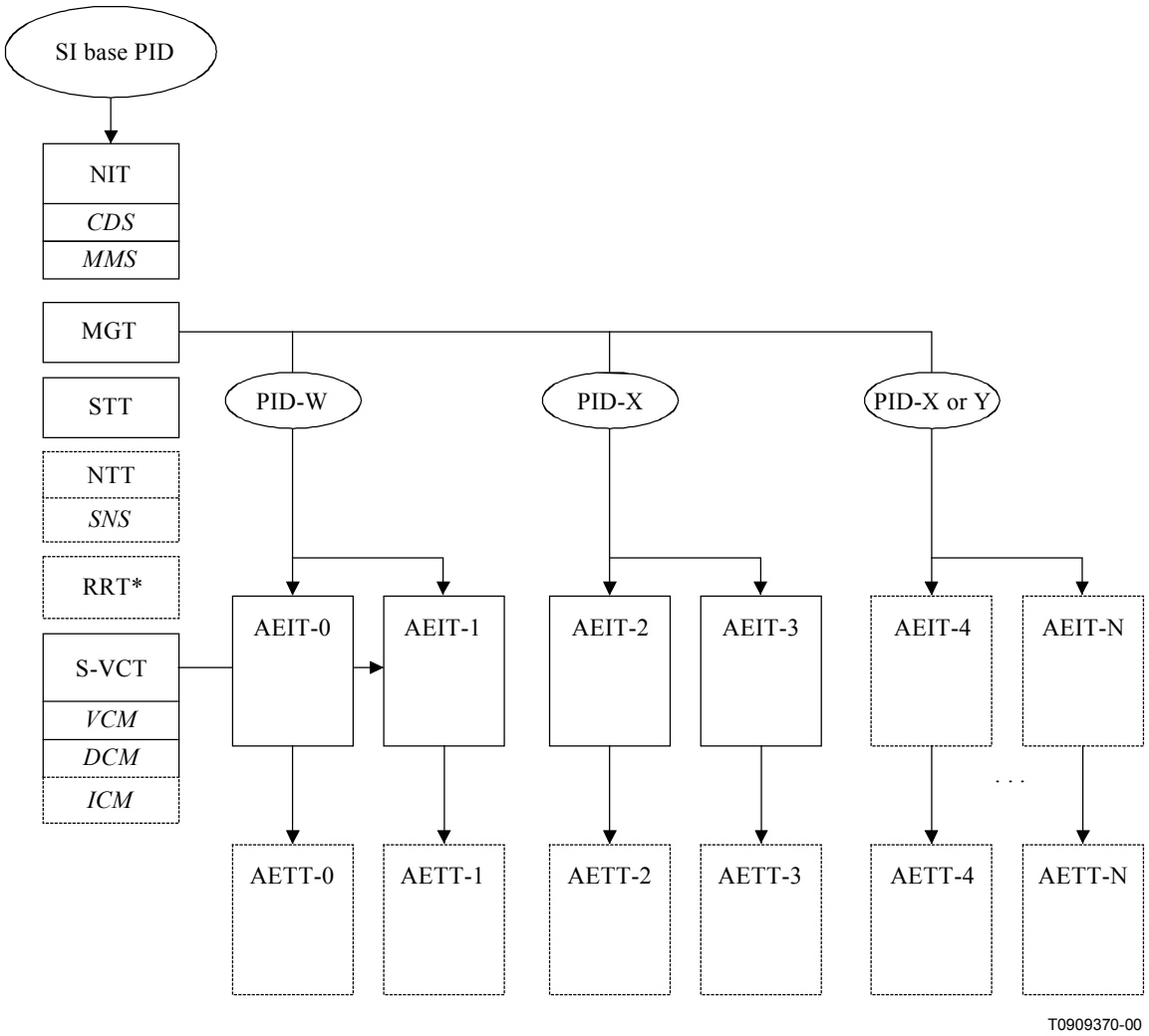

Figure B.II.3/J.94 – Hierarchy of Table Sections – Profile 4

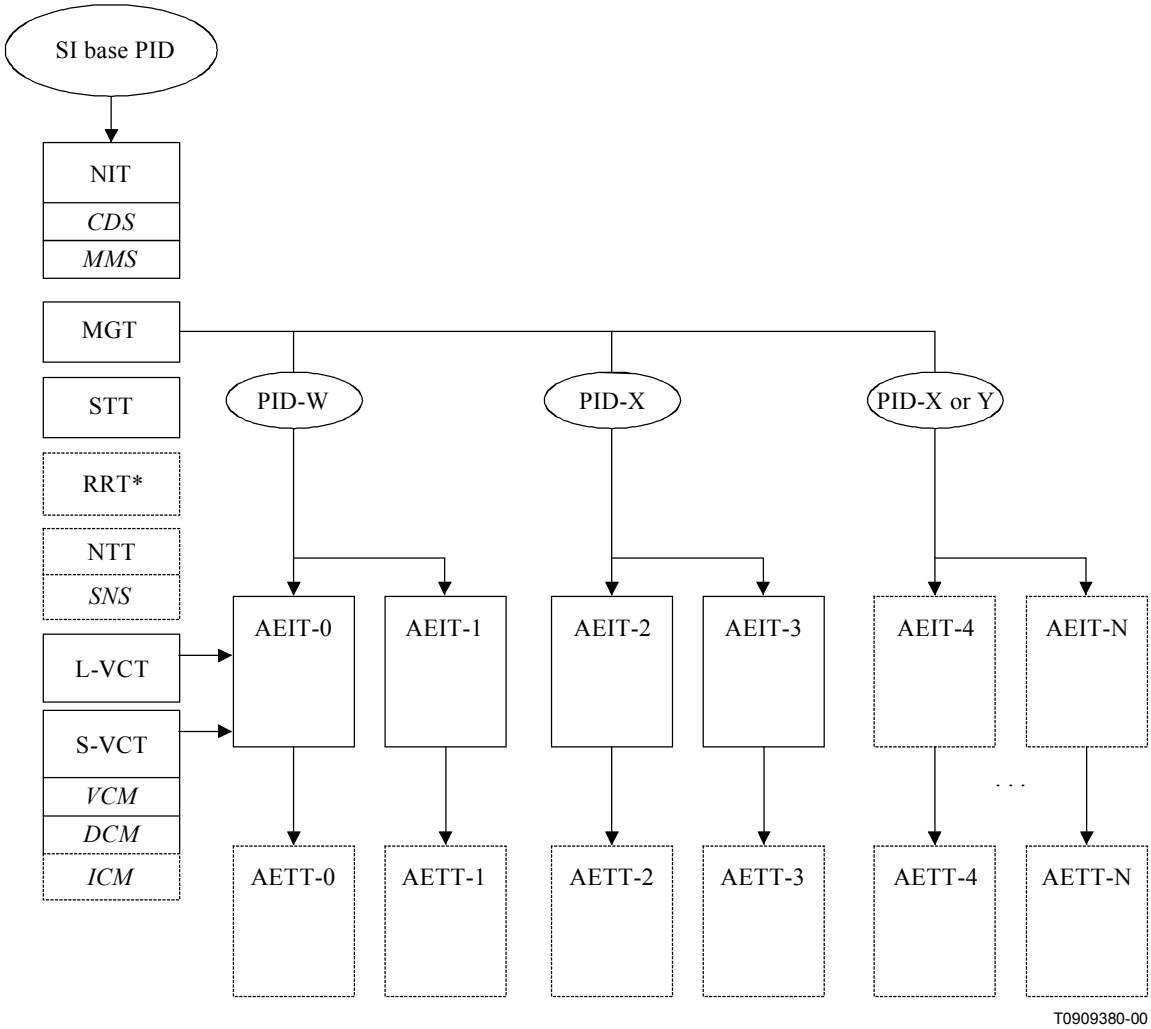

Figure B.II.4/J.94 – Hierarchy of Table Sections – Profile 5

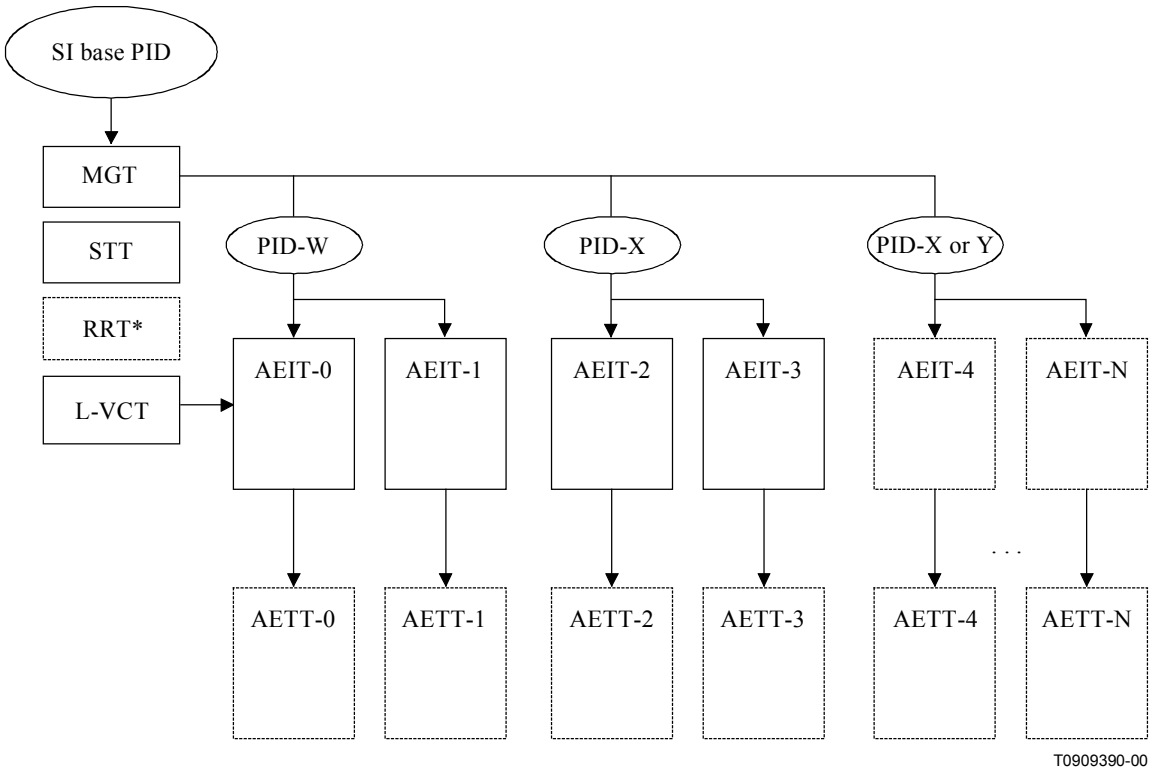

Figure B.II.5/J.94 – Hierarchy of Table Sections – Profile 6

The Short-form Virtual Channel Table section (table_ID 0xC4) or the Long-form Virtual Channel Table (table_ID 0xC9) provide navigation data on the out-of-band path. If MGT is provided, it references all tables present in Service Information (except the System Timetable).

The Master Guide Table provides general information about all of the other tables including the S-VCT, L-VCT, RRT, AEIT, and AETT. 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 identifier (PID) values associated with instances of AEITs and AETTs.

In Profile 3 and higher, the Rating Region Table must be included, with one exception, to describe rating regions in use. The exception is that delivery of version 0 of the RRT for region 0x01 (US and possessions), need not be sent because this table is standardized in EIA-766. Furthermore, for Profile , the MGT need not be sent if no RRT is sent.

Aggregate Event Information Tables are included in the out-of-band data in Profiles 4-6. Each AEIT instance describes the events or TV programs associated with a particular three-hour time slot. In the AEIT table structure, program schedule and title data for all virtual channels is aggregated together.

Each AEIT instance is valid for a time interval of three hours. As shown in Figure B.II.3, at minimum, AEIT-0 through AEIT-3 must be sent. Therefore, when Profiles 4-6 are used, current program information and information covering nine to twelve hours of future programming will be available to the Host.

Up to 256 AEITs may be transmitted; over 30 days of future programming may therefore be described. For the fourth timeslot and beyond (AEIT-4 through AEIT-N), the tables may be associated with the same or different PID values.

The start time for any AEIT is constrained to be one of the following UTC times: 00:00 (midnight), 03:00, 06:00, 09:00, 12:00 (noon), 15:00, 18:00, and 21:00. Imposing constraints on the start times as well as the interval duration simplifies re-multiplexing. During re-multiplexing, AEIT 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 AEIT by content in real time, which is a difficult task.

However, it is also possible to regenerate one or several AEIT at any time for correcting and/or updating the content (e.g. in cases where "to be assigned" events become known). Regeneration of an AEIT may be flagged by updating version fields in the MGT. A new AEIT may also be associated with a PID value not in current use. The MGT may be updated to show this new PID value association.

In Profiles 4-6, there can be several Aggregate Extended Text Tables, each of them having its associated PID defined in the MGT. As its name indicates, the purpose of an Aggregate Extended Text Table is to carry textual data. For example, for an event such as a movie listed in the AEIT, the typical data is a short paragraph that describes the movie itself. Each Aggregate Event Information Table can have one associated AETT. Each AETT instance includes all the text associated with events starting within a particular timeslot. Aggregate Extended Text Tables are optional in Profiles 4-6.

B.II.2 SI_base PID

Data associated with the SI_base PID defines information of system-wide applicability such as frequency plans, channel maps, and channel names. The SI_base PID value is 0x1FFC. The types of table sections that may be included in the Network Stream include:

– Network Information Table, carrying the:
  - Carrier Definition Subtable,
  - Modulation Mode Subtable;
– Network Text Table, carrying the Source Name Subtable;
– Short-form Virtual Channel Table, carrying the:
  - Virtual Channel Map,
  - Defined Channels Map,
  - Inverse Channels Map;
– Long-form Virtual Channel Table;
– Master Guide Table;
– Rating Region Table;
– System Timetable.

Carrier Definition Subtable

The Carrier Definition Subtable provides a foundation for the definition of frequency plans by defining a set of carrier frequencies appropriate to a particular transmission medium. The CDS is stored in the Host as an array of as many as 255 CDS records, each consisting of:

• Carrier frequency, 15 bits, in units of 10 or 125 kHz.

Modulation Mode Subtable

The Modulation Mode Subtable provides a foundation for quick acquisition of digitally modulated waveforms. A separate MMS shall be transmitted in Network data for each transmission medium supported by that network. An MMS is stored in the Host as an array of up to 255 MMS records, each consisting of:

- Modulation format: analogue NTSC or QAM;
- Transmission system: ITU-T (North America) or ATSC;
- Symbol rate, in units of 1 Hz;
- Inner coding mode, expressed as either "none" or an integer ratio such as 1/2 or 3/4;
- For QAM modulation, the number of levels.

Each MMS contains entries for each modulation mode currently in use by any digital waveform, plus entries for any modes anticipated to be used. As with the CDS, changes to the table are rare.

Parameters defined within the MMS are not specifically manipulated by Hosts compliant with the SI protocol, but are referenced by the Host when attempting to acquire a digitally encoded and modulated waveform.

Short-form Virtual Channel Table and Virtual Channel Record

The Short-form Virtual Channel Table is a hierarchical data structure that may carry within it the Virtual Channel Map and Virtual Channel record, for support of up to 4096 channel definition records. Each virtual channel is associated with a 16-bit reference ID number called the source_ID. Each record in the VCM consists of:

- The MPEG program number, associating the virtual channel record with a program defined in the Program Association Table and TS Program Map Table.
- For virtual channels associated with programs carried in a program guide, the source_ID, a number that may be used to link the virtual channel to entries in the Electronic Program Guide (EPG) database.
- For virtual channels used as access paths to application code or data (such as EPG), the *application ID*[7].

Source ID

Source ID is a 16-bit number associated with each program source, defined in such a way that every programming source offered anywhere in the system described in this Service Information annex is uniquely identified. For example, HBO/W has a different assigned source ID than HBO/E, and both are different from HBO-2 or HBO-3. Uniqueness is necessary to maintain correct linkages between an EPG database and virtual channel tables. See below for a discussion of the relationship between source_ID, virtual channels, and an EPG database.

Source Names and Source Name Subtable

The Source Name is a variable length multilingual text string associating a source ID with a textual name. The Source Name Subtable is delivered within the Network Text Table section.

---

[7] Source ID and application ID need never be defined in the same virtual channel record, therefore they share a common 16-bit field in the stored map. Channels are defined as for "application access" or not; if they are application access, the field defines the application ID, if not, it defines the source ID.

Source name information is delivered in a table format separate from the table containing other information comprising the virtual channel table. Name information is not strictly necessary for channel acquisition, and (depending on the memory management scheme employed in the Host) may not always be available from memory at acquisition time. Source name information may be refreshed often, and can be available within several seconds of acquisition.

An EPG database may define textual reference names associated with given program sources (referenced by source ID). Such a database may be used to derive virtual channel names in some applications, though in an EPG database the name is generally abbreviated due to display considerations.

Name data is, unlike the regular VCT data, language tagged, so that multilingual source names may be defined. Transmission format for multilingual text is defined to include references to multiple phonetic and ideographic character sets.

Defined Channels Map and Inverse Channels Map

For a given Standard-compliant channel, DCM data consist of a series of bytes that, taken as a whole, specify which channels in the map are defined, and which are not.

Each Virtual Channel Table has associated with it a table listing source_IDs and their associated virtual channel numbers. The source_ID values are sorted by value from the lowest to the highest in the table, to facilitate (using a binary search) lookup of a virtual channel given a source ID.

Master Guide Table

Use of the MGT is optional in certain profiles. Table B.II.1 shows a typical Master Guide Table indicating, in this case, the existence in the Transport Stream of a Long-form Virtual Channel Table, the Rating Region Table, four Aggregate Event Information Tables, and two Aggregate Extended Text Tables describing the first six hours' events.

Table B.II.1/J.94 – Example Master Guide Table content

table_type	PID	version_number	table size (bytes)
LVCT	0x1FFC	4	5 922
RRT – region 6	0x1FFC	0	1 020
AEIT-0 – MGT_tag = 56	0x1DD2	6	29 250
AEIT-1 – MGT_tag = 57	0x1DD2	4	28 440
AEIT-2 – MGT_tag = 58	0x1DD3	10	25 704
AEIT-3 – MGT_tag = 59	0x1DD3	2	27 606
AETT-0 – MGT_tag = 56	0x1DD2	2	24 004
AETT-1 – MGT_tag = 57	0x1DD2	7	25 922
AETT-2 – MGT_tag = 58	0x1DD3	8	27 711
AETT-3 – MGT_tag = 59	0x1DD3	0	19 945

The first entry of the MGT describes the version number and size of the Long-form Virtual Channel Table. The second entry corresponds to an instance of the Rating Region Table for region 6. 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.

The next entries in the MGT correspond to the four AEITs that must be supplied in the Transport Stream for profiles 4-6. After the AEITs, the MGT references four Aggregate Extended Text Tables. The PID values for AEIT-0 and AEIT-1 are both 0x1DD2. MGT_tag values 56 and 57 are used for these. For AEIT-2 AEIT-3, PID 0x1DD3 is used. The last four references are to Aggregate ETTs.

Note that AETT-n shares a common PID value with AEIT-n for every value of n. AEIT-0 and AETT-0 are associated with PID 0x1DD2, as are AEIT-1 and AETT-1. AEIT-2 and AETT-2 are associated with PID 0x1DD3, etc.

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 Host about the status of all the other tables (except the System Time which has an independent function). The MGT is continuously monitored at the Host to prepare and anticipate changes in the channel/event structure. When tables are changed at the broadcast side and the PID association is unchanged, their version numbers are incremented and the new numbers are listed in the MGT. Another method that can be used to change tables is to associate the updated tables with different PID values, and then update the MGT to reference the new PID values. Based on the MGT version or PID updates and on the memory requirements, the Host can reload the newly defined tables for proper operation.

Table B.II.2 is an example MGT that may be sent after the instance in Table B.II.2 has expired due to the passage of time. In this example, three hours have passed, and the time slot covered in the old AEIT-0 is in the past. The AEIT with MGT_tag = 57 moves now to become AEIT-0. The AEIT with MGT_tag = 58, the new AEIT-1, moves to PID 0x1DD2. A new AEIT is added to the mix, the AEIT with MGT_tag = 60.

Table B.II.2/J.94 – Example Revised Master Guide Table content

table_type	PID	version_number	table size (bytes)
LVCT	0x1FFC	4	5 922
RRT – region 6	0x1FFC	0	1 020
AEIT-0 – MGT_tag = 57	0x1DD2	4	28 440
AEIT-1 – MGT_tag = 58	0x1DD2	10	25 704
AEIT-2 – MGT_tag = 59	0x1DD3	2	27 606
AEIT-3 – MGT_tag = 60	0x1DD3	0	30 055
AETT-0 – MGT_tag = 57	0x1DD2	7	25 922
AETT-1 – MGT_tag = 58	0x1DD2	8	27 711
AETT-2 – MGT_tag = 59	0x1DD3	0	19 945
AETT-3 – MGT_tag = 60	0x1DD3	0	22 522

L-VCT

The L-VCT combines all the data pertinent to the description of a virtual channel into a single table. Use of the L-VCT instead of the S-VCT eliminates the need to send CDS, MMS, SNS, DCM, or ICM. The L-VCT follows the standard MPEG-2 long-form section syntax (section_syntax_indicator = 1).

Rating Region Table

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 clause 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 B.II.3 shows an example of one instance of an RRT, defined for rating region 99 and carrying an example rating system. 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.

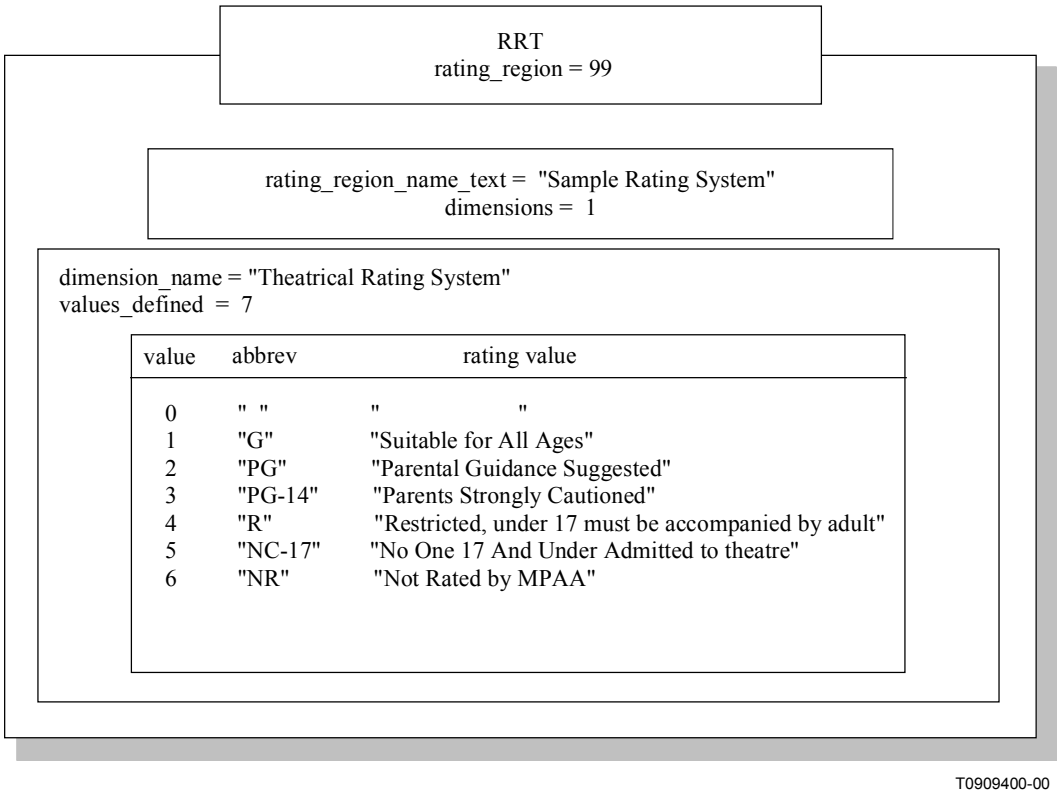

Figure B.II.3/J.94 – An instance of a Rating Region Table

Aggregate Event Information Tables and Aggregate Extended Text Tables

The purpose of an AEIT is to list all events for those channels that appear in the VCT for a given time window. As mentioned before, AEIT-0 describes the events for the first 3 hours and AEIT-1 for the second 3 hours. AEIT-0 and AEIT-1 share a common associated PID value as defined in the MGT. In MPEG, tables can have a multitude of instances. When different instances of a table share the same table_id value and PID, they are distinguished by differences in the 16-bit table_id_extension field.

In this SI appendix for out-of-band use, each instance of AEIT-k contains a list of events for a each virtual channel. Linkage to each channel in the VCT is made via the source_ID. For the AEIT, the table_id_extension field appears as MGT_tag.

Figure B.II.4 shows, for example, a program provider's instance for AEIT-0.

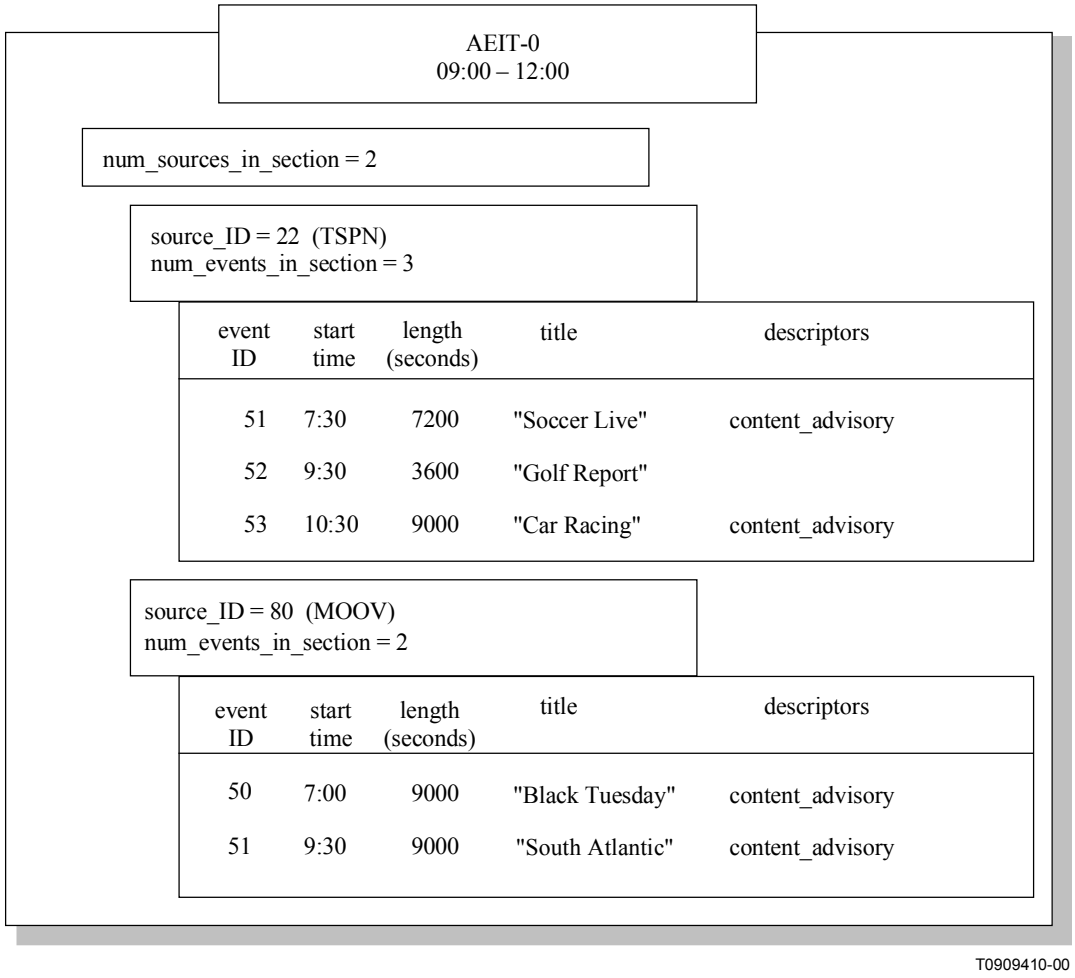

Figure B.II.4/J.94 – Example AEIT-0

AEIT-0 is unique in that it must list all events starting within the three-hour time period it covers, as well as any events that started earlier but extend into the covered period. For all other AEITs, only those events actually starting within the three hour time period are included. The Host is expected to collect AEITs in order of their time coverage. If AEIT-4 is available to the Host but AEIT-3 is not, for example, information for events that started in the time period covered by AEIT-3 but extending into AEIT-4 will not be available for display.

Figure B.II.4 shows an example of a small AEIT-0, including event data for two sources, a channel called "TSPN" (source_ID 22) and one called "MOOV" (source_ID 80). For the three-hour period covered by AEIT-0, 9 a.m. to noon, three events are listed for TSPN and two for MOOV. The field event_id is a number used to identify each event. The event_id is used to link events with associated text delivered in the AETT. The assignment of an event_ID value must be unique within a source ID and a 3-hour interval defined by one AEIT instance. The event_id is followed by the start_time and then the length_in_seconds. Notice that for AEIT-0 only, events can have start times before the activation time of the table. ETMs are simply long textual descriptions. The collection of ETMs constitutes an Aggregate Extended Text Table (ETT).

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.

Figure B.II.5 diagrams the AEIT data structure. As shown, the AEIT includes event data for all sources listed in the VCT. In the figure, the hatched box represents one or more "event data" blocks, each comprised of the data items shown in the upper left.

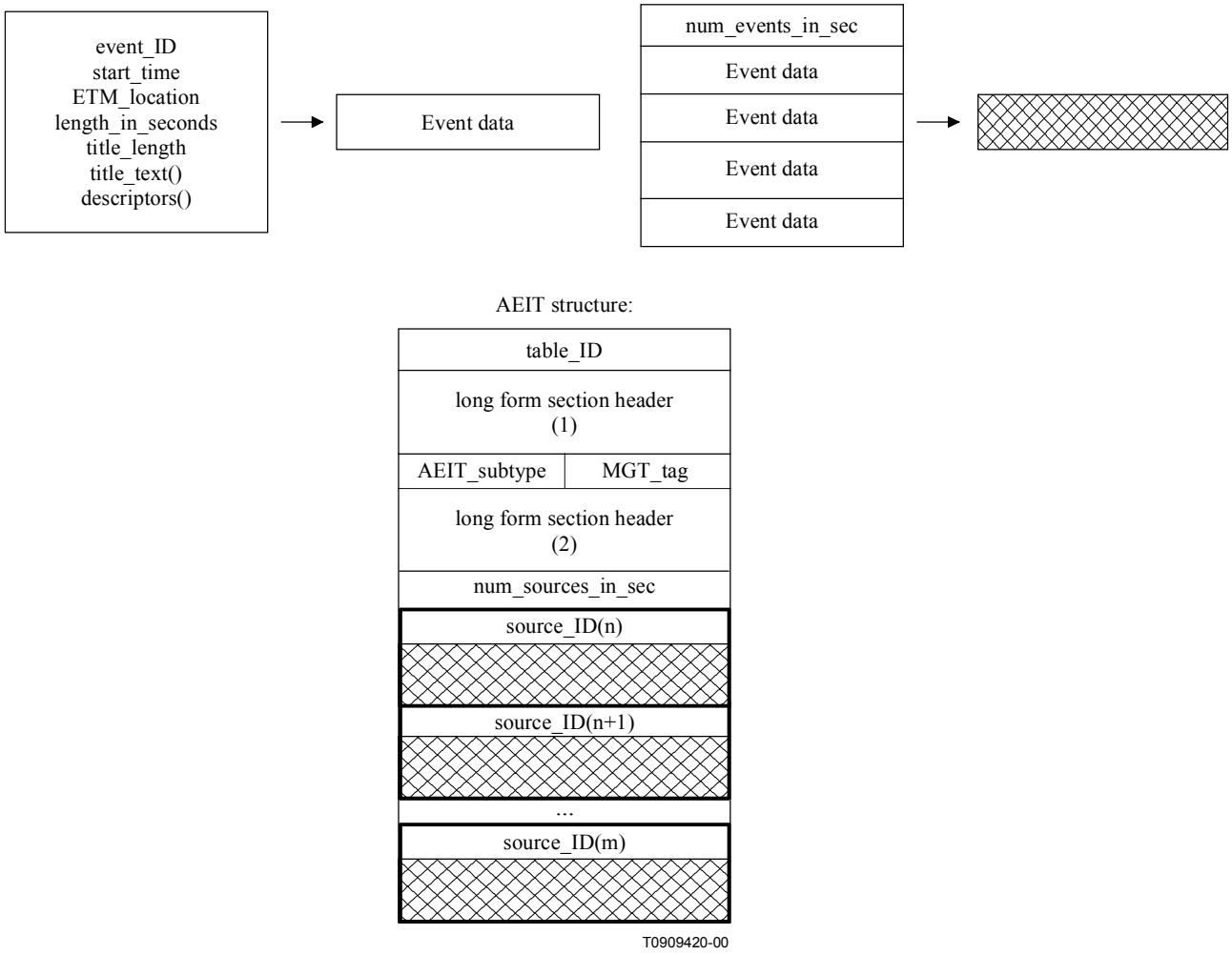

Figure B.II.5/J.94 – AEIT data structure

Figure B.II.6 diagrams the AETT data structure. The AETT aggregates text for a given timeslot into one sectioned MPEG table.

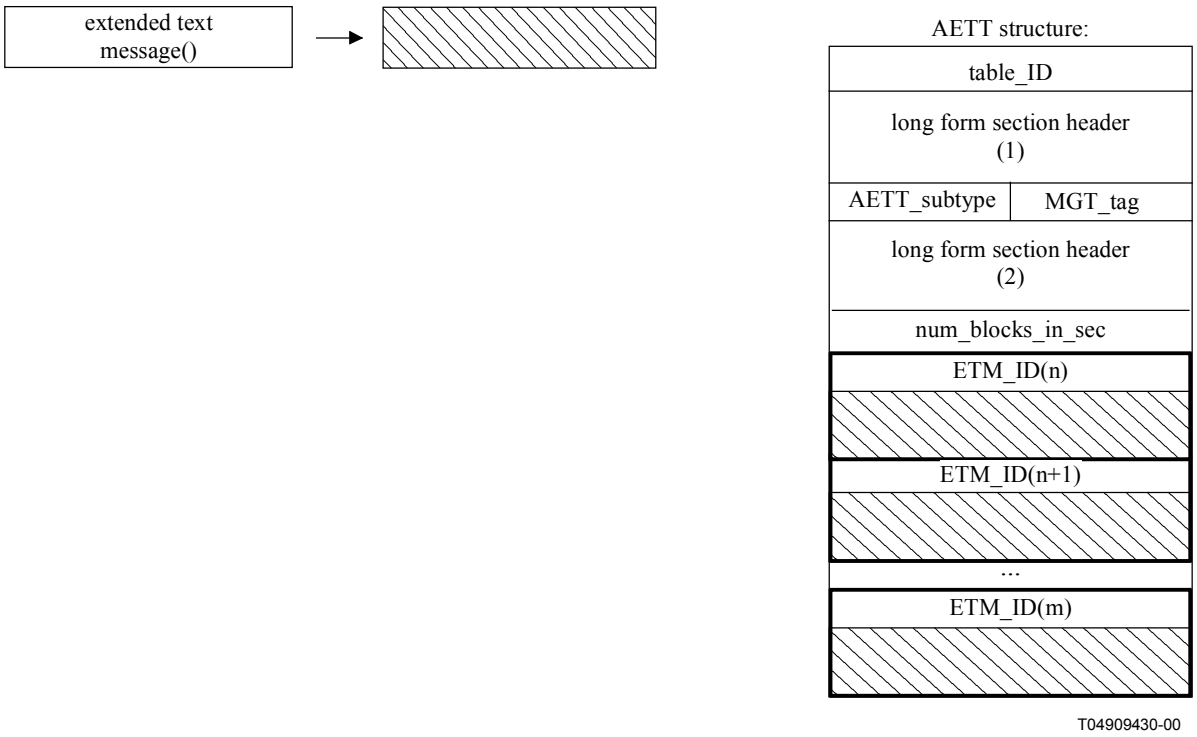

Figure B.II.6/J.94 – Structure of AETT

An AETT-*n* instance for a given value of *n* (timeslot) is associated with the same PID value as AEIT-*n*. This means that they can be collected using a single Extended Channel data flow between Host and POD.

Inactive Channels

Any channels in the L-VCT which are not currently active shall have the hidden attribute set to 1 and the hide_guide attribute set to 0. Inactive channels in the S-VCT shall have the hidden attribute in channel_type, and the hide_guide flag in the channel_properties_descriptor() set to 0.

Table B.II.3 shows expected DTV behavior for the various combinations of the hidden and hide_guide attributes. In the table the "x" entry indicates "don't care." A check in the "surf" column indicates the channel is available by channel surfing and via direct channel number entry. A check in the "guide" column indicates that the channel may appear in the program guide listing.

Table B.II.3/J.94 – Receiver Behavior with hidden and hide_guide attributes

hidden	**Hide_guide**	**Receiver Behavior**		
		Surf	Guide	
0	x	✓	✓	Normal channel
1	1			Special access only
1	0		✓	Inactive channel

B.II.3 Representation of Time

The System Timetable provides time of day information to Hosts. In this Service Information appendix, time of day is represented as the number of seconds that have elapsed since the beginning of "GPS time," 0000 Hours UTC, January 6th, 1980. GPS time is referenced to the Master Clock at the US Naval Observatory and steered to Coordinated Universal Time (UTC). UTC is the current time of day at the time zone local to Greenwich, England, and is the time source we use to set our clocks.

The cycle of the seasons, technically known as the tropical year, is approximately 365.2422 days. Using the Gregorian calendar we adjust for the fractional day by occasionally adding an extra day to the year. Every fourth year is a leap year, except that three leap years in every 400 are skipped (the centennial years not divisible by 400). With this scheme there are 97 leap years in each 400 year span, yielding an average year that is 365.2425 days long.

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

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

This protocol defines various time-related events and activities, including starting times for programs, text display, changes to VCTs, and others. Two methods of time distribution are used in headend systems. One method derives time in the form of GPS seconds from GPS Hosts. These Hosts also provide current GPS/UTC offset data. The second method of time distribution relies on the Internet Standard Network Time Protocol (NTP). NTP servers provide output in the form of UTC time, and do not provide GPS/UTC offset data. The Standard-compliant Host is synchronized to system time by the System Timetable, which provides time either in the form of GPS seconds since week zero of GPS time, January 6th, 1980, or directly in UTC time. The interpretation depends on the value of the GPS/UTC offset field. The special value of zero is used to indicate that the system is being driven by a UTC time source directly, and that GPS/UTC offset data is not available.

System Time

GPS satellites typically output GPS time in a format consisting of a week count (Tw) and a seconds within the week count (Ts), where week zero is defined as starting January 6th, 1980. For purposes of building the System Timetable, the following formula may be used:

$$T = (Tw * 604\,800) + Ts$$

There are 604 800 seconds per week.

When converting between GPS seconds and current local time in hours/minutes/seconds, the following factors must be taken into account:

- **GPS to UTC offset** – Given a time represented as GPS seconds, the Host first subtracts the GPS/UTC offset to convert to UTC.
- **1980** – The first year of GPS time started on January 6th, yielding 361 days in the first year (1980 was also a leap year).
- **Leap years** – The number of leap years that occurred between the current GPS second and 1980 must be accounted for. A leap year is a year whose number is evenly divisible by four, or, in the case of century years, by 400.

NOTE – According to this rule, the year 2000 *is* a leap year even though it is a century year, because it is also divisible by 400.

- **Time zones** – Time zones are signed integer values in the range –12 to +13 hours, where positive numbers represent zones east of the Greenwich meridian and negative numbers west of it. Pacific Standard Time (PST) is 8 hours behind standard time, and Eastern Standard Time (EST) is 5 hours behind. The system defined by this Service Information standard accommodates time zones that are not an integral number of hours offset from Greenwich by defining time zone as an 11-bit signed integer number in units of minutes. To convert to local time, the time zone is added to Greenwich time using signed integer arithmetic.
- **Daylight savings time** – If applicable, daylight savings time must be taken into account. On a unit by unit basis, each Host may be given a definition for when daylight savings time is entered into in Spring, and when it is exited in Fall. Entry/exit points are given as absolute times (GPS seconds), and hence are given in one second resolution.

Transmission Format for Event Times

In this messaging protocol, the absolute time of action is specified for most events in terms of an unsigned 32-bit integer number, the count of GPS seconds since January 6th, 1980. This count does not wrap until after the year 2116[8].

Handling of Leap Second Events

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

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

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

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

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

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

---

[8] Prior to that time, all initial Receivers will surely be out of service, and new ones can be designed to handle the wrap condition.

Handling of Leap Second Events

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

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

The data in the System Timetable is:

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

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

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

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

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

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

APPENDIX B.III

Daylight Savings Time control

In order to convert GPS into local time, the Host 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 Host memory.

Since there is a common time (GPS) transmitted in SI, a mechanism to indicate when the Host should switch into (or out of) daylight savings time at the appropriate local time can be very useful. Once all the Hosts 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 daylight_savings_time_descriptor() the STT. The basic use of daylight savings fields through the year is shown in Table B.III.1.

Table B.III.1/J.94 – 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

SERIES OF ITU-T RECOMMENDATIONS

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

Geneva, 2001