SMPTE 333-2008
Revision of
SMPTE 333M-1999

SMPTE STANDARD

DTV Closed-Caption Server to Encoder Interface

Table of Contents

Approved
April 25, 2008

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in Part XIII of its Administrative Practices.

SMPTE Standard 333 was prepared by Technology Committee D27.

1 Scope

This standard defines a serial interface for transmission of CEA-708 digital television closed caption data from a caption server to video encoder to be used for populating the closed-caption structure in the compressed video stream.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified the order of precedence of the types of normative information in this document shall be as follows. Normative prose shall be the authoritative definition. Tables shall be next, followed by formal languages, then figures, and then any other language forms.

3 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

SMPTE 334-2-2007, Caption Distribution Packet (CDP) Definition

CEA-708-C (2007), Digital Television (DTV) Closed Captioning

TIA 574 (2003), 9-Position Non-Synchronous Interface between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange[1]

[1] TIA 574 is an extension of TIA-232 (2002) that specifies a 9-pin connector and speeds faster than 20 kb/s; i.e., an RS-232 port on a personal computer (TIA-232 specifies neither a 9-pin connector nor speeds faster than 20 kb/s).

4 Definitions

4.1 Acronyms and Abbreviations

The following acronyms and abbreviations are used within this specification:

ATSC: Advanced Television Systems Committee.

DTVCC: Digital Television Closed Captioning.

4.2 Definition of Terms

The following terms are used throughout this standard:

caption server: A device that delivers DTVCC formatted data for insertion in picture level user-data within an MPEG video elementary stream.

cc_data triplet: A sequence of three 8-bit bytes which consists of an identifier/flag byte plus two data bytes. Normally just referred to as a "triplet" in this standard.

video encoder: A device that encodes video into a compressed format.

4.3 Section and Data Structure Syntax Notation

This standard 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**).

4.3.1 Syntactic Notation

The formats of sections and data structures in this standard are described using a C-like notational method employed in ISO/IEC 13818-1.

4.3.2 Numeric Notation

Within this standard, conventional numbers denote decimal values, numbers preceded by 0x are to be interpreted as hexadecimal values and numbers within single quotes (e.g., '10010100') are to be interpreted as a string of binary digits.

5 Physical Interface

The physical interface between the caption server and the video encoder shall be an TIA 574 interface, with restrictions as specified in § 5.2.

5.1 9-Pin Connector

The encoder shall be DCE and provide a female 9-pin connector as described in TIA 574 with the pin-out described in Table 1.

Table 1 – 9-pin connector pin-out

Pin	Signal	Signal source	Description
1	NC	N/A	Not connected
2	TXD	Video encoder	Serial data transmit
3	RXD	Caption server	Serial data receive
4	NC	N/A	Not connected
5	GND	N/A	Signal ground
6	NC	N/A	Not connected
7	RTS	Caption server	Request to send. The video encoder shall ignore the state of this pin.
8	CTS	Video encoder	Clear to send. When asserted, the caption server may transmit data to the video encoder.
9	NC	N/A	Not connected

5.2 Interface Parameters

The video encoder and caption server shall each provide an TIA-574 interface with the parameter settings from Table 2.

Table 2 – TIA 574 interface parameters

Parameter	Setting
Baud rate	38,400 b/s
Data bits	8
Parity	None
Stop bits	1
Start bits	1

6 Communications Protocol

The caption server shall provide data to the video encoder as the video encoder requests it. The communications protocol is described below.

6.1 Closed-Caption Packet

The caption server shall provide captioning data to the video encoder using the syntax described in Table 3. This data shall consist of one or more fields-worth of data. The data may arrive in any field order per CEA-708-C § 4.4.2.

Table 3 – Bit stream syntax for closed-caption packet

Syntax	Bits	Format
closed_caption_packet() {		
SOH	8	0x01
cc service available	1	bslbf
cc message type	7	uimsbf
cc message length	8	uimsbf
if (cc_message_type == 0x44) {		
for (i = 0; i < cc_message_length - 3; i += 3) {		
one bits	5	'11111'
cc_valid	1	bslbf
cc_type	2	bslbf
data_byte_1	8	bslbf
data_byte_2	8	bslbf
}		
} else if (cc_message_type == 0x53) {		
reserved	1	'1'
csn_size	1	uimsbf
If (csn_size == 1) {		
reserved	1	'1'
caption_service_number	5	uimsbf
} else		
caption_service_number	6	uimsbf
service_data_byte_1	8	bslbf
service_data_byte_2	8	bslbf
service_data_byte_3	8	bslbf
service_data_byte_4	8	bslbf
service_data_byte_5	8	bslbf
service_data_byte_6	8	bslbf
} else {		
for (i = 0; i < N; i++) {		
filler byte	8	bslbf
}		
}		
}		
packet_checksum	8	uimsbf
EOT	8	0x04
}		

SOH – ASCII SOH, fixed packet prefix.

cc_service_available – This 1-bit field is set to 1 to indicate that the caption server has caption service information available for transfer.

cc_message_type – This 7-bit unsigned integer specifies the type of table, based on Table 4.

Table 4 – cc_message_type values

cc_message_type	Meaning
0x00 – 0x43	SMPTE reserved
0x44	DTVCC/NTSC data
0x45 – 0x52	SMPTE reserved
0x53	Caption service data
0x54 – 0x6f	SMPTE reserved
0x70 – 0x7f	User reserved

cc_message_length – 8-bit field specifying the number of bytes in this packet, **SOH** through **EOT** inclusive.

one_bits – These fields shall be set to all '1's.

cc_valid – This 1-bit field indicates that the subsequent **data_byte_1** and **data_byte_2** are valid. This field carries the same meaning as CEA-708-C, § 4.4.1.

cc_type – This 2-bit field indicates the type of data carried in **data_byte_1** and **data_byte_2**. This field carries the same meaning as CEA-708-C, § 4.4.1.

data_byte_1, **data_byte_2** – These fields carry DTVCC or NTSC data. These fields carry the same meaning as CEA-708-C, § 4.4.1 and § 4.4.3.

csn_size – This 1-bit field signals the size of the caption_service_number field. The value of '1' shall indicate 5-bits, and the value of '0' shall indicate 6-bits.

caption_service_number – This field carries the caption service number for the service delivered by the following 6 service data bytes. This field shall have a value of 0x00 when the service data applies to the line 21 (CEA-608) service. Otherwise, this field shall have a value between 0x01 - 0x10 inclusive, when **csn_size** = '1', and 0x01 - 0x3F inclusive when csn_size = '0'. Note that the least significant bits of this field are a duplication of the least significant bits of **service_data_byte_4** (as specified in SMPTE 334-2) when the service data applies to one of the DTVCC services.

service_data_byte_n – These 6 fields carry caption service data for one service. These fields carry the same meaning as SMPTE 334-2, § 5.5.

filler_byte – This 8-bit field carries reserved information when **cc_message_type** is marked "reserved" in Table 4. These bytes shall take the value of 0xFF.

packet_checksum – This 8-bit field shall contain the 8-bit value necessary to make the arithmetic sum of the entire packet (**SOH** through **EOT**, inclusive) modulo 256 equal zero.

EOT – ASCII EOT, fixed packet postfix.

6.2 Encoder Requests and Responses

The video encoder shall request caption data and indicate status from the closed-caption server via messages formatted according to the syntax defined in Table 5. These messages are also used by the video encoder to indicate the status of the last transfer from the closed-caption server.

Table 5 – Bit-stream syntax for encoder requests and responses

Syntax	Bits	Format
encoder_req_resp() {		
service_data_inhibit	1	bslbf
req_or_resp	7	uimsbf
}		

service_data_inhibit – When set to 1, the caption server shall not send a **closed_caption_packet()** with **message_type** = 0x53 in response(s) to this **encoder_req_resp()**. If **req_or_resp** == 0x06 or **req_or_resp** == 0x15, this field shall be ignored.

req_or_resp – This 7-bit field shall have values from Table 6.

Table 6 – Encoder requests and responses

Value	Name	Meaning
0x06	ACK	Previous transfer acknowledged
0x15	NAK	Previous transfer was not received, or was not completely received, or was received with an invalid checksum.
0x1a	SYN0[1]	Transfer **closed_caption_packet()** with **cc_message_type** == 0x44 and **cc_message_length** == 0x05.
0x1b	SYN5	Transfer **closed_caption_packet()** with **cc_message_type** == 0x44 and **cc_message_length** == 0x14.
0x1c	SYN10	Transfer **closed_caption_packet()** with **cc_message_type** == 0x44 and **cc_message_length** == 0x23.
0x1d	SYN15	Transfer **closed_caption_packet()** with **cc_message_type** == 0x44 and **cc_message_length** == 0x32.
0x1e	SYN20	Transfer **closed_caption_packet()** with **cc_message_type** == 0x44 and **cc_message_length** == 0x41.
0x1f	SYN25	Transfer **closed_caption_packet()** with **cc_message_type** == 0x44 and **cc_message_length** == 0x50.
[1]Note that SYNx is used to refer to SYN0, SYN5, SYN10, SYN15, SYN20, or SYN25.		

6.3 Determination of the Value 'x' for SYNx (Informative)

A caption data stream compliant with CEA-708-C is required to be at 9600 bps regardless of the frame rate of the video signal to be compressed. This Standard is designed to allow an encoder to request the CEA-708-C data one video frame at a time. The different SYNx commands support the different encoder video frame rates. For SYNx, the 'x' indicates how many cc_data triplets are to be sent from the server to the encoder. A

message typically is comprised of CEA-608-C triplets followed by CEA-708-C triplets and CEA-708-C filler triplets. The relationship between 'x' and the video frame rate can be expressed as follows:

$$x = \frac{9600\,bps}{fps * (2\,bytes\ per\ triplet) * (8\,bits\ per\ byte)}$$

For example, at 60 fps, 'x' is 10; 30 fps, 'x' is 20.

It is expected that the video encoder will reframe caption data when necessary to handle film-based material while pulldown repetitions are present and being removed to create 24 frames per second encoded output. It is noted that the video encoder must handle the resultant reframing when video material without pulldown is edited into film-based material breaking the pulldown cadence, per CEA-708-C, § 4.4.2.

The reader is referred to § 4.4.1 and § 4.4.2 of CEA-708-C for further information.

6.4 Bit Rate Jitter to Reach Average Bit Rate

Implementers must remember that each component of this system (a video encoder and a closed caption server) are each synchronized independently to the same video source. As the caption server may be slaved via time code samples rather than direct video, the situation is even less deterministic. As a result, each component must expect some unexpected behavior from the other, especially when dealing with film material delivered via 29.97 fps video.

6.4.1 Video Encoder Required Jitter

Although CEA-708-C requires that the bit rate of caption data be fixed at 9600 bps, for certain frame rates and telecine situations, this does not allow an integer number of caption data packets. Therefore, the video encoder may send Encoder_req_resp() SYNx requests that vary request by request. This is required by § 4.4.2 of CEA-708-C.

6.4.2 Caption Server Required Jitter

In a manner analogous to that of the video encoder, should the caption server (which is also slaved to the incoming video rate) fail to deliver all of the caption data made available to it, it shall deliver it to the video encoder in the next available Encoder_req_resp() SYNx request, with the data order differing, perhaps, from that assumed by the video encoder. It is the responsibility of the video encoder to properly reframe the data as appropriate (in compliance with § 4.4.2 of CEA-708-C). This includes handling field 1 and field 2 order reversals.

6.5 Startup Case

At startup, the video encoder shall initiate communication by sending an encoder_req_resp() with req_or_resp = SYNx. Timeout shall be as described in § 6.8.

At startup, the caption server shall be in a quiescent state.

6.6 Normal Case

The video encoder shall send **encoder_req_resp()** with **req_or_resp** = SYNx to request DTVCC data. The caption server shall respond to an **encoder_req_resp()** packet with **req_or_resp** =SYNx with a **closed_caption_packet()** with **cc_message_type** == 0x44.The video encoder shall respond to a **closed_caption_packet()** with **cc_message_type** == 0x44 with an **encoder_req_resp()** with **req_or_resp** = ACK or NAK.

If the **encoder_req_resp()** with **req_or_resp** = SYNx was sent with **service_data_inhibit** == 0, the caption server may send a **closed_caption_packet()** with **cc_message_type** == 0x53 after sending a **closed_caption_packet()** with **cc_message_type** == 0x44. In this case, the video encoder shall respond to the **closed_caption_packet()** with **cc_message_type** == 0x53 with an **encoder_req_resp()** with **req_or_resp** = ACK or NAK.

The caption server should send a **closed_caption_packet()** with **cc_message_type** == 0x53 when there is new caption service information available for transmission, as indicated by **cc_service_available** in the previous **closed_caption_packet()** with **cc_message_type** == 0x44.

If caption service information becomes available after the **cc_service_available** bit in a **closed_caption_packet()** has been sent, a **closed_caption_packet()** with **cc_message_type** == 0x53 shall not be sent until after the video encoder has sent an **encoder_req_resp()** with **req_or_resp** = SYNx and **service_data_inhibit** == 0.

6.7 Server Response (Informative)

The caption server usually begins responding to the video encoder's request (under the above conditions) within 10 msec of receipt of the encoder_req_resp() with req_or_resp = SYNx. Care must be taken to ensure that the serial link runs at full rate to permit the video encoder sufficient time to associate the caption data with the proper video frame.

6.8 Timeout

If the video encoder has sent an **encoder_req_resp()** with **req_or_resp** = SYNx, and a complete **closed_caption_packet()** message has not been received within 500 ms of the transmission of the **encoder_req_resp()** with **req_or_resp** = SYNx, the video encoder shall discard any partial **closed_caption_packet()** message received and restart communication by sending an **encoder_req_resp()** with **req_or_resp** = SYNx.

If the caption server has sent an entire **closed_caption_packet()** message, the caption server shall ignore subsequent **encoder_req_resp()** with **req_or_resp** = SYNx until one of the following conditions have been satisfied: 500 ms have passed since transmission of the last byte of the **closed_caption_packet()** message, or an **encoder_req_resp()** with **req_or_resp** = ACK is received, or an **encoder_req_resp()** with **req_or_resp** = NAK is received.

6.9 Caption Services

6.9.1 Addition of caption services

The caption server shall indicate additional caption services by sending a **closed_caption_packet()** with a **caption_service_number** different from the existing values of **caption_service_number**.

6.9.2 Modification of caption services

The caption server shall indicate modifications of existing caption services by sending a **closed_caption_packet()** with a **caption_service_number** value that duplicates an existing **caption_service_number**. The values of **service_data_byte_n** shall be interpreted to be correct for the **caption_service_number** specified, effective immediately.

6.9.3 Removal of caption services

The caption server shall indicate removal of existing caption services by sending a **closed_caption_packet()** with a caption_service_number value that duplicates an existing caption_service_number value, with **service_data_byte_n** set to the value zero (0x00).

6.10 Protocol State Tables

The message exchange protocol state for the video encoder is fully described in Table 7. Note the following shorthand: ccp(a) means **closed_caption_packet()** received with **cc_message_type** = a. The message exchange protocol state for the caption server is fully described in table 8. Note the following shorthand: ccp(a,b) means **closed_caption_packet()** sent with **cc_message_type** = a, **caption_service_available** = b.

In case of apparent conflict with the wording describing operations in written text and the language in the Protocol State Tables, the tables shall take precedence.

Table 7 – Video encoder protocol states

State number	Condition	Action	Next state	Timer
Initial state	None	Set local variable **caption_service_available** to 0	1	
1	**caption_service_available** == 0	Send SYNx **service_data_inhibit** ignored	2	Start T1 500-ms countdown
	caption_service_available == 1 and caption service data desired	Send SYNx **service_data_inhibit** = 0		
	caption_service_available == 1 and caption service data not desired	Send SYNx **service_data_inhibit** = 1		
2	Receive complete ccp (0x44), checksum matches. Local variable **caption_service_available** == 1 and last sent **service_data_inhibit** = 0	Send ACK, save received **caption_service_available** in local variable	3	Start T1 500-ms countdown
	Receive complete ccp (0x44), checksum matches.	Send ACK, save received **caption_service_available** in local variable	1	Stop T1
	Receive complete ccp (0x44), checksum does not match. Previous **caption_service_available** == 1 and last sent **service_data_inhibit** = 0	Send NAK, save received **caption_service_available** in local variable	3	Start T1 500-ms countdown
	Receive complete ccp (0x44), checksum does not match.	Send NAK, save received **caption_service_available** in local variable	1	Stop T1
3	Receive complete ccp (0x53), checksum matches.	Send ACK	1	Stop T1
	Receive complete ccp (0x53), checksum does not match.	Send NAK	1	Stop T1
Any	T1 expires	Discard buffered data, set local variable **caption_service_available** to 0	1	Stop T1

Table 8 – Caption server protocol states

State number	Condition	Action	Next state	Timer
Initial state	None		1	
1	Received SYNx; new caption service data not available	Send ccp(0x44, 0)	2	Start T2, 500-ms countdown
	Received SYNx; new caption service data are available, **service_data_inhibit** == 1	Send ccp(0x44, 1)	2	
	Received SYNx; new caption service data are available, **service_data_inhibit** == 0	Send ccp(0x44, 1)	3	
2	ACK received		1	Stop T2
	NAK received			
3	ACK received, more caption service data are available	Send ccp(0x53, 1)	4	Start T2, 500-ms countdown
	NAK received, more caption service data are available	Send ccp(0x53, 1)		
	ACK received, more caption service data are not available	Send ccp(0x53, 0)		
	NAK received, more caption service data are not available	Send ccp(0x53, 0)		
4	ACK received		1	Stop T2
	NAK received			
Any	T2 expires		1	Stop T2

Annex A (Informative)
Bibliography

SMPTE 334-1-2007, Vertical Ancillary Data Mapping of Caption Data and Other Related Data

SMPTE RP 2007-2007, Closed-Caption CDP and "Grand Alliance" Serial Interfaces for DTV

SMPTE EG 43-2004, System Implementation of CEA-708-B and CEA-608-B Closed Captioning

ATSC A/53 Part -4:2007, Digital Television Standard, MPEG-2 Video System Characteristics

ATSC A/65C (2 January 2006), Program and System Information Protocol for Terrestrial Broadcast and Cable, Revision C

ATSC A/76A (18 September 2006), Programming Metadata Communication Protocol Standard, Revision A

CEA-608-C (2006), Line 21 Data Services

ISO/IEC 13818-1:2006, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Systems

TIA 232 (2002), Interface between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange

Annex B (Informative)
Related Standards

Users are advised that caption data may be carried in the Vertical Ancillary (VANC) data areas of a serial digital bitstream. This is standardized by SMPTE 334-1. In addition, SMPTE RP 2007 documents an additional protocol for the transfer of caption data along with additional metadata using a streaming-type method (known informally as the "Grand Alliance" protocol), as opposed to the request/response type protocol defined in this standard.

A protocol for carriage of the Caption Service Descriptor, required to be associated with the caption data, is defined in ATSC A/76A (2006) Programming Metadata Communication Protocol (PMCP).