SMPTE STABLE DOCUMENT

The attached SMPTE Engineering Document has been declared "Stable" by the controlling Technology Committee.

The SMPTE Operations Manual for Standards states:

A document should be stabilized if it is believed to be substantially correct, does not contain harmful or misleading recommendations, may still be relevant to equipment or practices in use, is stable, but does not represent current technology, and need not be subject to future reviews.

A Stable document shall still be made available and offered for sale by the Society, but it shall be prefaced by a cover page explaining its current status.

At any time, a Technology Committee may revise, amend, or otherwise initiate a new Project on a Stable document.

A Stable document is "In Force", and not deprecated or withdrawn.

* * * * * * * * * *

Note:

SMPTE "Stable" documents were previously described as "Archived" and the attached document may be marked as "Archived". The status of a SMPTE document described as "Archived" is exactly as described above for a "Stable" document.

Stable documents may not adhere to the latest style and format of SMPTE documents, or to current usage of normative language. Suitable care should be taken in interpretation.

SMPTE 360-2009
Revision of
SMPTE 360M-2004

SMPTE STANDARD

General Exchange Format

Table of Contents

Page

Approved
February 11, 2009

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. This SMPTE Engineering Document was prepared by Technology Committee N26.

Intellectual Property

At the time of publication no notice had been received by SMPTE claiming patent rights essential to the implementation of this Standard. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

Introduction

This Standard describes one file format that can be used to move simple clips or compound clips. It also describes how the file transfer protocol (FTP) can be used for the transfers. This includes file name syntax to support an implementation of partial stream transfers and other extensions.

The byte streams described in this Standard can be moved using any reliable transfer protocol and a network that supports that protocol. Annex A describes one protocol stack and a network technology that are in use today. There is no reason other protocols such as XTP could not be used. XTP supports features such as reliable multicast which are not part of FTP.

In many applications, real-time broadcast schemes are used. The byte stream for simple clips described in this Standard can be used under those conditions. However, this practice is not recommended for compound clips.

The EBU/SMPTE Joint Task Force report also describes objectives for rich file formats and rich metadata capabilities in future designs. The file format described in this Standard was in place before the task force report was issued and therefore does not meet all of the report's stated objectives.

This type of transfer was initially implemented on a Fibre Channel network using FTP with transmission control protocol (TCP) and Internet protocol (IP). This transfer technique is well suited to implementations on IP networks such as Fibre Channel, Ethernet, or ATM, although it can be used on any reliable protocol stack and data network technology. An example of transfers over an IP link using the IETF standard FTP is provided in Annex A.

1 Scope

This Standard specifies a file exchange format for the transfer of simple clips and compound clips between television program storage systems. This Standard provides documentation of formats and contents, including specification of packet types, contents, and order. Informative information on a control scheme, a protocol stack, and an interface is included.

Physical links (data network technology) are not specified in this Standard. Any appropriate physical link may be used.

Simple clips and compound clips, as defined in this Standard, are specified as alternatives to traditional approaches to transferring television programming between storage devices.

This Standard defines two simple constructs for transfer of material:

1) A simple clip format in which all audio, video and time code tracks start and stop coincidentally and where the audio and video tracks are contiguous, and;

2) A compound clip format, which is a collection of audio, video, and time code tracks, any of which may contain several segments, with cut transitions between the segments. Transitions may occur on each track independently of the other tracks.

This Standard is independent of material format, whether compressed or not, and allows material to be transferred across a network before recording has finished. The transferred material specified in this Standard includes Motion JPEG (MJPEG), MPEG-1, MPEG-2, and DV-based video compression standards, with associated audio, time code, and user data that may include user-defined metadata. The video, audio, and time code material is time multiplexed to provide immediate availability at the receiver.

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.

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 registered disclosure document. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this registered disclosure document are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

SMPTE 12M-1-2008, Television — Time and Control Code

SMPTE 292-2008, 1.5 Gb/s Signal/Data Interface

SMPTE 314M-2005, Television — Data Structure for DV-Based Audio, Data and Compressed Video — 25 and 50 Mb/s

SMPTE 337-2008, Format for Non-PCM Audio and Data in an AES3 Serial Digital Audio Interface

SMPTE 338-2008, Format for Non-PCM Audio and Data in AES3 — Data Types

SMPTE 340-2008, Format for Non-PCM Audio and Data in AES3 — ATSC A/52B Digital Audio Compression Standard for AC-3 and Enhanced AC-3 Data Types

SMPTE RP 202-2008, Video Alignment for Compression Coding

AES3-2003, AES Standard for Digital Audio — Digital Input-Output Interfacing — Serial Transmission Format for Two-Channel Linearly Represented Digital Audio Data

ANSI INCITS 4-1986 (R2002), Information Systems — Coded Character Sets — 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII) [formerly ANSI X3.4-1986 (R1997)]

ATSC A/52:2005, Digital Audio Compression (AC-3) Standard

IEEE 754-1985 (R1990), Standard for Binary Floating-Point Arithmetic

ISO/IEC 11172-2:1993, Information Technology — Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to About 1.5 Mb/s — Part 2: Video

ISO/IEC 11172-2:1993/Cor 1:1996

ISO/IEC 11172-2:1993/Cor 2:1999

ISO/IEC 13818-1:2007, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Systems

ISO/IEC 13818-1:2007/Cor 1:2008

ISO/IEC 13818-2:2000, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Video

ISO/IEC 13818-2:2000/Cor 1:2007

ISO/IEC 13818-2:2000/Cor 2:2007

ISO/IEC 13818-2:2000/Amd 1:2001 — Support for Colour Spaces

ISO/IEC-61834-2 (1998-08), Recording — Helical-Scan Digital Video Cassette Recording System Using 6.35mm Magnetic Tape for Consumer Use (525-60, 625-50 and 1250-50 Systems), Part 2: SD Format for 525-60 and 625-50 Systems

ISO/IEC 10918-1:1994, Information Technology — Digital Compression and Coding of Continuous-Tone Still Images (JPEG)

ISO/IEC 10918-1:1994/Cor 1:2005 ITU-R BT.601-5 (10/95), Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios

ITU-R BS.1196 (1995) (Annex 2), Audio Coding for Digital Terrestrial Television Broadcasting

ITU-T H.261 (03/93), Video Codec for Audiovisual Services at p x 64 kb/s

ITU-T T.81 (1992) Information Technology – Digital Compression and Coding of Continuous-Tone Still Images: Requirements and Guidelines (JPEG)

4 Definitions

The following terms are defined for use in this Standard.

Notes:

1) For standard definition material, the traditional style of total number of lines per frame (525 and 625) has been used. For high-definition material, the new convention of active lines per frame (720 and 1080) has been used

2) The terminology for MPEG video streams includes “MPEG” when describing an arbitrary MPEG video type. The terms “MPEG-1” and “MPEG-2” are used when a specific version of MPEG is required. MPEG-1 video is always standard definition. When a MPEG-2 stream is standard definition video in either the 525 or 625 format, the term ”MPEG-2 SD” is used. Finally “MPEG-2 HD” is used when a MPEG-2 stream is a high definition video stream. GXF MPEG video streams are always encoded as single frames of MPEG Elementary Stream video elements.

4.1 clip: Clips shall consist of collections of time code, audio, and compressed video tracks. The contents of the tracks determine whether the material is a simple clip or a compound clip. See also simple clip and compound clip.

4.2 compound clip: In a compound clip, each track may be comprised of one or more segments arranged in a specified sequence as shown in Figure 1. Each segment of material on a track of a compound clip shall originate from a single source media file. Segments may comprise all or part of the material from a source media file. If only part of the material is used, it shall be a single contiguous segment delimited by a mark in and a mark out point. Transitions between segments shall be cut transitions.

Video track 1	Media segment V1	Media segment V2
Audio track 1	Media segment A1	Media segment A3
Audio track 2	Media segment A2	Media segment A4
Time code	Media segment T1	

Figure 1 – Example of a compound clip

4.3 data types: Table 1 describes the data types used in this Standard.

Numerical values in this Standard that are preceded by 0x are hexadecimal values. All other numerical values are decimal numbers.

Table 1 – Data type descriptions

Data type	Description
BINARY	8-bit binary data without an implicit length
INT32	32-bit signed integer
UINT64	64-bit unsigned integer
UINT32	32-bit unsigned integer
UINT16	16-bit unsigned integer
UINT8	8-bit unsigned integer
CHARACTER	A single character code using ANSI INCITS 4-1986
STRING	An ordered vector of ANSI INCITS 4-1986 coded characters that is 0x00 (NULL) terminated. NOTE: Unicode is not allowed.
FLOAT64	64-bit signed floating point (see IEEE 754)

The byte order for multibyte values defined in this Standard shall be the least significant byte (LSB) first, followed by progressively more significant bytes to the most significant byte (MSB) unless otherwise specified. Multibyte values in predefined items such as MPEG and DV compressed video streams shall be as stored/transported in their native order.

4.4 field identifier: The field identifier indicates the field number within the associated video frame:

1 = Field 1
2 = Field 2

For interlaced video, the field identifier alternates between 1 and 2. For progressive video, the field identifier is 1 on every field/frame.

Note: The field 1/2 nomenclature is always used, even when traditional standards would use field 0/1.

4.5 field locator table: The optional field locator table (FLT) provides a method to locate the desired material in the stream without processing the entire stream. The FLT shall contain up to 1000 offsets into the stream, which can be used to locate a point in the stream near a specified field. If an FLT is not present, unneeded fields or frames may have to be processed in order to locate the desired field or frame. The FLT facilitates partial file transfers from devices such as data tape archives by providing the ability to locate the desired portion of the streamed material without reading the entire stream.

4.6 field number: An unsigned integer that designates the position of a video field on a playout time line. The field number time line shall be zero-based and increment with each field. Frame based compression systems shall represent frames numbers with even field numbers.

4.7 field and frame rates: The field rate for 625-line material is exactly 50 fields per second. The field rate for 525-line material is documented as 59.94 fields per second. The actual field rate is 60/1.001 fields per second. The field rate for 1080-line interlaced material can be 60 fields/sec, 59.94 (the exact rate is 60/1.001) fields/sec, or 50 fields/sec. The field rate for 1080-line progressive material can be 30 frames/sec, 29.97 (the exact rate is 30/1.001) frames/sec, 25 frames/sec, 24 frames/sec, or 23.98 (the exact rate is 24/1.001) frames/sec. The field rate for 720-line material can be 50 frames/sec, 60 frames/sec, or 59.94 (the exact rate is 60/1.001) frames/sec (see SMPTE 292, Table 1).

4.8 first field number: The first field number shall represent the position on a playout time line of the first recorded field on a track (see field number above).

Note: The first/last field numbers describe the recorded material on a track. The mark in/out field numbers delimit the active (played) portion of a track.

4.9 last field number: The last field number shall represent the position on a playout time line of the last recorded field plus one (see field number above).

4.10 map packet: The map packet contains information (metadata) describing segments and tracks in a stream. The map packet includes information such as the type of each track and the media file name. Map packets shall be repeated in the stream. Map packet values may change within a stream. Examples of values that may change include media file names in the case of compound clips, and the size of the stream if the material is streamed while being recorded.

4.11 mark in field number: The mark in field number shall represent the position on a playout time line of the first field to be played from a track (see field number above).

Note: The mark in/out field numbers delimit the active (played) portion of a track. The first/last field numbers describe the material recorded for a given track.

4.12 mark out field number: The mark out field number shall represent the position on a playout time line of the last field to be played plus one (see field number above).

4.13 material: Material is a collection of audio, video, time codes, data, and synchronization information that may be source or finished content.

4.14 media file: A media file conceptually defines the storage format of a single track on a source or receiving device. The format of this file is not specified in this Standard. Conformance with this Standard shall only require a logical media file to be capable of providing or accepting the video, audio, time code, or data values and their associated time line positions or synchronization information.

Note: A compliant media file can be generated by an encoder, received by a decoder, stored on a file storage system, or processed by any other device that can provide or accept material.

4.15 media file name: A media file name shall be a string with the name of the media file on the source device that contains the material. The null media file name shall be used for segments that have not been recorded or otherwise provided. A null media file name shall have a 0x00 (NULL) as its first character. Receiving devices may use media file names at their discretion.

4.16 media packet: The stream packet that contains audio sample words, time code entries, or a field or frame of compressed video. Any audio or time code packet may have unused entries if there are unrecorded sections in the clip.

4.17 PCM audio: Linearly digitized audio samples represented in two's compliment integers. The integer sample size varies.

4.18 receiver: The receiver is the device that receives a streaming transfer. The receiver may store the stream as is, convert it another format, display the contents of the stream, or perform other functions.

4.19 reserved codes: These are code values that should not be emitted by a source and shall be ignored by a receiver.

4.20 reserved tags: These are tag values that should not be emitted by a source and shall be ignored by a receiver.

4.21 reserved values: These are data values that should be coded as zero by a source unless otherwise indicated. A receiver shall ignore these.

4.22 segment: A single contiguous set of audio samples, time code values, or video fields or frames of material. The boundaries of a segment shall be defined by the mark in and mark out field numbers. A segment may reference material that is outside the bounds of the first and last field numbers of the track.

4.23 simple clip: In a simple clip, each track shall be comprised of a single segment as shown in Figure 2. The content on any track shall originate from a single source media file.

Video track 1	Media segment V1
Audio track 1	Media segment A1
Audio track 2	Media segment A2
Time code	Media segment T1

Figure 2 – Example of a simple clip

4.24 source: The source is the device that provides or generates the stream.

4.25 striped time codes: Striped time codes are a virtual track of time codes that are calculated given a starting time code value and successive frame numbers.

4.26 track: A track is an ordered collection of material samples. Each sample represents a video field, a video frame, an audio sample, or a time code. Each sample includes an actual or logical playout or presentation time. A track is not required to have samples at every sample time slot.

4.27 time line: A time line defines the playout or presentation time of video, audio, data, and time codes.

4.28 under construction: The collection of media files on a source device are considered to be under construction when a copy is transferred while the original is being recorded or being created by another concurrent file transfer operation.

4.29 unified material format: The unified material format (UMF) packets provide a description of the streamed material. The UMF includes the equivalent of an edit decision list, which allows a compound clip composed of several segments to be included in a single stream. This is not a complete SMPTE EDL (SMPTE 258M). The UMF also includes all user data associated with the material.

4.30 uniform media locator: A uniform media locator (UML) shall describe the location of the material to be streamed on a source or receiver device. UMLs are similar in concept to the universal resource locator (URL) scheme used with HTTP on the Internet.

4.31 user data: Information that the user chooses to associate with the material. This may include descriptive text, user-defined metadata, user-defined edit decision information, or any other information that the user chooses. User data is contained in the UMF packets.

Note: User data can be encoded using SMPTE metadata encapsulation techniques (SMPTE 336M encoding and SMPTE RP 210 dictionary).

4.32 vector: A single dimensional ordered set of items.

5 Stream Composition

Using streaming to transfer material allows the transferred material to be used at the receiver while the material is still being transferred from the source. Similarly, material may be transferred from the source while it is still being recorded. This is accomplished by packet multiplexing all of the material types.

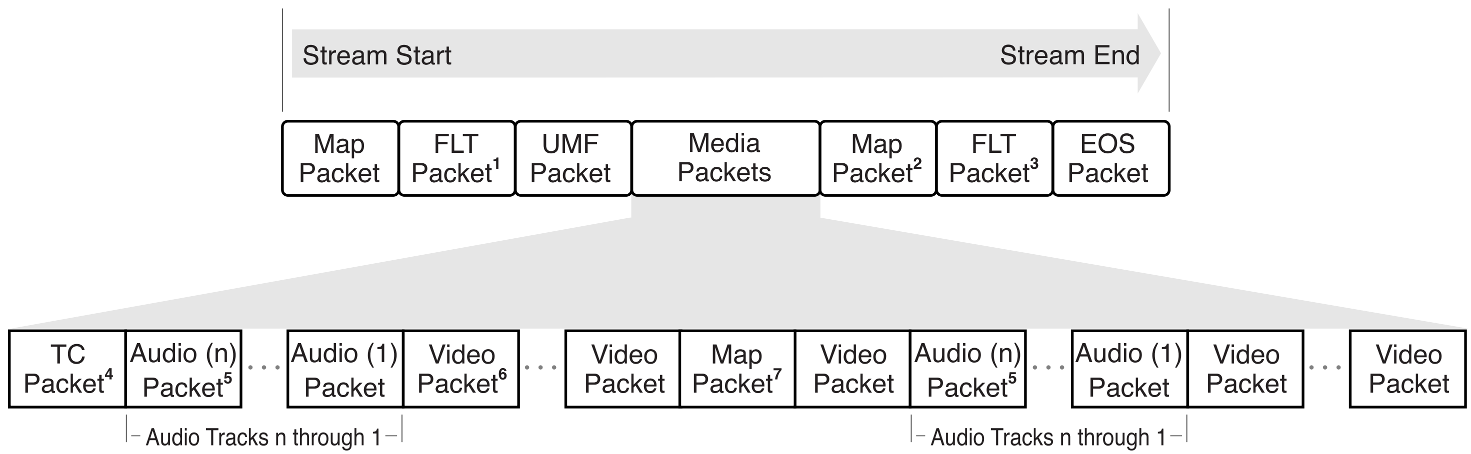
5.1 Overall Stream Sequence

All information in the stream is contained in packets. A packet shall consist of a header and a payload. Packets shall be a multiple of 4 bytes in length. Each packet shall start with a packet header. The header shall identify the information contained in the packet's payload and the payload length. The packet header shall include leader and trailer bytes.

Every stream shall start with a map packet, shall include at least one unified material format (UMF) packet, and end with an end of stream (EOS) packet. The material being streamed determines the type and position of any additional packets. The packet order for a stream is shown in Figure 3.

A field locator table (FLT) should be included in the stream. Receiving devices shall not depend on the presence of an FLT for correct operation. No more than one FLT shall be included in a stream. If present, the position of the FLT shall be determined by the status of the material (under construction or not) on the source system. Streams generated when the material is not under construction shall have the optional FLT placed immediately after the first map packet. If the material is under construction while the stream is being created, the optional FLT shall be immediately prior to the EOS packet.

Note: This Standard recommends all streams contain an FLT. However, in some cases such as an implementation of a partial retrieve from a data tape archive, construction of an FLT could be difficult and will add limited value. Therefore, this Standard does not require an FLT.



Notes:

1. FLT packet position if material is not under construction.
2. Extra map packet to update final field number if material is under construction.
3. FLT packet position if material is under construction.
4. TC packets contain 504 TC values.
5. Audio packets hold 32768 audio samples.
6. Video packets containing one field or one frame inserted until a TC or Audio packet is required.
7. Map packets inserted at least every 100 packets, or at any change of media segments.

Figure 3 – Packet order in stream

5.2 Sequence of MAP Packets and Active Material Media Packets

Media packets shall contain one field or frame of video, a vector of audio sample words, or a vector of time code values. The video, audio, and time code media packets shall be interleaved (time multiplexed) in the stream in a manner that allows the material to be played while the stream is being received. Video, audio, and time code packets shall only be inserted in the stream if they contain material.

All audio, and time code packets, including the first and last packets, shall include a complete set of sample locations. Audio, and time code packets may have unused (unrecorded) sample locations in the first packet (at the start of material on a track) or at the end of the last packet (at the end of material on a track).

The mapping between a sample's location in a packet and the sample's playout or presentation time is always maintained, even if part of the packet contains unrecorded samples.

Note: Additional information on audio and time code packet formatting, synchronization, and stream construction can be found in clause 7.4 and Annex B.

A source device may record time codes as media tracks. Recorded time code tracks may include invalid values and discontinuities in the time line they represent.

Time code packets, if present, shall be placed in the stream before audio packets. In turn, audio packets, if present, shall be placed in the stream before active video packets for the current field. This guarantees that all the time code, and audio information is available to the receiver device before the playable video field or frame. Media packets shall be inserted in the stream in a sequence such that all packets needed to play a given video field or frame are available by the time the track 0 (video) packet arrives.

A map packet shall be present in the stream at least every 100 packets. These recurring map packets allow playout of a stream "joined in progress". They also allow for the recovery of part of the material if earlier parts of the stream were unreadable.

If a simple clip is streamed and the material is not under construction, the additional map packets shall be duplicates of the first one in the stream. If a simple clip is streamed and the material is under construction, the additional map packets may differ from all the preceding map packets. Successive map packets should contain the correct (current) last field number.

A closing MAP packet shall be inserted after the last media packet if the stream being transferred was under construction. This MAP packet shall contain the correct last field number value.

Video, audio and time code packets that have no active material shall not be inserted into the stream. Partially recorded audio and time code packets shall be inserted in the stream when the material starts or stops at boundaries that are not packet boundaries.

Note: The recommended playout behavior is missing video will be presented as black, missing audio will be presented as silence, and missing time codes values will rot be present in the output stream. This is not a file/stream format policy choice and therefore this recommendation is not part of this Standard.

5.3 Decoder Pre-charge Media Packets

Some audio and video compression systems require additional information to pre-charge decoders or successfully complete the playout of a track at cuts or the start or end points. These packets shall be placed in timeline sequence for their respective track. Decoder pre-charge or completion packets may not be in the correct stream position in relationship to the other tracks media packets.

5.4 Sequence of Media Packets for Compound Clips

The active material in a compound clip shall be transferred as if it was a simple clip (see clause 5.2).

Additional map packets shall be included at least every 100 packets. In addition, new map packets shall be present when the media file used for any given track changes.

Notes:

1) The additional MAP packets allow a stream of compound clips to be thought of as a sequence of simple clips.

2) A long GOP MPEG video stream usually needs additional media packets at the transitions to precharge the playout decoders.

Inactive material packets (handles) shall be transferred after the active material. Media packets shall be sent in timeline order within each track. The sequencing of tracks for unseen material is not specified in this Standard.

5.5 Mixed Media Tracks in Compound Clips (Informative)

A detailed specification of source and receiver device implementation strategies for processing complex files is beyond the scope of this Standard. This section offers some guidance to source and receiver implementers. For example, the source and receiving devices must have appropriate audio, video, and time code codecs.

The source of compound clips should not generate streams with media type transitions on a track that the receiver cannot process or display. Transitions for field and frame based compression systems is relatively simple. Transitions for tracks containing material from temporal compression systems may require multiple decoders to correctly play the exact sequence of frames specified in the stream.

The source stream may need overlapping material at transitions. In a long GOP MPEG stream the stream must have “pre-charge” frames starting with a closed GOP I-frame. Including a sequence header and extension with each GOP will improve interoperability.

A stream that includes material from multiple compression families requires decoders for each compression family; overlapping material may be needed in the stream for temporal compression systems. In addition, decoder-to-decoder output stream switching may be required in the receiver.

MPEG material is represented as an MPEG Elementary Stream. This minimizes the difficulties normally associated with cutting MPEG material. When MPEG ES streams are converted to Transport Streams several additional constraints are placed on the stream. An example is the requirement to meet MPEG buffer fullness requirements.

6 Packet Headers

Each packet shall begin with a packet header as shown in Figure 4.

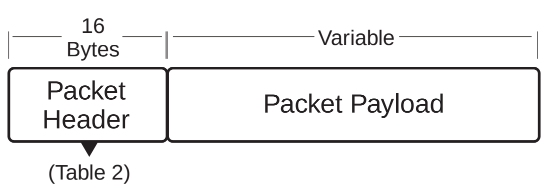


Figure 4 – Packet header

Each packet header shall consist of 16 bytes. A packet header shall contain a 5-byte leader and a 2-byte trailer as part of the packet header. The packet header shall include the packet type and the packet length. The specified packet length shall be in bytes and shall include the length of the packet header. The contents of a packet header shall be as shown in Table 2.

Table 2 – Packet header description

Offset	Value	Usage
0x00	0x00	Packet leader
0x01	0x00	
0x02	0x00	
0x03	0x00	
0x04	0x01	
0x05	Variable	Packet type: 0xBC = map 0xBF = media 0xFA = reserved 0xFB = end of stream 0xFC = field locator table 0xFD = UMF file 0xFE = reserved 0xFF = reserved Source devices should not generate reserved packet types. Receivers shall ignore packets with these packet types.
0x06	Variable	Packet length in bytes, including the 16-byte header. This shall be a 32-bit unsigned integer encoded most significant byte first.
0x0A	0x00	Reserved
0x0B	0x00	
0x0C	0x00	
0x0D	0x00	
0x0E	0xE1	Packet trailer
0x0F	0xE2	

7 Packet Type Descriptions

The following clauses describe the various packet types and their contents.

7.1 Map Packets

Map packets contain all the information about the subsequent media packets that is required to store the packets in appropriate media files and set their attributes. Map packets shall also include a current description of the streamed material. These values shall reflect the changing description over time if the material is under construction.

The map packet may be used by the receiver to identify material segments and track information for those segments.

The map packet shall include the type of each track in the material, the media file names, and other information described in the following clauses. Not all values are present in all map packets. Missing values shall default to 0 for numeric values and the null string (0x00) for text values. Certain map packet values may change within a stream. Examples are media file names (for compound clips) and the size of the stream (material streamed while being recorded).

Map packets do not contain a complete definition of the streamed material. The complete material definition shall be contained in the unified material format packet (see clause 7.3).

A map packet shall have two sections: a header and a payload (see Figure 5).

7.1.1 Map Packet Header

The map packet header shall take the form of a packet header described in clause 6 of this Standard.

Refer to Table 2 to identify the packet type code for a map packet.

7.1.2 Map Packet Payload

The map packet payload shall consist of three sections; a fixed length preamble, a variable-length material data section that contains update information common to the entire file, and a variable-length track description section that describes the material tracks.

A map packet may have fill bytes following the track descriptions. The packet header defines the length of the map packet payload. The size of the preamble is fixed and the sizes of the material data section and the track description section shall be given in their respective headers. The size of all payload components shall be less than or equal to the map packet payload length.

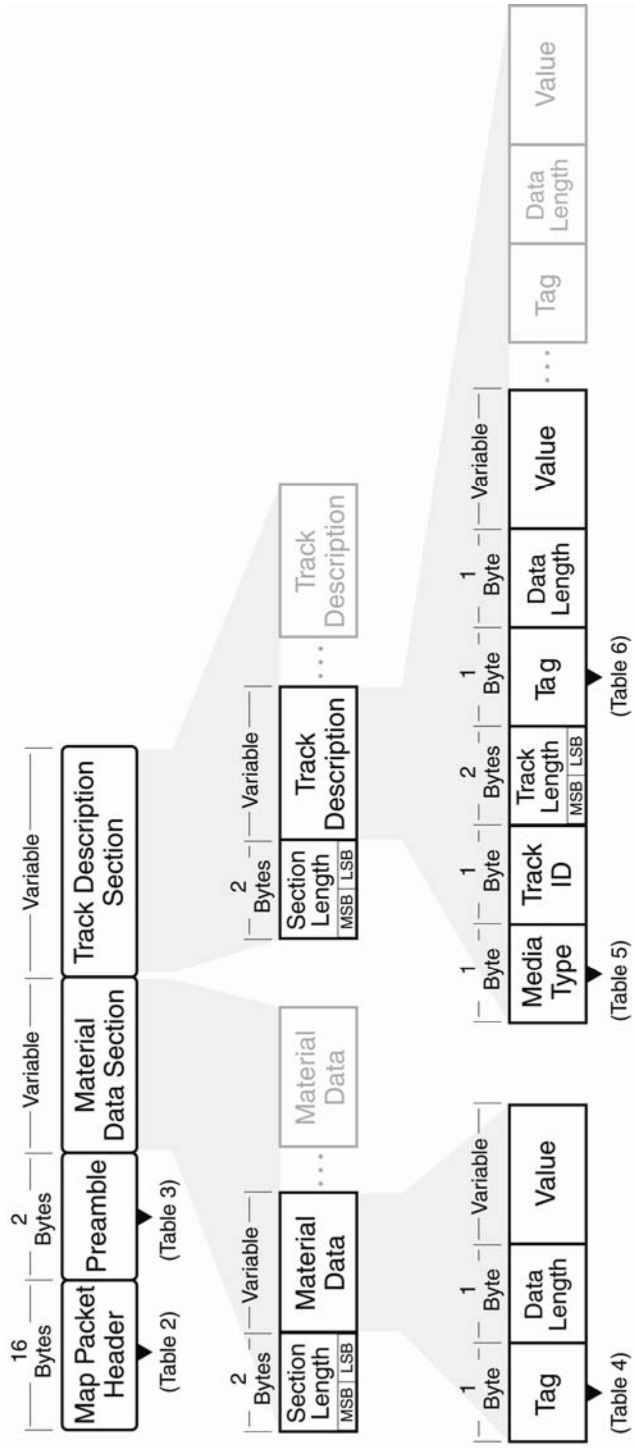


Figure 5 – Map packet composition

7.1.2.1 Map packet preamble

The map packet preamble shall have a fixed length and shall contain the two bytes shown in Table 3.

Table 3 – Map packet preamble

Offset	Contents	Description
0x00	0xE0	The low-order 5 bits of byte 0 shall be the version number of the stream format. This Standard describes version 0. The high order 3 bits shall be 1s.
0x01	0xFF	This byte is reserved for future use and shall be coded as a 0xFF by a source.

7.1.2.2 Material data section

The variable-length material data section shall have its byte length defined at the start of the section. The length value shall not include itself and is in units of bytes. The length value shall be a 16-bit unsigned integer with the most significant byte preceding the least significant byte.

The material data section contains a variable number of variable-length data values. Each data value contains one attribute of the stream. All data values shall be of the following format:

Tag	Length	Value Bytes

The tag value identifies the data. The tags and the associated types are listed in Table 4.

The length value contains the length of the value in bytes; it does not include itself or the tag value.

The tag and length are one byte each.

Value bytes shall be streamed with the most significant byte first, followed by progressively less significant bytes.

7.1.2.3 Tag values, type, length and usage

Table 4 describes the tags used to identify the material data values.

Table 4 – Tag descriptions

Tag	Data type	Description
0x40	STRING	Media file name of material
0x41	UINT32	First field of material in stream (inclusive)
0x42	UINT32	Last field of material in stream (exclusive)
0x43	UINT32	Mark in for the stream (inclusive)
0x44	UINT32	Mark out for the stream (exclusive)
0x45	UINT32	Estimated size of stream in 1024 byte units
0x46	UINT32	Reserved
0x47	UINT64	Reserved
0x48	STRING	Reserved
0x49	STRING	Reserved
0x4A	STRING	Reserved
0x4B	STRING	Reserved

Media file name: A string with the media file name from the source device.

First and last field: The numbers of the first field (inclusive) and last field (exclusive) of the material contained in the stream.

The field number is the time line position of the field. If a segment is being streamed, the first and last field values shall be the segment start field number (inclusive) and the segment end field number (exclusive). If the stream is being created while the material is being recorded, the last field value shall increase as the material is recorded. The last field value in the first map packet will be the last recorded field at the time the stream was started. Subsequent map packets shall have an updated last field value.

There shall be a map packet after the last media packet containing the final value for the last field.

Mark in and mark out: The mark in and mark out field numbers for this stream.

Estimated size: The space estimate shall be calculated by the source when the stream is started. The space estimate shall be specified in units of 1024 bytes. If the stream is being created while the material is still being recorded, the space estimate shall be 0.

7.1.3 Track Description Section

The track description section shall have its length in bytes defined at the start of the section. The track description section length value shall not include itself. The track description section length value shall be a 16-bit unsigned integer with the most significant byte preceding the least significant byte.

There shall be one entry in the track description section for each track in the material being streamed. The track description section contains the media file type, name, and auxiliary information.

Each track description shall have the following value items. There shall not be any padding between individual track description items or track descriptions.

Media type shall be the media type plus 0x80. Media types are listed in Table 5.

Media type values shall be a single byte.

Track ID shall be the track number plus 0xC0. The track number shall be in the range 0 to n - 1. Track number values shall be a single byte.

Track description value length shall be in bytes and shall not include itself, the type, or the ID byte. The track description value length shall be a 16 bit unsigned integer with the most significant byte preceding the least significant byte.

Track description value(s) shall have the same tag, length, and value format as described above for the material data section. The tags for the track description section are defined in Table 6.

Value bytes shall be streamed with the most significant byte first, followed by progressively less significant bytes.

Table 5 – Media types

Type number	Type name	Description
3	Motion JPEG 525	A video track encoded using JPEG (ITU-R T.81 or ISO/IEC 10918-1) for 525 line material (see clause 7.4.2.4).
4	Motion JPEG 625	A video track encoded using JPEG (ITU-R T.81 or ISO/IEC 10918-1) for 625 line material (see clause 7.4.2.4).
7	Time code 525	SMPTE 12M-1 time code tracks (see clause 7.1.3.3 and clause 7.4.2.2).
8	Time code 625	SMPTE 12M-1 time code tracks (see clause 7.1.3.3 and clause 7.4.2.2).
9	Audio PCM 24	A mono 24-bit PCM audio track (see clause 7.4.2.3.2).
10	Audio PCM 16	A mono 16-bit PCM audio track (see clause 7.4.2.3.1).
11	MPEG-2 525	A 525 video track encoded using ISO/IEC 13818-2 (MPEG-2. See clause 7.4.2.5).
12	MPEG-2 625	A 625 video track encoded using ISO/IEC 13818-2 (MPEG-2. See clause 7.4.2.5).
13	DV-based 25 Mb/s structure, 525	A video track encoded using SMPTE 314M or ISO/IEC 61834-2 DV encoded at 25 Mb/s for 525/60i (see clause 7.4.2.8).
14	DV-based 25 Mb/s structure, 625	A video track encoded using SMPTE 314M or ISO/IEC 61834-2 DV encoding at 25 Mb/s for 625/50i (see clause 7.4.2.8).
15	DV-based 50 Mb/s structure, 525	A video track encoded using SMPTE 314M DV encoding at 50 Mb/s for 525/50i (see clause 7.4.2.8).
16	DV-based 50 Mb/s structure, 625	A video track encoded using SMPTE 314M DV encoding at 50 Mb/s for 625/50i (see clause 7.4.2.8).
17	AC-3 16-bit audio	An AC-3 audio track (see clause 7.4.2.3.3 for details).
18	Compressed 24-bit audio	A non-PCM AES data track (see clause 7.4.2.3.4 for details).
19	Reserved	A receiver shall ignore this media type.
20	MPEG-2 HD	A video track encoded using ISO/IEC 13818-2 (MPEG-2) main profile at main level or high level, or 4:2:2 profile at main level or high level (see clause 7.4.2.7).
21	Reserved	A receiver shall ignore this media type.
22	MPEG-1 525	A 525 video track encoded using ISO/IEC 11172-2 (MPEG-1) (see clause 7.4.2.6).
23	MPEG-1 625	A 625 video track encoded using ISO/IEC 11172-2 (MPEG-1) (see clause 7.4.2.6).
24	Time Code HD	SMPTE 12M-1 time codes for HD material (see clause 7.1.3.3 and clause 7.4.2.2).

Table 6 – Track description tags

Tag	Data type	Length	Usage
0x4C	STRING	Var	Media file name
0x4D	BINARY	8	Auxiliary Information. The exact meaning depends on the track type.
0x4E	UINT32	4	Media file system version. Receivers may ignore this value.
0x4F	STRING	Var	MPEG video auxiliary information
0x50	INT32	4	Frame rate 1 = 60 frames/sec 2 = 59.94 frames/sec (the exact rate is 60/1.001) 3 = 50 frames/sec 4 = 30 frames/sec 5 = 29.97 frames/sec (the exact rate is 30/1.001) 6 = 25 frames/sec 7 = 24 frames/sec 8 = 23.98 frames/sec (the exact rate is 24/1.001) -1 = Not applicable for this track type -2 = Not available
0x51	INT32	4	Lines per frame 1 = 525 2 = 625 4 = 1080 5 = Reserved 6 = 720 -1 = Not applicable -2 = Not available
0x52	INT32	4	Fields per frame 1 = Progressive 2 = Interlaced -1 = Not applicable -2 = Not available

The media file name shall be a string with the name of the media file on the source device that contains the material. If the source of the media packets changes, as occurs when streaming a compound clip, a map packet containing the new source media file name shall be inserted in the stream. A receiver may use the media file name or may use a different media file name or identifier.

A null media file name shall have a 0x00 (NULL) as its first character. The null media file name shall be used for segments that have not been recorded or otherwise provided. For tracks with null media file names, auxiliary information and media file system version information values should not be sent by a source and shall be ignored by a receiver.

Auxiliary information shall be associated with each media file. The auxiliary information contains attributes of the material. The format of the auxiliary information is dependent on the media file type. For MPEG video media files, only auxiliary information tagged 0x4F shall be included. The MPEG video auxiliary information shall be encoded as a string described in Table 7. For other media file types, only auxiliary information tagged

0x4D shall be included, and it shall be encoded as an 8-byte binary structure. Only one form of auxiliary information, either 0x4D or 0x4F, shall be valid for any one track.

7.1.3.1 MPEG video auxiliary information

MPEG video auxiliary information shall be encoded as a variable-length 0x00 (NULL) terminated string. The string shall contain 12 parameters. Each parameter shall consist of an identifier, followed by a space character (0x20), followed by the value, followed by a new line character (0x0A). The parameters are described in Table 7. All parameter values except Br shall be the string representations of decimal numbers. Br shall be an IEEE 754 64-bit floating-point number.

Table 7 – MPEG video auxiliary information

Identifier	Description
Ver	Version of this MPEG video auxiliary information. This value shall be 1.
Br	The data rate of the MPEG video stream expressed as bits per second.
Ipg	The number of I-pictures per group of pictures (GOP).
Ppi	The number of P-pictures per I-picture.
Bpiop	The number of B-pictures per I-picture or P-picture.
Pix	Picture structure: 1 for field picture structure 0 for frame picture structure.
Cf	Color difference format 0 = Reserved. 1 = 4:2:0 2 = 4:2:2
Cg	Closed GOP: 0 = open GOP 1 = closed GOP
Sl	Starting line number in a field of the recorded picture. Shall be in the range of 1 – 31 inclusive.
nl16	Number of encoded lines in the picture divided by 16.
Vi	Set to 1 if this structure has valid data.
f1	1 = field 1 is interlaced above field 2, or if Pix is 0 (frame based); 0 = field 1 is interlaced below field 2.
cbpg	Constant bytes per group indicator: 0 = variable bytes per MPEG GOP 1 = constant bytes per MPEG GOP If this identifier is not detected by the receiver, the receiver shall assume variable bytes per MPEG GOP. Note: See SMPTE 356 for encoding recommendations.
ar	Aspect Ratio: 0 = 4x3 1 = 16x9 If this identifier is not detected by the receiver, the receiver shall assume a 4x3 aspect ratio.

The format of the binary auxiliary information (0x4D) for Motion JPEG and audio packets is 8 bytes of reserved data. The receiver shall ignore data in these bytes.

7.1.3.2 DV-based auxiliary information

The format of the binary auxiliary information (0x4D) for DV-based packets shall be implemented as defined in Table 8 and Table 9.

Note: Earlier versions of SMPTE 360M defined binary auxiliary information for DV-based packets to be 8 bytes of reserved data. These bytes can now be discerned by checking the values of the auxiliary information validity bits as defined in Table 8 and Table 9.

Table 8 – DV-based 25Mb/s auxiliary information

Bits (0:LSB, 63:MSB)	Contents
0:3	DV Format: 0 = DVCPRO 25 Mbps 1 = DVCAM Other values are reserved
4:4	Aspect Ratio: 0 = 4x3 1 = 16x9 The other value are reserved
5:29	Reserved
30:31	Auxiliary information validity: 0 = DV auxiliary information invalid 1 = DV auxiliary information valid The other values are reserved
32:63	Reserved

Table 9 – DV-based 50Mb/s auxiliary information

Bits (0:LSB, 63:MSB)	Contents
0:0	Aspect Ratio: 0 = 4x3 1 = 16x9
1:29	Reserved
30:31	Auxiliary information validity: 0 = DV auxiliary information invalid 1 = DV auxiliary information valid The other values are reserved
32:63	Reserved

7.1.3.3 Time code auxiliary information

Time code values can be recorded in time code packets representing a stream of captured time code values. A continuous stream of time codes can also be represented as a starting value and successive values which are calculate using the starting value and field numbers. These “striped time code” values are not stored as a stream of time code media packets. The initial time code value is stored in the time code auxiliary information with a flag to identify the time code representation (recorded or striped.)

The format of the binary auxiliary information (0x4D) for time code packets shall be defined by Table 10. See SMPTE 12M-1 for additional information.

The invalid bit (bit 7 of byte 3) shall be set to 1 for time code tracks with recorded time code values which are stored in a stream of time code media packets. The remaining bits in the time code auxiliary information are reserved. These bits should be set to zero by sources and they shall be ignored be receivers.

The invalid bit (bit 7 of byte 3) shall be set to 0 for striped time codes. The starting time code value (for field number 0) shall be stored in the time code auxiliary information field defined in Table 10.

Note: During playout receivers are expected to use the initial time code value (stored in the auxiliary time code information) and a given field number to calculate the appropriate time code to be inserted in the output stream.

Table 10 – Time code track auxiliary information

Offset	Contents
0x00	Fields – This value shall be calculated as follows: (“Tens of frames” BCD time code value * 20) + (“Units of frames” BCD time code value * 2).
0x01	Seconds – This value shall be calculated as follows: (“Tens of seconds” BCD time code value * 10) + (“Units of seconds” BCD time code value).
0x02	Minutes – This value shall be calculated as follows: (“Tens of minutes” BCD time code value * 10) + (“Units of minutes” BCD time code value).
0x03	Hours (bits 0-4) – This value shall be calculated as follows: (“Tens of hours” BCD time code value excluding the Drop frame, Color frame, and Invalid bits * 10) + (“Units of hours” BCD time code value). Drop frame flag (bit 5) Color frame flag (bit 6) Invalid flag (bit 7)
0x04	User bits 1 - bits 0-3 User bits 2 - bits 4-7
0x05	User bits 3 - bits 0-3 User bits 4 - bits 4-7
0x06	User bits 5 - bits 0-3 User bits 6 - bits 4-7
0x07	User bits 7 - bits 0-3 User bits 8 - bits 4-7

7.2 Field Locator Table Packet

The field locator table (FLT) packet consists of a header and a payload as shown in Figure 6.

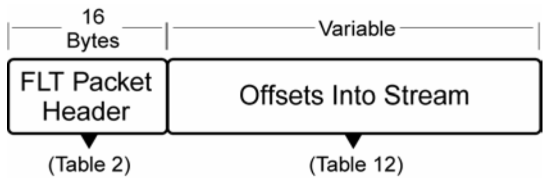


Figure 6 – Field locater table packet

7.2.1 Field Locator Table Packet Header

The field locator table packet header shall take the form of a packet header described in clause 6 of this Standard.

Refer to Table 2 to identify the packet type code for a field locator table packet.

7.2.2 Field Locator Table Packet Payload

A device may use the FLT to locate the starting point of a segment to be extracted from a stream.

The FLT payload shall start with two unsigned 32-bit integers: the number of fields per FLT entry and the number of valid entries in the FLT. These shall be followed at least 1 and not more than 1000 offsets into the stream. The number of offsets shall be the smallest of 1000, the number of fields of video. The offsets shall be in units of 1024 bytes. A field number that describes the segment starting point divided by the number of fields per entry shall be used as an index into the field locator table offset vector. The offset to the first active field in the stream shall not be stored in the FLT offset vector. The FLT offset vector entry shall be the offset from the start of the stream to a point before the desired field. The location may not be aligned with the start of a packet; in this case the receiver shall search for a packet header containing the field number for the desired field (see Table 11).

Value bytes shall be streamed with the least significant byte first, followed by progressively more significant bytes.

Table 11 – FLT packet payload description

Offset	Data type	Description
0x00	UINT32	Number of fields per FLT entry Number of fields per FLT entry shall be a 32-bit unsigned integer that contains the number of fields represented by each FLT entry. This value shall be defined by the following formula: ((last_field - first_field) / maximum_number_of_entries) + 1. The division shall be an integer (truncating) division. The value last_field shall be one greater than the actual last active field number. The value first_field is the first active field number. The maximum_number_of_entries is 1000.
0x04	UINT32	Number of FLT entries Number of valid FLT entries shall be a 32-bit unsigned integer that contains the number of active FLT offsets. The offset to the first active field is not stored in the FLT. The number of valid FLT entries shall be defined as: ((last_field - first_field) / number_of_fields_per_FLT_entry) – 1. Number_of_fields_per_FLT_entry is defined in “Number of fields per FLT entry” (see above).
0x08	UINT32	A vector of UINT32 offsets to fields Offsets to fields shall be a vector of 32-bit unsigned integers used to store offsets to the approximate locations of fields in a stream. The offset to the first active field in the stream shall not be stored in this vector. The offset for the nth entry (where n varies from 1 to 1000) represents the field found by the following formula: (n * number_of_fields_per_FLT_entry) + first_field. The offset to a field shall be the byte offset into the stream divided by 1024

Notes:

1) Video material is always addressed as fields, not frames. For progressive formats, odd field numbers are not meaningful.

2) To locate a field in a stream the following procedure is used. Let N be the desired field number. FLT index = (N – first_field) / number_of_FLT_entries. If FLT index is 0, offset to field is 0 (there is no FLT entry.) Otherwise use the FLT entry (FLT index – 1).

3) For video material that is compressed as frames (e.g. DV and MPEG) the FLT can have entries pointing to the second field in a frame. For example; a FLT representing a file with less than 500 frames of DV video will have 500 sets of pointers. The first pointer in the Nth pair points to field 1 of frame N and the second points to field 2 of frame N. In fact both pointers represent frame N so the information is redundant. For files with 501 or more frames of video, the pointers in a FLT can logically point to the second field in a frame.

4) It is important to recognize that the purpose of the FLT is to support partial file restore (retrieve). The FLT is not intended to be used as a file on-disk frame lookup table.

7.3 Unified Material Format Packets

The unified material format (UMF) packet contains a complete description of the material and includes user data. The user data may include any type of information. This allows information that is associated with the material but not part of the material's media files to be packaged into a single byte stream for transport (see Figure 7).

Some properties in the UMF are also in MAP packets. These values should be the same. The exceptions are values that are changing in a file that is being transferred while under construction. In this case, the values in MAP packets shall have priority over values in the UMF.

7.3.1 Unified Material Format Packet Header

The unified material format packet header shall take the form of a packet header described in clause 6 of this Standard.

Refer to Table 2 to identify the packet type code for a unified material format packet.

7.3.2 Unified Material Format Packet Payload

The unified material format payload is composed of several sections. The sections are listed here in the order in which they shall appear in the packet. Descriptions of the sections are as follows:

- Preamble
- Payload description
- Material description
- Track description(s)
- Media descriptions(s)
- User data

All UMF packets shall have a preamble. If the UMF payload exceeds the 32,768-byte length limit, the payload shall be split across several consecutive UMF packets. The requirement for each and every UMF packet preamble is independent of the number of UMF packets required to transport the payload.

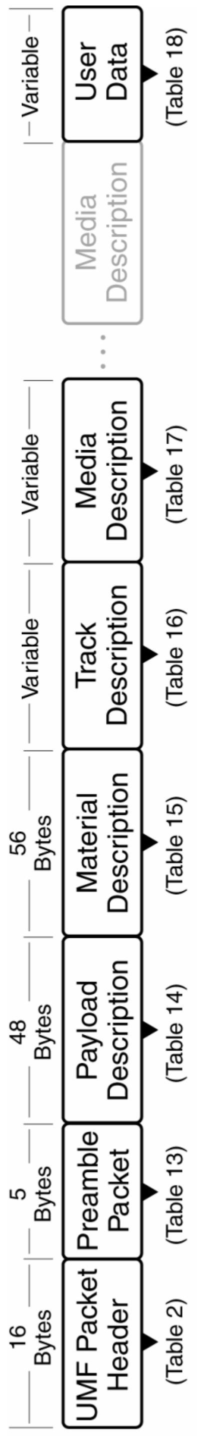


Figure 7 – UMF packet composition

7.3.2.1 Unified material format packet preamble

Unified material format packets shall contain a preamble immediately following the packet header. The UMF payload may be split across several consecutive packets. In this case, each UMF payload packet shall contain a preamble section. The contents of the preamble shall be implemented as described in Table 12.

Table 12 – UMF packet preamble

Offset	Contents
0x00	First/last packet flag bits 0 = Not first or last (an intermediate) packet 1 = First packet 2 = Last packet 3 = First and Last (only) packet
0x01	Payload data length (MSB)
0x02	Payload data length
0x03	Payload data length
0x04	Payload data length (LSB)

First/last packet flag: UMF packet payload shall not exceed 32,768 bytes in length. If the UMF data requires less than or equal to 32,768 bytes, it shall be a single packet with a preamble section and the First packet flag and Last packet flags set. If the UMF payload data requires more than 32,768 bytes, it shall be split across several packets. The first/last UMF packet flag shall identify the first, last, and intermediate UMF packets. Each packet will contain its own preamble section, and then continue the UMF payload where the previous packet ended.

All UMF packets shall be sent consecutively. No other packet types shall be inserted between the first and last UMF packets.

Payload Data Length: The payload data length of the preamble defines the number of valid UMF payload bytes in this packet.

Note: This value is not the same as the UMF packet length that is contained in the UMF packet header and it does not account for the bytes that form the UMF preamble.

If the UMF payload data are not a multiple of four bytes, the last packet of UMF payload shall be padded to a multiple of four bytes. The contents of the padding bytes shall be undefined.

7.3.2.2 Unified material format payload description

The payload description shall describe the size and location of the various sections of the UMF packet (see Table 13), Value bytes shall be streamed with the least significant byte first, followed by progressively more significant bytes.

All offsets used in Table 13 shall be in bytes and shall be the offset from the origin of the payload description section to the defined section. These offsets shall not include UMF preamble value bytes.

Table 13 – UMF payload description

Offset	Data type	Description
0x00	UINT32	Total length of the UMF data (bytes)
0x04	UINT32	Version of this file. This value shall be three (3)
0x08	UINT32	Number of tracks in the material
0x0C	UINT32	Offset in bytes from start of payload description section to track description section of the UMF (bytes)
0x10	UINT32	Size of the track description section (bytes)
0x14	UINT32	Number of segments referenced by the UMF
0x18	UINT32	Offset from start of payload description section to media description section of the UMF (bytes)
0x1C	UINT32	Size of the media description section (bytes)
0x20	UINT32	Offset from start of payload description section to the user data section of the UMF (bytes)
0x24	UINT32	Size of the user data section (bytes)
0x28	UINT32	Reserved
0x2C	UINT32	Reserved

7.3.2.3 Unified material format material description

The UMF material description section shall provide information about the material (see Table 14). Value bytes shall be streamed with the least significant byte first, followed by progressively more significant bytes.

Table 14 – UMF material description

Offset	Data type	Description	
0x00	UINT32	**Material attributes, coded as a bit mask.**	
		Bit mask	**Description**
		0x00000001	Material is read only. This bit and the locked bit should be treated as equivalent.
		0x00000002	Reserved
		0x00000004	Material is locked. This bit and the read only bit should be treated as equivalent.
		0x00000008	Reserved
		0x00000010	Reserved
		0x00000020	Reserved
		0x00000040	Video sample rate is 50 Hz
		0x00000080	Video sample rate is 59.94 Hz (the exact rate is 60/1.001)
		0x00000100	Video sample rate is 24 or 23.98Hz (the exact rate is 24/1.001)
		0x00000200	Video sample rate is 25 Hz
		0x00000400	Video sample rate is 30 or 29.97 Hz (the exact rate is 30/1.001)
		0x00000800	Material contains MPEG-2 high-definition video
		0x00001000	Material contains DV-based video in a 25 Mb/s structure
		0x00002000	Material contains DV-based video in a 50 Mb/s structure
		0x00004000	Material contains Motion JPEG video
		0x00008000	Material contains MPEG-2 standard definition video
		0x00010000	Material is open
		0x00020000	Material is open and shared
		0x00040000	Material is open exclusively
		0x00080000	Material is a simple clip
		0x00100000	Reserved
		0x00200000	First time code track is non-drop frame format
		0x00400000	First time code track is drop-frame format
		0x00800000	Under construction by streaming
		0x01000000	Reserved
		0x02000000	Reserved
		0x04000000	Audio is 16-bit PCM.
		0x08000000	Audio is 24-bit PCM.

		0x10000000	Reserved
		0x20000000	Under construction by recording
		0x40000000	Under construction by being copied from other material
		0x80000000	Under construction by being restored from an archive
0x04	UINT32	Maximum length of the material in fields The maximum length of the material shall be the length of the longest track in the material. The minimum length of the material shall be the length of the shortest track (with material) in the material. These may be unequal since individual tracks can contain different amounts of material.	
0x08	UINT32	Minimum length of the material in fields	
0x0C	UINT32	Material mark in value in fields	
0x10	UINT32	Material mark out value in fields	
0x14	UINT32	Time code at mark in value The format of the time code shall be a UINT32 and shall be the same as the first 4 bytes of the time code auxiliary information in the MAP packet (see Table 10).	
0x18	UINT32	Time code at mark out value.	
0x1C	UINT32	Most significant 32 bits of last modified time, as a count of 100 ns intervals since Jan 1, 1601 Note – Most operating systems provide support for this time/date format.	
0x20	UINT32	Least significant 32 bits of the last modified time, as a count of 100 ns intervals since Jan 1, 1601	
0x24	UINT32	Most significant 32 bits of creation time, as a count of 100 ns intervals since Jan 1, 1601	
0x28	UINT32	Least significant 32 bits of the creation time, as a count of 100 ns intervals since Jan 1, 1601	
0x2C	UINT16	Reserved	
0x2E	UINT16	Reserved	
0x30	UINT16	Number of audio tracks	
0x32	UINT16	Number of time code tracks	
0x34	UINT16	Reserved	
0x36	UINT16	Number of MPEG-1, MPEG-2 SD, and MPEG-2 HD video tracks	

If a clip is being recorded when a transfer starts, the under construction by recording bit shall be set. When set this bit indicates the following clip properties could be changing as a result of the concurrent recording and transfer operations:

- Maximum length of material in fields (see Table 14)
- Minimum length of material in fields (see Table 14)
- Material mark out value in fields (see Table 14)
- Time code at mark in value (see Table 14)
- Time code at mark out value (see Table 14)
- Number of fields in segment (see Table 16)

- Mark out value for the media file in fields (see Table 16)
- MPEG specifications
- GOP structure
- Target P-pictures per GOP
- Target B-pictures per P-picture or I-picture

The last MAP packet transferred shall indicate all final clip properties.

The under construction by recording bit shall be set in the UMF packet for the clip that is being recorded or received when an outbound file transfer operations starts.

7.3.2.4 Unified material format track description

The UMF track description section shall describe the track and the number of media files referenced by each track. There shall be as many consecutive track descriptors as are specified by the number of tracks in the payload description section (see Table 15).

Value bytes shall be streamed with the least significant byte first, followed by progressively more significant bytes.

Table 15 – UMF track description

Offset	Data type	Description
0x00	UINT16	Track information shall be two CHARACTERS in the form Xx, where X shall be the track type, and x shall be the track logical number within track type. Track types are defined as: A = Audio D = DV-based 25 Mb/s E = DV-based 50 Mb/s F = Reserved – Receivers shall ignore this track type H = MPEG-2 HD video L = MPEG-1 video M = MPEG-2 SD video T = Time code V = Motion JPEG The track number shall be the character encoding of the number; track 0 is 0x30, 1 is 0x31, etc. The audio track logical numbers for the first ten audio tracks shall be 0 to 9 and the next 22 tracks shall be A to V. For example, the first DV-based 25 Mb/s track would be D0, encoded as 0x4430 and the first audio track would be A0, encoded as 0x4130. The eleventh audio track would be AA, encoded as 0x4141.
0x02	UINT16	Number of segments on this track

7.3.2.5 Unified material format media description

There shall be as many consecutive media descriptions as specified by the number of segments referenced by the UMF (see Table 13) in the payload description section (see Table 16). Value bytes shall be streamed with the least significant byte first, followed by progressively more significant bytes.

Table 16 – UMF media description

Data type	Description
UINT16	Length of this media description
UINT16	Track information; same as in track description above
UINT16	Media Sequence number - For simple clips, this value is reserved. For compound clips, this value shall start with 1 and increment for each media description entry.
UINT16	Reserved
UINT32	Number of fields in segment
UINT32	Reserved
UINT32	Mark in value for the media file in fields
UINT32	Mark out value for the media file in fields
STRING	Source device media file name. This string shall be a maximum 88 characters including the 0x00 (NULL) termination. This value shall be coded as 88 bytes with unused bytes set to 0.
UINT32	Type of media track: 1 = Motion JPEG video 2 = Audio 3 = Time code 4 = MPEG-2 SD video 5 = DV-based 25 Mb/s video 6 = DV-based 50 Mb/s video 7 = MPEG-2 HD video 8 = Reserved 9 = MPEG-1 video
UINT32	Sampling rates for this track. Audio sampling is 48,000 samples / sec. Video and the associated time code sampling rates. 60 = 525I (representing 60 / 1.001 fields per sec) 50 = 625I (representing 50 fields / sec) 25 = 1080I (representing 50 fields / sec) 30 = 1080I (representing 60 or 60 / 1.001 fields per sec) 24 = 1080P (representing 24 or 24 / 1.001 frames per sec) 50 = 720P (representing 50 frames per sec) 60 = 720P (representing 60 or 60 / 1.001 frames per sec)
UINT32	Size of sample for audio and time codes, data rate for video as follows: JPEG and MPEG video = Approximate data rate in bits per second DV-based 25 = 25,000,000 bits per second DV-based 50 = 50,000,000 bits per second Audio: 16 bit PCM = 16 bit sample words (see clause 7.4.2.3.1) Audio: 24 bit PCM = 24 bit sample words (see clause 7.4.2.3.2)

	Audio: AC-3 = 16 bit sample words (see clause 7.4.2.3.3) Audio: non-PCM audio data = 24 bit sample words (see clause 7.4.2.3.4) Time code = 64 bits per time code value (see SMPTE 12M-1) Receivers shall not reject streams with extraneous values.
UINT32	Reserved
Media specifications The following 32 bytes shall contain an appropriate media specification. The order of the media descriptions in the UMF packet(s) shall be the same order as the segments on the track. Each track's media descriptions shall be together, and in the order the segments shall be played on that track.	
MPEG video specifications	
UINT32	Color difference format: 1 = 4:2:0 2 = 4:2:2
UINT32	GOP structure: 0 = Open end 1 = Closed end 0xFF = Unknown
UINT32	Frame/field structure: 0 = Unknown 1 = Top field 2 = Bottom field 3 = Frame
UINT32	Target I-pictures per GOP (see the note following this table)
UINT32	Target P-pictures per I-picture (see the note following this table)
UINT32	Target B-pictures per P-picture or I-picture (see the note following this table)
UINT32	MPEG video attributes: 0 = Unknown 1 = MPEG-1 video 2 = MPEG-2 video 4 = MPEG-2 video constant bytes per group (Note: See SMPTE 356 for encoding recommendations.) These values may be bitwise OR-ed with one of the following values indicating the aspect ratio; 0x40 = 4x3 aspect ratio 0x80 = 16x9 aspect ratio In the absence of these values a 4x3 aspect ratio shall be assumed.
UINT32	Reserved
Time code specifications	
UINT32	Time code attributes 0 = Drop-frame 1 = Non-drop frame 0xFF = Unknown
UINT32	Reserved
UINT32	Reserved

UINT32	Reserved
UINT32	Reserved
UINT32	Reserved
UINT32	Reserved
UINT32	Reserved
Audio specifications	
FLOAT64	Level at which to play this segment (0.0 to 8.0). A value of 1.0 represents unity gain. Note: A receiver can use this value as a multiplier for the PCM audio samples during playout. Playout device policy is not formally part of this standard so this is not a normative requirement.
FLOAT64	Level at which to terminate this segment (0.0 to 8.0). A value of 1.0 represents unity gain. Note: A receiver can use this value as a multiplier for the PCM audio samples during playout. This is the final multiplier for the audio ramp at the end of an audio segment and the initial multiplier for the audio ramp at the beginning of an audio segment. Playout device policy is not formally part of this standard so this is not a normative requirement.
UINT32	Number of fields over which to ramp up to the play level from current level.
UINT32	Number of fields over which to ramp down to the terminate level.
UINT32	Reserved
UINT32	Reserved
DV-based specifications	
UINT32	DV-based attributes: The following DV-based attributed may be bitwise OR-ed Aspect Ratio: 0x40 = 4x3 aspect ratio 0x80 = 16x9 aspect ratio In the absence of these values, a 4x3 aspect ratio shall be assumed. Format: 0x20 = DVCAM In the absence of the DVCAM indicator, the media shall be assumed to be DVCPRO
UINT32	Reserved
UINT32	Reserved
UINT32	Reserved
UINT32	Reserved
UINT32	Reserved
UINT32	Reserved
UINT32	Reserved

Motion JPEG specification	
FLOAT64	Maximum color difference quantification level
FLOAT64	Maximum luma quantification level
FLOAT64	Minimum color difference quantification level
FLOAT64	Minimum luma quantification level

Note on MPEG video GOP structure target values: The parameters (I-pictures per GOP, P-pictures per I-picture, and B-pictures per I-picture) are target values provided to the MPEG video encoder. The result stream will not always meet this cadence at the discretion of the encoder. These values are provided for information purposes only.

7.3.2.6 Unified material format user data

The unified media format packet may contain optional user data items, as specified in Table 17. The binary user data shall consist of any data of any type including user-defined metadata. A UMF shall contain zero or more user data items as described in Table 17. The number of user data items shall be determined by the number of bytes specified for user data in the UMF. Value bytes for user data item values shall be streamed with the least significant byte first, followed by progressively more significant bytes with the exception of the actual binary user data. The data format and byte sequence for the encapsulated user data is not specified in this Standard.

Table 17 – UMF user data items

Offset	Data type	Description
0x00	UINT32	The length of this user data record shall be in bytes and shall include all bytes from the user data record origin up to and including the string termination (0x00; null) byte and a reserved byte at the end of the record. The next user data record shall be located by adding this length to this user data records origin.
0x04	UINT32	Position on the material time line (expressed as a field number) associated with this user data record. This value should be 0xFFFFFFFF if the user data record is not associated with a segment. Note: A value of 0 was used in early implementations when a user data record was not associated with a segment.
0x08	UINT16	Track associated with the user data record; 0 means no specific track association; otherwise, track type and logical number (see Table 15). The track association cannot be 0 if the user data record is associated with a specific segment.
0x0A	UINT16	This value shall be 0 if this user data record is not associated with a track or media segment. In this case the media position and track association values above should not be referenced. If this user data record is associated with a track or media segment, this value shall be the Media Sequence Number (see Table 16.)
0x0C	UINT32	User-defined key for the user data. This key shall be used at the application implementer's discretion. This Standard does not specify the intended use for the user-defined key.
0x10	BINARY	First byte of the user data record; the record length is the size specified in the first value – 17 (the user data record length minus the length of the header and trailing string termination byte).
—	BINARY	The user data record shall end with a 0x00 (NULL) byte and a reserved byte. A source should encode the second byte with a 0x00 value and a receiver shall ignore the value in the reserved byte.

7.4 Media Packets

Media packets contain the video, audio, or time code material for the stream. Each media packet shall consist of a header and a payload as shown in Figure 8.

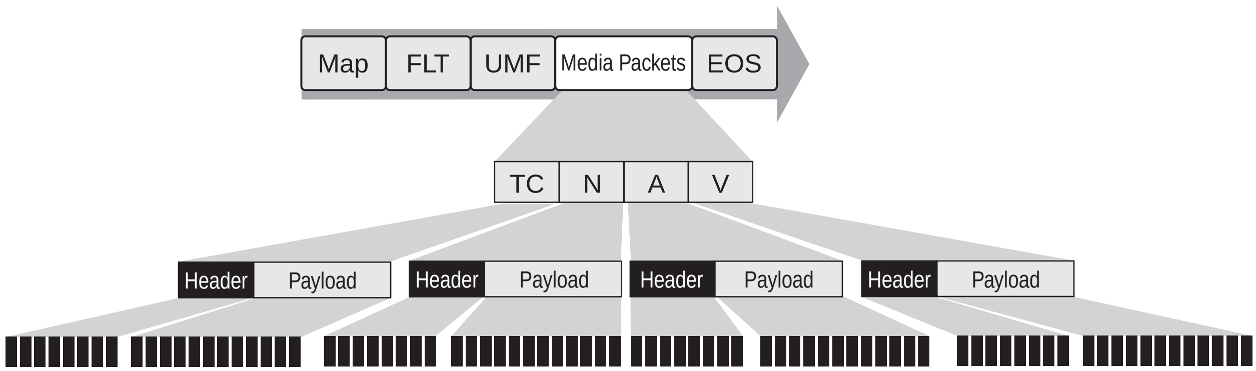


Figure 8 – Media packets

7.4.1 Media Packet Header

The media packet header shall take the form of a packet header described in clause 6 of this Standard. Refer to Table 2 to identify the packet type code for a media packet.

7.4.2 Media Packet Payload

Each media packet payload shall contain the following elements. They are described in the subsequent paragraphs.

- Media preamble
- Media data

7.4.2.1 Media preamble

Each media packet shall have a preamble following the stream packet header. This media preamble shall contain details of the media contained in the packet. The media preamble shall always be 16 bytes in length (see Table 18). Value bytes shall be streamed with the most significant byte first, followed by progressively less significant bytes.

Table 18 – Media preamble

Offset	Contents
0x00	Media type (see clause 7.4.2.1.1)
0x01	Track number (see clause 7.4.2.1.2)
0x02	Media field number (see clause 7.4.2.1.3)
0x06	Field information, byte 0 (see clause 7.4.2.1.4)
0x07	Field information, byte 1
0x08	Field information, byte 2
0x09	Field information, byte 3
0x0A	Time line field number (see clause 7.4.2.1.5)
0x0E	Flags (see clause 7.4.2.1.6)
0x0F	Reserved

7.4.2.1.1 Media types

The media type shall be represented in a byte. Table 5 lists the media types that may be inserted as the payload of media packets in the stream. A receiver shall ignore reserved media file types.

7.4.2.1.2 Track number

The track number is a reference into the track description section of the map packets. It identifies the media file to which the current media packet belongs. Track descriptions shall be considered as consecutive elements of a vector and track numbers are the index into that vector.

Clips shall have 1 to 48 tracks in any combination of types. Tracks shall be numbered from 0 to n–1. Media packets shall be inserted in the stream starting with track n-1 and proceeding to track 0. Time code packets shall have the greatest track numbers. Audio packets shall have track numbers less than time code packets. Video packet track numbers shall be less than audio track numbers.

Note: In a simple clip each track is a single segment. In a compound clip a track is a set of segments played back to back. The number of tracks is constant throughout the playout. This Standard does not specify a maximum number of segments.

7.4.2.1.3 Media field number

The media field number is a 32-bit unsigned number representing the media field location on a virtual time line within the current media file. In all cases the virtual time line is measured in units of video fields with the origin of the time line at field zero.

For video media, the field number shall be the number of a single field or frame. If the video is frame-encoded, the frame numbers (time line locations) shall be represented by even field numbers.

Audio packets shall contain 32,768 sample words acquired at 48,000 samples per second. The relationship of video and audio on the virtual timeline is shown in Figure 9. Also refer to Annex B.

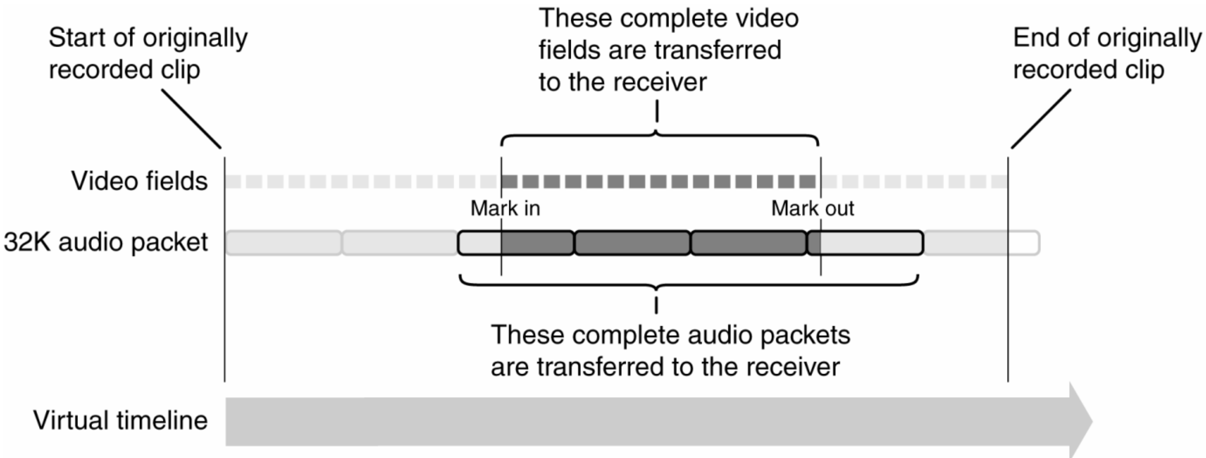


Figure 9 – Audio packets

The number of video fields or frames represented by 32,768 audio sample words depends on the field or frame rate of the corresponding video. The field number of an audio packet shall be determined by converting the position of the first audio sample location (recorded or not) in the packet to a video field number using the field rate of the current video standard. If there are no video tracks in the material, the conversion shall be based on the 59.94Hz (the exact rate is 60/1.001) field rate.

Time code packets contain 504 samples. The field number of a time code packet shall be the video field number for the first time code value (recorded or not) in the packet.

7.4.2.1.4 Field information

The field information is dependent on the media type.

For Motion JPEG and DV-based video, byte 0 of the media type shall contain the number of 4096 byte blocks required to store the field or frame. The remaining three bytes should be set to zero by sources and shall be ignored be receivers.

For audio and time code, the field information contains the first valid sample number (inclusive) and the last valid sample number (exclusive). The first and last valid sample numbers apply to the current packet only. If the entire packet of audio is valid, the first and last valid sample numbers shall be 0 and 32,768, respectively. If the entire packet of time code is valid, the first and last valid sample numbers shall be 0 and 504, respectively. Like field numbers, first valid sample shall be inclusive and last valid sample shall be exclusive. The first and last sample numbers shall be used to identify the samples that have been recorded and are therefore valid. These values shall be sent first valid sample before the last valid sample with their most significant bytes proceeding least significant bytes.

Note: In the following cases the packet could contain more samples than those referenced by the clip and the first or last valid sample in the packet will not always correspond to the first or last video field of the clip:

1) A simple clip that has mark in and mark out points different than the source media file.

2) A compound clip with segments that have mark in and mark out points different than the source media file.

3) In these cases the playback system must calculate the correct first and last sample values. See clause 5.2, clause 7.4.2.1.3, and Annex B.

For all MPEG video formats, only bytes 1, 2, and 3 shall be used for the length. They shall contain the size in bytes of the MPEG picture. This value shall be sent most significant byte first. For MPEG, byte 0 shall contain codes that provide frame information (see Table 19).

Table 19 – MPEG video frame descriptive information

Bits (0 is LSB; 7 is MSB)	Meaning
0:1	Picture coding: 00 = None 01 = I-frame 10 = P-frame 11 = B-frame
2:3	Picture structure: 00 = None 01 = Top field 10 = Bottom field 11 = Frame
4:7	Not used – shall be 0

7.4.2.1.5 Time line field number

The 32-bit unsigned field number relative to the start of the material (time line position). The time line field number shall be determined in the same way as the audio and time code packets as described in clause 7.4.2.1.3. Time line field numbers shall be assigned consecutively from the start of the material. The first time line field number shall be 0. Only even field numbers shall be used for frame-based compression systems.

7.4.2.1.6 Flags

A 1-byte value, bit 0 of which indicates the validity of the time line field number. If bit 0 is set, the time line field number is valid. If bit 0 is not set, the time line field number is not valid. Material archived before the implementation of SMPTE 360 may not have a valid time line field number. Receivers shall check the time line valid bit and if it is 0, the time line field number shall be assumed to be the same as the media field number. Senders shall provide a valid time line field number and set the time line field number valid bit. All other bits in the flag byte shall be set to 0 by the source and should be ignored by the receiver.

7.4.2.2 Time code packets

Time code packets shall have a payload length of 4112 bytes. Each time code packet contains 504 time code values plus a bitmap that defines which time code values are valid. The 504 time code values are encoded based on SMPTE 12M-1. Table 20 describes the format of a time code sample in stream byte order. In the table, bit 0 is the low-order bit of a byte; bit 7 is the high-order bit.

The valid sample bit map shall be contained in the last 64 bytes of the time code packet. It shall be comprised of sixteen 32-bit unsigned integers. Only 504 of the 512 bits in the 64-byte map are used.

Each bit in the bitmap indicates whether its corresponding time code value is valid or not. For valid time code samples, the bit value shall be one; for invalid (unrecorded) time code samples, the bit value shall be zero. The bit map shall start at bit 0, low-order bit, of the first byte and proceed through the remaining bits of the first bit map byte in ascending order. The bit map shall continue in ascending byte order and ascending bit order within each byte until all 504 time code samples are mapped. The last time code sample will be mapped by bit 7 of byte 62. Byte 63 pads the packet. Its value is undefined and shall be ignored by receivers.

Note: See clause 7.1.3.3 for a description of time code auxiliary information and striped time codes. Striped time codes are not represented as media packets.

Table 20 – Time code packet value description

Offset	Bits in byte	Contents
0x0	3:0 7:4	Units of frames First binary group
0x1	1:0 2 3 7:4	Tens of frames Drop frame flag Color frame flag Second binary group
0x2	3:0 7:4	Units of seconds Third binary group
0x3	2:0 3 7:4	Tens of seconds Frame mark Fourth binary group
0x4	3:0 7:4	Units of minutes Fifth binary group
0x5	2:0 3 7:4	Tens of minutes Binary group flag 0 Sixth binary group
0x6	3:0 7:4	Units of hours Seventh binary group
0x7	1:0 2 3 7:4	Tens of hours Binary group flag 1 Binary group flag 2 Eighth binary group

7.4.2.3 Audio packets

An audio packet shall contain 32,768 sample words with sample word sizes of either 16 or 24 bits according to the media type. The sample words will contain linear PCM audio samples for media types 9 and 10 (see Table 5), or SMPTE 337M formatted non-PCM data for media types 17 and 18 (see Table 5). An audio media packet may not have a complete set of recorded samples (see clause 7.4.2.1.4 for details). For all audio tracks types, sample words shall have the least significant byte (LSB) first and be recorded or created at 48 kHz. A track of audio samples shall consist of a single type of audio media packets.

Missing audio material should be indicated as unrecorded by not inserting the appropriate audio packets (see clause 5.2) and/or by using the first/last sample values in the audio media packet field information values (see clause 7.4.2.1.4). Missing audio sample words may be synthesized if audio media packets are inserted in the stream.

Notes:

1) Audio user-data (U bits) metadata is not transmitted by this Standard.

2) This Standard describes the interchange of material, not the details of how a source encodes a file or a receiver decodes a file. A user of this Standard will find other audio interface specifications and standards helpful in understanding the data stream.

The following are valid audio media packet types.

7.4.2.3.1 Uncompressed 16-bit PCM audio

A track with media type 10 (see Table 5) shall encode uncompressed mono audio tracks with 16-bit linear PCM audio, sampled at 48 kHz (see AES3, clause 2.1.2).

Note: Audio samples with less than 16-bits of precision can be coded using the most significant bits of 16-bit audio samples. The recommended value for the unused (least significant) bits is zero.

7.4.2.3.2 Uncompressed 24-bit PCM audio

A track with media type 9 (see Table 5) shall encode uncompressed mono audio tracks with 24-bit linear PCM audio, sampled at 48 KHz. See AES3, clause 2.1.2.

Note: Audio samples with less than 24-bits of precision can be coded using the most significant bits of 24-bit audio samples. The recommended value for the unused (least significant) bits is zero.

7.4.2.3.3 16-bit AC-3 audio

A track with media type 17 (see Table 5) shall encode one half of a single AC-3 compressed audio stream (see SMPTE 338, Table 1, data type 1). See ITU-R BS.1196 (Annex 2) or ATSC A/52 (AC-3) for a description of the data type. The AC-3 audio data shall be formatted into 16-bit sample words according to SMPTE 337 and SMPTE 340 utilizing the SMPTE 337 16-bit mode. An audio stream that is not a 16-bit AC-3 audio channel shall not use this media type.

Note: A single compressed bit stream is usually carried on one AES-3 bit stream which normally carries two mono audio tracks. One track is carried on X sub-frames and the other on Y sub-frames as defined in the AES-3 standard. The data from the X sub-frame is assumed to be stored as a one track audio file and the data from the Y sub-frame is assumed to be stored as another one track audio file, so that each file contains one half of the original compressed bit stream.

7.4.2.3.4 24-bit non-PCM audio

A track with media type 18 (see Table 5) shall encode one half of a single non-PCM audio stream containing compressed audio data of the type specified by SMPTE 338, Table 1, data type 28. See clause 7.4.2.3.3 above on what is meant by one half. The non-PCM data shall be formatted into 24-bit sample words according to SMPTE 337 utilizing any of the valid SMPTE 337 data modes (16, 20, or 24-bit modes). For the 16 and 20-bit modes, the unused bits in the 24-bit sample word should be set to zero. An audio stream that is not a non-PCM audio data stream of the appropriate type (SMPTE 338 data type 28) shall not use this media type.

Note: A single compressed bit stream is usually carried on one AES-3 bit stream which normally carries two mono audio tracks. One track is carried on X sub-frames and the other on Y sub-frames as defined in the AES-3 standard. The data from the X sub-frame is assumed to be stored as a one track audio file and the data from the Y sub-frame is assumed to be stored as another one track audio file, so that each file contains one half of the original compressed bit stream.

7.4.2.4 Motion JPEG packets

A Motion JPEG track shall consist of a sequence of JPEG compressed fields. Each JPEG packet shall contain one field of compressed video. Each field of compressed video data shall be preceded by a Motion JPEG media preamble, described in Table 21. ITU-R BT.601 8-bit video samples shall be compressed starting and finishing at the lines specified in Table 22. Value bytes shall be streamed with the least significant byte first, followed by progressively more significant bytes. The remainder of the JPEG media packet shall contain a field encoded in compliance with ITU-R T.81 (JPEG) or ISO/IEC 10918-1 (JPEG).

Note: JPEG interchange bit streams are inconsistent in the handling of data values of 0xFF and start of Information flags (0xFF.) Some JPEG file and stream formats include doubled 0xFF data values (to 0xFF, 0xFF) to distinguish this value from start of information flags. This JPEG coding does not use doubled 0xFF data values. More information can be found in JPEG-9-R7.

Table 21 – Motion JPEG media preamble

Offset	Type	Size	Description
0	INT32	256 (64 four byte words)	Luminance quantization table entries. Quantization tables shall contain 64 8-bit entries, and shall be included in the JPEG-specified zigzag order.
256	INT32	4	Reserved
260	FLOAT64	8	Quantization factor
268	BINARY	52	Reserved
320	INT32	256 (64 four byte words)	Chrominance quantization table entries. Quantization tables shall contain 64 8-bit entries, and shall be included in the JPEG-specified zigzag order.
576	INT32	4	Reserved
580	FLOAT64	8	Quantization factor
588	BINARY	52	Reserved
640	INT32	4	Identifier – Fixed value; shall be 0x95BA84F1
644	INT32	4	Version – Fixed value; shall be 1
648	INT32	4	Reserved
652	INT32	4	Reserved
656	INT32	4	Number of 4096 byte blocks used to store this field
660	UINT32	4	Field number
664	UINT32	4	Field length in words (Length in bytes divided by 4)
668	INT32	4	Field type: 0 = first field 1 = second field
672	FLOAT64	8	Reserved
680	INT32	4	Reserved

Table 22 – Motion JPEG encoding range

Number of lines	Field	Start line	Recorded end line	Lines blanked on playout
525	0	8	263	8 and 9
	1	271	Line 1 from the next frame	271, 272 and line 1 in the next frame
625	0	7	310	None
	1	320	623	None

7.4.2.5 MPEG-2 SD video packets

MPEG-2 SD (525 and 625 formats) video packets shall contain a single video frame encoded as an MPEG-2 Elementary Stream according to ISO/IEC 13818-2 (MPEG-2). MPEG-2 SD video packet lengths shall vary according to the size of the encoded video frame. If a frame length is not a multiple of 4 bytes, the frame shall be padded with trailing bytes that contain the value 0x00. The packet length in the packet header and the picture size in the media packet preamble shall reflect the padded length.

MPEG-2 SD video material shall be compliant with SMPTE RP 202.

Note: MPEG-2 SD video material will include a sequence header and extension with every GOP (preceding the I-picture).

7.4.2.5.1 Uncompressed VBI data in MPEG-2 SD

If the media packet type is either MPEG-2 525 or MPEG-2 625, the MPEG picture may contain lines of uncompressed vertical blanking interval data stored in the MPEG bit stream. Uncompressed VBI data, if present, shall be stored as MPEG user data in the MPEG bit stream, and shall be located immediately after the picture coding extension and immediately before the picture data, as defined by ISO/IEC 13818-2 (MPEG-2). The eight most significant bits of the 10-bit sample words shall be stored as uncompressed VBI data. The two least significant bits of each sample shall be discarded. Each line shall consist of 720 bytes of luma or color difference data.

The beginning of the MPEG user data shall be identified by a start code of four bytes with the following hexadecimal values: 0x00, 0x00, 0x01, and 0xB2. All subsequent bytes shall be MPEG user data until the next start code is detected in the stream. Start codes shall be four bytes in length, the first three of which have the following hexadecimal values: 0x00, 0x00, and 0x01.

To ensure that MPEG user data is not misinterpreted as start codes, MPEG user data shall be clipped so that no bytes have the hexadecimal values of 0x00 or 0xFF. Uncompressed VBI data that have a hexadecimal value of 0x00 shall be stored as 0x01. Uncompressed VBI data that have a hexadecimal value of 0xFF shall be stored as 0xFE.

The MPEG user data used to contain uncompressed VBI data shall take the form shown in Table 23.

Table 23 – Uncompressed VBI data format

Data type	Value	Description
UINT32	0x000001B2	User data start code
UINT16	0x7352	Required, OEM code
UINT16	0x2106	VBI data code
UINT16	0x8553	Data length, including the two length bytes and the line number
UINT16	Variable	Line number, from 01 to 23
UINT8	01 = Color difference line, field 1 02 = Color difference line, field 2 05 = Luma line, field 1 06 = Luma line, field 2	Line type
UINT8	Variable	720 bytes of 8 bit color difference or luma VBI data.

Uncompressed VBI data for one frame shall be in the following sequence in the MPEG User Data space. Uncompressed VBI data for each line shall be in ascending line number sequence with the luma data for a given line proceeding color difference data for the same line.

Notes:

1) Field 2 line numbers are greater than field 1 line numbers so all field 2 Uncompressed VBI data will follow the field 1 Uncompressed VBI data.

2) The reader needs to be careful to not confuse MPEG user data with user-supplied metadata (“user data”) that is described in 0 of this Standard.

7.4.2.6 MPEG-1 video packets

MPEG-1 video packets shall contain a single video frame (525 or 625) encoded as an MPEG Elementary Stream according to ISO/IEC 11172-2 (MPEG-1). MPEG-1 video packet lengths shall vary according to the size of the encoded video frame. If an MPEG-1 frame length is not a multiple of 4 bytes, the frame shall be padded with trailing bytes that contain the value 0x00. The packet length in the packet header and the picture size in the media packet preamble shall reflect the padded length.

Note: MPEG-1 material will include a sequence header and extension with every GOP (preceding the I-picture.)

7.4.2.7 MPEG-2 HD video packets

MPEG-2 HD video packets shall contain a single video frame encoded as an MPEG Elementary Stream according to ISO/IEC 13818-2 (MPEG-2) main profile at main level or high level, or 4:2:2 profile at main level or high level. MPEG-2 HD video packet lengths shall vary according to the size of the encoded video frame. If an MPEG-2 HD frame length is not a multiple of 16 bytes, the frame shall be padded with trailing bytes that contain the value 0x00. The packet length in the packet header and the picture size in the media packet preamble shall reflect the padded length.

Note: The MPEG video auxiliary information cf flag in the track description section identifies MPEG main profile and MPEG 4:2:2 profile media.

MPEG-2 HD video material shall be compliant with SMPTE RP 202 for 1080i and 720p. For 1080p, line 70 shall be the start line by encoders and decoders.

MPEG-2 HD video packets shall be a multiple of 16 bytes. If padding bytes are used with MPEG-2 media packets, the padding bytes shall have a value of 0.

Note: MPEG-2 HD material will include a sequence header and extension with every GOP (preceding the I-picture.)

7.4.2.8 DV-based video packets

DV-based video packets shall contain a single video frame encoded as a DIF packet stream according to SMPTE 314M or ISO/IEC 61834-2 for DV-based compression operating at 525 or 625. All information contained in the DV-based-frame, including audio and closed captioning data, shall be streamed from the source device to the receiver. However, the DIF stream audio data shall be duplicated and included in separate audio media packets as specified by this Standard.

Note: The source DIF stream audio data and AUX DIF packets are stored as part of the DV-based packets. Audio material is coded in audio media packets that can duplicate the DIF stream audio content. The number of audio tracks available in a stream/file is not necessarily related to the number of audio tracks in the source DIF stream. See clause 7.4.2.3 for a description of the audio media packets.

7.5 End-of-Stream Packet

The end-of-stream (EOS) packet consists of a header only. No payload section shall be included in the EOS packet (see Figure 10).

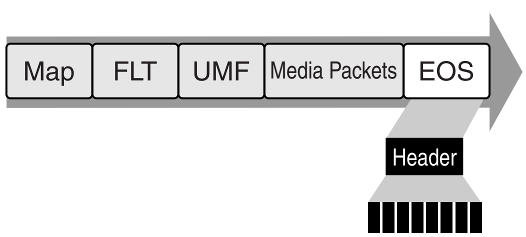


Figure 10 – End-of-stream packet

Refer to Table 2 to identify the packet type code for an end-of-stream packet.

Note: A source is required to terminate a compliant stream with an end-of-stream packet. It is recommended that a receiver recognize a stream without an end-of-stream packet is not complete. A receiver can choose to accept an incomplete stream as-is, or it can choose to reject the file. These policy decisions are beyond the scope of this Standard.

Annex A (Informative)
Transfer over IP using FTP

Streaming transfers of simple clips or compound clips over physical links that support IP may be accomplished through the Internet Engineering Task Force (IETF) File Transfer Protocol (FTP) transactions. This annex documents one implementation of transfers using FTP.

TCP/IP is used for the streaming of simple clips and compound clips because it provides the reliability required for the transmission of compressed digital video and other data, where a single bit error might result in the loss of a video frame. IP provides an industry-standard routing scheme that allows the stream to be transmitted via industry-standard network topologies, even over long distances or different physical links.

In this implementation, streaming transfers via FTP shall be invoked by the use of the movie username at the start of the FTP session. All subsequent SEND and RECV commands during that session shall be processed in such a way as to stream simple clips or compound clips between the transfer source and receiver.

A.1 Supported FTP Commands

Table A.1 lists the subset of the standard FTP commands that shall be supported by streaming FTP sessions. Unlike standard FTP operations, streaming FTP sessions are not required to support file system navigation commands. The following FTP commands shall not produce any results when attempted in streaming FTP sessions: cd, dir, and lcd.

A.2 Initiating a Transfer

The streaming transfer of material must be initiated by invoking an FTP session with the username movie and no password. All subsequent operations in that FTP session must be operations on material specified by a uniform media locator (UML) as described in the following clause. A UML provides a complete description of the source or receiver of a transfer. UMLs are similar in concept to the uniform resource locator (URL) scheme used with HTTP on the Internet.

The form for a UML must be:

<host>/<type>/<FileName>[?<options>…]

The <host> portion consists of the computer network name of the source or receiver computer. The host should be available on the network. If the host parameter is not specified, it must be assumed to be the local host.

The <type> portion must identify the type of transfer to be performed at the source or receiver. Presently, explodedFile is the only type defined for the transfer.

The <FileName> must describe the materials file name.

The bracketed ?<options> parameter indicates a group of optional arguments that may specify the priority for this transfer or a range of fields which are to be transferred or other transfer control options. This argument is specified in the following clauses.

Table A.1 – Supported FTP commands

FTP command name	FTP command description	Streaming support requirement
USER	User Name	Shall be set to “movie” with no password
PASS	Password	Not supported
ACCT	Account	Not supported
CWD	Change working directory	Not supported
CDUP	Change to parent directory	Not supported
SMNT	Structure mount	Not supported
REIN	Reinitialize	Not supported
QUIT	Logout	Shall be supported
PORT	Data port	Shall be supported
PASV	Passive	Shall be supported
TYPE	Representation type	Shall be supported
STRU	File structure	Not supported
MODE	Transfer mode	Not supported
RETR	Retrieve	Shall be supported
STOR	Store	Shall be supported
STOU	Store unique	Not supported
APPE	Append (with create)	Not supported
ALLO	Allocate	Not supported
REST	Restart	Not supported
RNFR	Rename From	Not supported
RNTO	Rename To	Not supported
ABOR	Abort	Shall be supported
DELE	Delete	Not supported
RMD	Remove directory	Not supported
MKD	Make directory	Not supported
PWD	Print working directory	Not supported
LIST	List	Not supported
NLST	Name List	Not supported
SITE	Site Parameters	Shall be supported
SYST	System	Not supported
STAT	Status	Not supported
HELP	Help	Shall be supported
NOOP	No Operation	Shall be supported

A.3 Transfer Options

The following transfer options are currently in use: (range), exact, and HOT. These options are described in the following clauses.

Multiple options may be specified in a single UML. Each option must be preceded with a question mark (‘?’).

A.3.1 Range Option

The (range) option accepts a (startField-endField) value range immediately after the question mark character (?). The startField and endField values must be expressed as a character string of decimal digits.

The number of fields to be transferred must be specified by the startField and endField arguments. The startField and endField must be field numbers as described for mark in/out and first/last field number values. This numbering is independent of the video standard in use and of time code captured as part of the recording process. It must be the responsibility of the source and receiver applications to translate between the time codes in use and the field numbers required by this Standard. To conform to the prevailing video editing convention, the frame that includes endField should not be transferred, except as described in the following paragraphs.

In general, a transfer request transfers only the media specified by the startField and the endField; however, if the material is MPEG video (any MPEG video format), additional material before the startField and after the endField must be transferred if it is required to decode the video correctly.

If the material is closed-GOP MPEG video and startField is not an I-picture, material back to and including the preceding I-picture must be transferred. If a compound clip is streamed, additional frames must be transferred as required for each video segment of the material.

If the material is open-GOP MPEG video and startField is not an I-picture, material back to and including the I-picture before the preceding I-picture must be transferred. If a compound clip is streamed, additional frames must be transferred as required for each video segment of the material.

If the material is MPEG video and the endField frame comes just after a P-picture, one additional frame must be transferred.

For DV-based material, the startField must be the first field (even field number) in a frame and the endField must be the second field (odd field number) in a frame.

If no range option is specified, the complete stream must be transferred to the receiver.

Sending devices should only emit the appropriate material as defined above. However, a receiver should accept a stream with media packets outside the requested range.

A.3.2 Exact Option

The use of this option requires the transfer of an exact copy of all the media files in their original format, including material that is not used (handles or unseen material).

Segments of several media files may be included in a compound clip. If the exact option is not specified, only the material from the media files that is included in the compound clip is transferred. Material from the media files that is not included in the compound clip is not transferred. The resulting media files on the receiver machine are not required to be exact duplicates of those on the source machine. However, the resulting media files on the receiver are subsets of those on the source machine.

If the exact option is specified, all the material of all the media files referenced by the compound clip is transferred, whether or not the material is actually used in the compound clip. The resulting media files on the receiver machine are exact duplicates of those on the source machine.

Since the exact option transfers all the media files specified by the compound clip, the startField and endField values must not be used, even if specified.

A.3.3 HOT Option

The HOT option reserves all available bandwidth for the specified transfer. All other transfers must be suspended until the HOT transfer is completed. After the HOT transfer has completed, all suspended transfers must resume.

Usage requirements for HOT transfers are:

- The word HOT must appear in capital letters after the question mark.
- Only one HOT stream must be allowed per source or receiver; a second request for a HOT stream must fail with an appropriate error message.

Annex B (Informative)
Synchronization of Media Elements

The synchronization model for video, audio, and time codes data packets is derived from independent continuous media tracks, not interleaved media blocks (i.e., SDI formatted raster time codes, embedded audio, and video.) Each media track is assumed to be a stream of samples starting at a track origin (logical zero location) for all tracks.

Audio samples are grouped into packets with 32,768 samples per packet. The stream of packets is logically addressed using video field numbers. Some video frame rates are not exact multiples of the audio sample rates so audio samples and video fields or frames will not always start at exactly the same time.

Each video "sample" (one field or one frame) is transferred as a single media packet. A fixed number (32,768) of audio samples are combined in a single audio media packet. The first audio packet for a file recorded from time zero starts at the logical origin of the track, even if the first part of the track was not recorded or the track was cut by a partial file transfer. This frequently results in unused audio samples at the beginning of the first audio packet in a file. Likewise, the last audio packet in a stream frequently has unused samples at its end.

The unit of time on the virtual timeline is measured in video field times expressed as field numbers. Since each video field (or frame) is transferred in a packet, the relationship between virtual time line time units and video packets is linear. The first video field (or frame) will not be zero if the recording started at a time other than zero or if the clip was cut or a partial file transfer occurred.

Time code samples are also combined into larger packets and they have similar start and end boundary condition resulting in partially filled packets.

B.1 An Example

The relationship of audio samples to video fields or frames is not important during file interchange. It is critical during record or playout.

For 625 line video formats, each video field has 960 audio samples, which is the quotient of 48,000 audio samples/second and 50 video fields/second. Given an audio sample number measured from the virtual timeline origin (zero), the following equations can be used to find the correct audio packet and the offset to the audio sample in that packet.

PacketNumber = (FieldNumber * 960) / 32768

PacketOffset = (FieldNumber * 960) % 32768

where '/' is the truncating integer division and '%' is the remainder of the integer division.

For 525 line video systems the number of audio samples per video field is 800.8; which is the quotient of 48,000 audio samples/second and 60/1.001 video fields/second. This results in an audio sample to video field cadence of 800, 802, 800, 802, and 800. Table B.1 shows the mapping of audio samples that are logically associated with each video field and the relationship of both to the audio media packets (32,768 samples per packet.). The following formulas were used to calculate the correct audio packet and the offset to the audio sample in that packet.

AudioPacketNumber = (FieldNumber * 800.8) % 32768

AudioSampleOffset = (FieldNumber / 5) * 4004 +

(FieldNumber % 5) * 800 + (((FieldNumber + 1) % 5) / 2) * 2

Table B.1 – Samples per field

Field number	Audio Packet number	Offset of first sample
0	0	0
1	0	800
2	0	1602
3	0	2402
4	0	3204
5	0	4004
6	0	4804
7	0	5606
8	0	6406
9	0	7208
10	0	8008
11	0	8808
. . .	. . .	. . .
39	0	31232
40	0	32032
41	1	64
42	1	866
43	1	1666
. . .	. . .	. . .

Annex C (Informative)
Bibliography

SMPTE 258M-2004, Television — Transfer of Edit Decision Lists

SMPTE 336M-2007, Data Encoding Protocol Using Key-Length-Value

SMPTE 339-2008, Format for Non-PCM Audio and Data in AES3 — Generic Data Types

SMPTE 356M-2001, Television — Type D-10 Stream Specifications — MPEG-2 4:2:2P @ ML for 525/60 and 625/50

SMPTE RP 210, Metadata Dictionary Registry of Metadata Element Descriptions

ANSI INCITS 230-1994/AM 2-1999, Information Technology — Fibre Channel — Physical and Signaling Interface (FC-PH)

ANSI INCITS 272:1996 (R2001), Information Technology — Fibre Channel — Arbitrated Loop (FC-AL)

ANSI INCITS 297-1997 (R2002), Information Technology — Fibre Channel — Physical and Signaling Interface — 2 (FC-PH-2)

EBU/SMPTE Task Force for Harmonized Standards for the Ex-change of Program Material as Bitstreams, Final Report: Analysis and Results, July 1998, SMPTE J. 107(9):603-815; 1998 September

IETF RFC 791, Internet Protocol — DARPA Internet Program Protocol Specification (IP)

IETF RFC 793, Transmission Control Protocol — DARPA Internet Program Protocol Specification

IETF RFC 959, File Transfer Protocol (FTP)

IETF RFC 4338, Transmission of IPv6, IPv4, and Address Resolution Protocol (ARP) Packets over Fibre Channel

JPEG-9-R7, Digital Compression and Coding of Continuous-Tone Still Images