FINAL
DRAFT

AMENDMENT

ISO/IEC 13818-1:2015 FDAM 2

ISO/IEC JTC **1**

Secretariat: **ANSI**

Voting begins on:
2015-03-16

Voting terminates on:
2015-05-16

Information technology — Generic coding of moving pictures and associated audio information —

Part 1: Systems

AMENDMENT 2 Delivery of timeline for external data

Technologies de l'information — Codage générique des images animées et du son associé —
Partie 1: Systèmes

AMENDEMENT 2

Please see the administrative notes on page iii

Reference number
ISO/IEC 13818-1:2013/FDAM 2:2015(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

INTERNATIONAL STANDARD
ISO/IEC 13818-1:201x/Amd.2 (E)
Rec. ITU-T H. 222.0 (xx/2013) /Amd.2
ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY -- GENERIC CODING OF MOVING PICTURES AND ASSOCIATED AUDIO INFORMATION: SYSTEMS

AMENDMENT 2

Delivery of timeline for external data

Replace the Table 2-6 with following table:

Table 2-6 – Transport Stream adaptation field

Syntax	No. of bits	Mnemonic
adaptation_field() {		
adaptation_field_length	**8**	**uimsbf**
if (adaptation_field_length > 0) {		
discontinuity_indicator	**1**	**bslbf**
random_access_indicator	**1**	**bslbf**
elementary_stream_priority_indicator	**1**	**bslbf**
PCR_flag	**1**	**bslbf**
OPCR_flag	**1**	**bslbf**
splicing_point_flag	**1**	**bslbf**
transport_private_data_flag	**1**	**bslbf**
adaptation_field_extension_flag	**1**	**bslbf**
if (PCR_flag == '1') {		
program_clock_reference_base	**33**	**uimsbf**
reserved	**6**	**bslbf**
program_clock_reference_extension	**9**	**uimsbf**
}		
if (OPCR_flag == '1') {		
original_program_clock_reference_base	**33**	**uimsbf**
reserved	**6**	**bslbf**
original_program_clock_reference_extension	**9**	**uimsbf**
}		
if (splicing_point_flag == '1') {		
splice_countdown	**8**	**tcimsbf**
}		
if (transport_private_data_flag == '1') {		
transport_private_data_length	**8**	**uimsbf**
for (i = 0; i < transport_private_data_length; i++) {		
private_data_byte	**8**	**bslbf**
}		
}		
if (adaptation_field_extension_flag == '1') {		
adaptation_field_extension_length	**8**	**uimsbf**
ltw_flag	**1**	**bslbf**
piecewise_rate_flag	**1**	**bslbf**
seamless_splice_flag	**1**	**bslbf**
af_descriptor_not_present_flag	**1**	**bslbf**
reserved	**4**	**bslbf**
if (ltw_flag == '1') {		
ltw_valid_flag	**1**	**bslbf**
ltw_offset	**15**	**uimsbf**

}		
if (piecewise_rate_flag == '1') {		
reserved	**2**	**bslbf**
piecewise_rate	**22**	**uimsbf**
}		
if (seamless_splice_flag == '1') {		
splice_type	**4**	**bslbf**
DTS_next_AU[32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
DTS_next_AU[29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
DTS_next_AU[14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
}		
if (af_descriptor_not_present_flag == '0') {		
for (i = 0; i < N; i++) {		
af_descriptor()		
}		
}		
for (i = 0; i < N; i++) {		
reserved	**8**	**bslbf**
}		
}		
for (i = 0; i < N; i++) {		
stuffing_byte	**8**	**bslbf**
}		
}		
}		

In 2.4.3.5, add to semantics:

`af_descriptor_not_present_flag` – This 1-bit field when set to '0' signals the presence of one or several `af_descriptor()` construct in the adaptation header. When this flag is set to '1' it indicates that the `af_descriptor()` is not present in the adaptation header.

`af_descriptor` may carry one or more descriptors as defined in Annex T. For descriptors carrying information associated with specific access units of an elementary stream, the descriptor applies to the first access unit that starts in the PES packet immediately following this adaptation field. There may be several TS packets carrying no payload before the start of the PES, in which case these descriptors apply to the next TS packet with payload on the same PID.
Adaptation field shall contain only complete `af_descriptor()` descriptors, i.e. a single descriptor is always contained in a single transport stream packet.
NOTE: adaptation field should remain relatively small; it is therefore recommended for large descriptors to use PES carriage as defined in Annex T.

Replace the Table 2-22 with following table:

Table 2-22 – Stream_id assignments

stream_id	Note	stream coding
1011 1100	1	program_stream_map
1011 1101	2, 9,10	private_stream_1
1011 1110		padding_stream
1011 1111	3	private_stream_2
110x xxxx		ISO/IEC 13818-3 or ISO/IEC 11172-3 or ISO/IEC 13818-7 or ISO/IEC 14496-3 audio stream number x xxxx

1110 xxxx		ITU-T Rec. H.262 | ISO/IEC 13818-2 or ISO/IEC 11172-2 or ISO/IEC 14496-2 or ITU-T Rec.H.264 | ISO/IEC 14496-10 video stream number xxxx
1111 0000	3	ECM_stream
1111 0001	3	EMM_stream
1111 0010	5	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Annex A or ISO/IEC 13818-6_DSMCC_stream
1111 0011	2	ISO/IEC_13522_stream
1111 0100	6	ITU-T Rec. H.222.1 type A
1111 0101	6	ITU-T Rec. H.222.1 type B
1111 0110	6	ITU-T Rec. H.222.1 type C
1111 0111	6	ITU-T Rec. H.222.1 type D
1111 1000	6	ITU-T Rec. H.222.1 type E
1111 1001	7	ancillary_stream
1111 1010		ISO/IEC14496-1_SL-packetized_stream
1111 1011		ISO/IEC14496-1_FlexMux_stream
1111.1100		metadata stream
1111.1101	8	extended_stream_id
1111 1110		reserved data stream
1111 1111	4	program_stream_directory

The notation x means that the values '0' or '1' are both permitted and results in the same stream type. The stream number is given by the values taken by the x's.

NOTE 1 – PES packets of type program_stream_map have unique syntax specified in 2.5.4.1.

NOTE 2 – PES packets of type private_stream_1 and ISO/IEC_13552_stream follow the same PES packet syntax as those for ITU-T Rec. H.262 | ISO/IEC 13818-2 video and ISO/IEC 13818-3 audio streams.

NOTE 3 – PES packets of type private_stream_2, ECM_stream and EMM_stream are similar to private_stream_1 except no syntax is specified after PES_packet_length field.

NOTE 4 – PES packets of type program_stream_directory have a unique syntax specified in 2.5.5.

NOTE 5 – PES packets of type DSM-CC_stream have a unique syntax specified in ISO/IEC 13818- 6.

NOTE 6 – This stream_id is associated with stream_type 0x09 in Table 2-29.

NOTE 7 – This stream_id is only used in PES packets, which carry data from a Program Stream or an ISO/IEC 11172-1 System Stream, in a Transport Stream (refer to 2.4.3.8).

NOTE 8 – The use of stream_id 0xFD (extended_stream_id) identifies that this PES packet employs an extended syntax to permit additional stream types to be identified. .

NOTE 9 - JPEG 2000 video streams (stream_type = 0x21) are carried using the same PES packet syntax as private_stream_1.

NOTE 10 – Timeline and External Media Information streams (stream_type = 0x26) are carried using the same PES packet syntax as private_stream_1.

Replace the Table 2-34 with following table:

0x26-0x7E	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Reserved

with:

0x26	Timeline and External Media Information Stream (See Annex T)
0x27-0x7E	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Reserved

Replace Table AMD8-1 by:

Table AMD8-1 – Extension descriptor

Syntax	No. of bits	Mnemonic
Extension_descriptor () {		
descriptor_tag	**8**	**uimsbf**

Syntax	No. of bits	Mnemonic
descriptor_length	**8**	**uimsbf**
extension_descriptor_tag	**8**	**uimsbf**
if (extension_descriptor_tag == 0x02) {		
ObjectDescriptorUpdate()		
}		
else if (extension_descriptor_tag == 0x03) {		
HEVC_timing_and_HRD_descriptor()		
}		
else if (extension_descriptor_tag == 0x04) {		
af_extensions_descriptor **()**		
}		
else {		
for (i=0; i<N; i++) {		
reserved	**8**	**bslbf**
}		
}		
}		

Add the following immediately before Table AMD8-2:

af_extensions_descriptor()– This structure is defined in 2.6.97.

Replace Table AMD8-2 by:

Table AMD8-2: Extension descriptor Tag values

Extension_descriptor_tag	TS	PS	Identification
0	n/a	n/a	Reserved
1	n/a	X	Forbidden
2	X	X	ODUpdate_descriptor
3	X	n/a	HEVC_timing_and_HRD_descriptor()
4	X	n/a	af_extensions_descriptor()
3-255	n/a	n/a	Rec. ITU-T H.222.0 | ISO/IEC 13818-1 Reserved

Add the following immediately after clause 2.6.96 as new subclause:

2.6.97 AF Extensions descriptor

The AF extensions descriptor is used to signal that Adaptation Field descriptors could be present in the adaptation header of the component, as defined in 2.4.3.5.
NOTE: There may be AF descriptors in an adaptation field of a TS packet even though this descriptor is not set for the component.

Table X-1 – Adaptation Field Extension Descriptor

Syntax	No. Of bits	Mnemonic
af_extensions_descriptor() { }		

In section 2.14.1 add to Note 5:

"When carrying AVC base and SVC enhancement layers in different elementary streams, usage of VDRD is strongly recommended if access units are not aligned with PES packets."

Insert following annex T:

Annex T
Carriage of Timeline and External Media Information over MPEG-2 Transport Streams

(This annex forms an integral part of this Recommendation | International Standard.)

T.1 Introduction

This annex specifies a format for carriage of timeline and location of external media resource that may be used as a synchronized enhancement of an MPEG-2 transport stream. The possible resolving, consumption and rendering of external media indicated in the stream are out of scope of this Recommendation | International Standard.

The format specifies the mapping of the transport stream program clock to an embedded timeline, the signaling of associated external resources, hereafter called add-on(s), and the signaling of prefetching events. The format is designed to be compact in order to fit within one TS packet for common use cases. The mapping of the embedded timeline indicated in the PES packet payload or in the adaptation field descriptor with the PTS value of the PES header of the PES packet provides a stable timeline for media streams in the program, regardless of PCR discontinuities or other timestamps rewriting that may happen in the network.

In the context of this annex, the “timeline and external media information” stream is called TEMI stream. The TEMI stream describes external data and associated timing for the program in the MPEG-2 Transport Stream with which the TEMI stream is associated through the Program Map Table.

T.2 TEMI access unit and TEMI elementary stream

The format of the TEMI access unit is defined in Table T-1. TEMI Access Units shall be carried as PES packets using private_stream_1 streamID and identified in the Program Map Table by the stream type 0x26. There shall be at most one TEMI elementary stream declared in the Program Map Table.

The payload of a TEMI PES Packet is a single complete TEMI_AU, i.e. there shall be one and only one complete TEMI Access Unit in a TEMI PES packet.

The TEMI PES Packet header shall contain a PTS timestamp, whose value is used to match the current System Time Clock with the timeline value embedded in the TEMI packet payload, as defined in Table T-1.

A TEMI_AU is made of one or several AF descriptors. These AF descriptors may be sent in different access units and at different rates, and are independently decodable. All TEMI access units are therefore random access points.

NOTE1: in order to avoid interpolation issues when frame-accurate synchronization is required, the indicated PTS should be the same as the PTS of the associated video or audio stream for which frame accurate sync is needed.

NOTE2: It is possible to perform timeline interpolation in-between TEMI access units, for example if multiple audio frames are packed in a single PES packet, or when the TEMI AU frequency is less than the media AU frequency. However, receivers detecting PCR discontinuities in-between TEMI AUs should be careful when performing interpolation.

Table T-1 – TEMI Access Unit

Syntax	Nb bits	Mnemonic
TEMI_AU {		

Syntax	Nb bits	Mnemonic
CRC_flag	1	bslbf
reserved	7	bslbf
for (i=0; i<N; i++) {		
af_descriptor();		
}		
if (CRC_flag) {		
CRC_32	32	rpchof
}		
}		

Each TEMI AU is composed of an entire number of AF descriptors.

CRC_flag : A 1-bit flag, which when set to '1' indicates that a CRC field is present in the packet.

CRC_32 – This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex A after processing the entire payload of the TEMI access unit.

T.3 AF Descriptors

T.3.1 Introduction

AF descriptors are structures used to carry various features of the timeline or other information. All AF descriptors have a format that begins with an 8-bit tag value. The tag value is followed by an 8-bit AF descriptor length and data fields. The following semantics apply to the descriptors defined throughout annex T.

af_descr_tag – The af_descr_tag is an 8-bit field which identifies each AF descriptor.
Table T-2 provides the Rec. ITU-T H.222.0 | ISO/IEC 13818-1 defined, Rec. ITU-T H.222.0 | ISO/IEC 13818-1 reserved, and user available AF descriptor tag values.
af_descr_length – The af_descr_length is an 8-bit field specifying the number of bytes of the AF descriptor immediately following af_descr_length field.

Table T-2 – AF Descriptor Tags

AF Descriptor Tag	Identification
0x00-0x03	Rec. ITU-T H.222.0 | ISO/IEC 13818-1 Reserved
0x04	Timeline Descriptor
0x05	Location Descriptor
0x06	BaseURL Descriptor
0x07 - 0x7F	Rec. ITU-T H.222.0 | ISO/IEC 13818-1 Reserved
0x80 – 0xFF	User Private

AF descriptors may be carried in the adaptation field of TS packets of a media elementary stream, as defined in subclause 2.4.3.5.

T.3.2 Location Descriptor

The location descriptor is used to signal the location of external data that can be synchronized with the program. It conveys several locations and their type (optionally including MIME types), along with the ability to signal upcoming external data association though a countdown until activation of the external data. It is possible to signal splicing of external data, by signaling that the newly associated data is temporary and the previous association will be re-used later on.