Professional Metadata (PMD) Specification

Table of Contents

List Of Tables

1 Scope

This document defines the syntax of Professional Metadata (PMD).

2 Normative References

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

ISO 639-1:2002, Codes for the representation of names of languages -- Part 1: Alpha-2 code

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

ISO/IEC 10646:2014, Information technology -- Universal Coded Character Set (UCS)

ETSI TS 103 190-1 V1.3.1 (2018-02), Digital Audio Compression (AC-4) Standard; Part 1: Channel based coding

ATSC A/336:2018, Content Recovery in Redistribution Scenarios

IETF RFC 4122:2005, A Universally Unique Identifier (UUID) URN Namespace

SMPTE RP 2092-1:2015, Advertising Digital Identifier (Ad-ID®) Representations

SMPTE RP 2079:2013, Digital Object Identifier (DOI) Name and Entertainment ID Registry (EIDR) Identifier Representations

3 Terms and Definitions

3.1 Payload

Group of metadata

3.2 Audio Metadata Payload

Group of audio metadata

3.3 PMD

Professional Metadata

3.4 CM

Complete Main

3.5 M&E

Music and Effects

3.6 D

Dialog element (mono)

3.7 VDS

Video Descriptive Service (mono)

3.8 O

Generic object (mono), i.e. PA feed

3.9 ED2

Enhanced Dolby E

3.10 UTF-8

Variable width character encoding in Unicode using one to four 8-bit bytes, as defined in ISO/IEC 10646

3.11 Audio Essence Signal

Mono audio signal (track) included in the audio essence associated with PMD

3.12 Audio Bed

Combination of audio essence signals that produce a set of channels corresponding to a fixed speaker configuration

3.13 Audio Object

Audio element that is sourced from a single audio essence signal and which may include dynamic characteristics

3.14 Audio Element

Audio beds or audio objects, generated from audio essence signals

3.15 Audio Presentation

Combination of audio elements that correspond to a single end-user experience

4 PMD Payload Syntax

PMD Payloads are described in a syntax roughly based on C language syntax, but simplified for ease of reading. Field elements contained in the payload bit stream are designated by bold face font.

5 PMD Payloads

5.1 Audio Presentation Description (APD)

5.1.1 Audio Presentation Description General

An Audio Presentation Description payload describes the audio elements that are combined to produce audio presentations. Each audio presentation corresponds to a single end-user experience.

5.1.2 Audio Presentation Description Payload Syntax

Syntax	No. of Bits
audio_presentation_description_payload() {	
do {	
audio_presentation_id	9
speaker_config	5
langcod	15
do {	
audio_element_id	12
} while (audio_element_id != 0)	
} while (bytes remain)	
}	

5.1.3 Audio Presentation Description Payload Fields

5.1.3.1 **audio_presentation_id**

This field specifies the audio presentation identifier, which is used to uniquely identify this presentation in other PMD payloads. Allowed values are 1-511, 0 is reserved.

5.1.3.2 **speaker_config**

This field specifies the channel makeup of the presentation being specified. It takes values as shown in Table 3.

5.1.3.3 **langcod**

This field specifies a two or three character code value representing an ISO 639-1 or 639-2 language identifier. The 15 bits are split into 3 groups of 5 bits, each group represents a lower case character logically called ***charval***, encoded with the scheme listed in Table 1.

Table 1 – Character Encoding

charval	Meaning
0x0	ISO 639-1 termination
0x1 – 0x1b	a – z (1-27)
0x1c – 0x1f	RESERVED

5.1.3.4 **audio_element_id**

This field specifies the audio element identifier for any audio beds or objects that belong to the presentation. Allowed identifiers are in the range 1 – 4095, as listed in Table 2. A presentation may have any number of element identifiers associated to it. The list is terminated by an element identifier of value 0. In use cases

where multiple beds are within a single presentation they are summed prior to the presentation being rendered.

Table 2 – Audio Element ID

element_id	Meaning
0x0	Element list termination
0x1 – 0xfff	Audio element identifier

5.2 Audio Presentation Names (APN)

5.2.1 Audio Presentation Names General

The Audio Presentation Names payload describes human-readable informative descriptions of audio presentations.

5.2.2 Audio Presentation Names Payload Syntax

Syntax	No. of Bits
audio_presentation_names_payload() {	
do {	
audio_presentation_id	9
langcod	15
do {	
charval	8
} while (charval != 0)	
} while (bytes remain)	
}	

5.2.3 Audio Presentation Names Payload Fields

5.2.3.1 **audio_presentation_id**

This field specifies the identifier of the audio presentation to which the following name will be associated. Allowed values are 1-511, 0 is reserved.

5.2.3.2 **langcod**

This field specifies a two or three character code value representing an ISO 639-1 or 639-2 language identifier. The 15bits are split into 3 groups of 5 bits, each group represents a lower case character logically called ***charval***, encoded with the scheme listed in Table 1.

5.2.3.3 **charval**

This field specifies a series of UTF-8 encoded characters that will be appended to this string to form the complete name. To terminate and complete the string this value is set to 0.

5.3 Presentation Loudness Description (PLD)

5.3.1 Presentation Loudness Description General

The Presentation Loudness Description payload describes the loudness information of an audio presentation.

5.3.2 Presentation Loudness Description Payload Syntax

Syntax	No. of Bits
presentation_loudness_description_payload() {	
audio_presentation_id	9
do {	var
further_loudness_info()	
} while (bytesremain)	
}	

5.3.3 Presentation Loudness Description Payload Fields

5.3.3.1 **audio_presentation_id**

This field specifies an audio presentation identifier to which this loudness payload applies, allowed values are 1-511, 0 is reserved.

5.3.3.2 **further_loudness_info**

This field contains programme loudness data as specified in the futher_loudness_payload defined in ETSI TS 103 190-1 v1.2.1 section 4.2.14.3.

5.4 Audio Bed Description (ABD)

5.4.1 Audio Bed Description General

The Audio Bed Description payload describes audio elements that are channel-based beds. A bed is created by mixing up to 255 source audio signals, each with its own mixing gain value, to produce a set of channels that are intended to be output to the specified speaker configuration.

5.4.2 Audio Bed Description Payload Syntax

Syntax	No. of Bits
audio_bed_description_payload() {	
do {	
audio_element_id	12
speaker_config	5
type	1
if (type == 1) {	
source_abd_id	12
}	

do {	
target	6
source	8
source_gain	6
} while (target != 0)	
} while (bytes remain)	
}	

5.4.3 Audio Bed Description Payload Fields

5.4.3.1 **audio_element_id**

This field specifies the unique audio element identifier for this bed audio element. The audio element identifier is used to uniquely identify this particular audio element within the overall audio presentation. Each audio_element_id value shall be uniquely assigned to either a bed or object audio element within an audio presentation. Allowed values are 1 – 4095, a value of 0 is reserved.

5.4.3.2 **speaker_config**

This field specifies the number of channels present and their position within the bed. It takes values as shown in Table 3.

Table 3 - Audio Bed Speaker Config

speaker_config	Meaning
0x0	2.0
0x1	3.0
0x2	5.1
0x3	5.1.2
0x4	5.1.4
0x5	7.1.4
0x6	9.1.6
0x7 – 0x1c	RESERVED
0x1d	Portable Speakers
0x1e	Portable Headphone
0x1f	Not Indicated

5.4.3.3 **type**

This field indicates whether or not the audio source is from or is derived from another bed, *i.e*. a custom downmix or remap of an existing ABD element. It takes values as shown in Table 4.

Table 4 - Audio Bed Type

type	Meaning
0x0	Original
0x1	Derived

5.4.3.4 **source_abd_id**

This field, when present, specifies an ABD instance to be used as the audio signal source, 1 – 4095, 0 is reserved. The value of this field shall match the audio_element_id of a channel bed audio element, and shall not match the audio_element_id of an object audio element. The ABD instance is used as the signal source rather than the audio essence signals, i.e. the referenced bed is rendered first, then its output channels are routed to the input channels of this ABD instance.

5.4.3.5 **target**

This field specifies the identity of the output speaker position. It takes values as shown in Table 5.

Table 5 - Audio Output Target

target	Meaning
0x0	END-OF-CHANNELS*
0x1	Left
0x2	Right
0x3	Center
0x4	LFE
0x5	Left Surround
0x6	Right Surround
0x7	Left Rear Surround
0x8	Right Rear Surround
0x9	Left Top Front
0xa	Right Top Front
0xb	Left Top Middle
0xc	Right Top Middle
0xd	Left Top Rear
0xe	Right Top Rear
0xf	Left Front Wide
0x10	Right Front Wide
0x11 – 0x3f	RESERVED

*NOTE: target='0' is used to indicate that the channel list has finished.

5.4.3.6 **source**

This field specifies the index (track or channel number) of the audio essence signal that contains the source audio data for the target speaker position.

5.4.3.7 **source_gain**

This field specifies the gain to apply to the audio signal source when mixing into the target speaker position. It takes the range -25.0 dB to +6.0 dB in steps of 0.5 dB plus a setting for muting the audio signal source as illustrated in Table 6.

Table 6 - Audio Object Source Gain

source_gain	Meaning
0x0	Infinite attenuation (i.e., muted)
0x1	-25.0 dB
0x2 - 0x32	-24.5dB to -0.5dB
0x33	0 dB
0x34	0.5 dB
0x35 - 0x3e	1dB to 5.5dB
0x3f	6.0 dB

5.5 Audio Object Description (AOD)

5.5.1 **Audio Object Description** General

The Audio Object Description payload describes audio objects (audio elements that are sourced from single audio signals). Objects can be dynamic in nature. Parameters such as position or size can change over time. The Audio Objection Description provides a set of parameters that establish the state of audio objects that is valid until the a subsequent payload (AOB or other payload type) modifies the state.

For example, in applications where audio metadata is aligned with frames of related video content, Audio Object Description payloads may be sent once per video frame, and finer control of positional metadata may be achieved by sending one or more Dynamic Position Update payloads (see Section 5.7) prior to the AOD payload aligned with the following video frame.

5.5.2 Audio Object Description Payload Syntax

Syntax	No. of Bits
audio_object_description_payload() {	
do {	
audio_element_id	12
class	4
dynamic_update	1
x_pos	10
y_pos	10
z_pos	10
size	5
size_3d	1
diverge	1
source	8
source_gain	6
} while (bytes remain)	
}	

5.5.3 Audio Object Description Payload Fields

In the following section cartesian coordinates are defined that represent locations inside of a nominal cube. The screen is located on the front wall, positions left and right are defined as relative to an observer facing the screen.

5.5.3.1 **audio_element_id**

This field specifies the unique audio element identifier for this object audio element. The audio element identifier is used to identify this particular audio element within the overall audio presentation. Each audio_element_id value shall be unique within an audio presentation. Allowed values are 1 – 4095, 0 is reserved.

5.5.3.2 **class**

This field indicates the semantic content the object carries. It takes values as shown in Table 7.

Table 7 - Audio Object Class

class	Meaning
0x0	Dialog
0x1	VDS (Video Descriptive Service)
0x2	Voiceover
0x3	Generic Object
0x4	Spoken Subtitle
0x5	Emergency Alert
0x6	Emergency Information
0x7 - 0xf	RESERVED

5.5.3.3 **dynamic_update**

This field indicates whether or not an object's metadata values will update throughout the duration of the essence. It takes values as shown in Table 8.

Table 8 - Audio Object Dynamic Updates

dynamic_updates	Meaning
0x0	Metadata values are static
0x1	Metadata values may change

5.5.3.4 **x_pos**

This field specifies the X coordinate of the object's position within the cube at the current moment in time. It takes values in the range 1 – 0x3ff, representing points on and within the cube in increments of 1/1023 as shown in Table 9.

Table 9 - Audio Object X Coordinate

X Coordinate	Meaning
0x0	RESERVED
0x1	Left wall
0x2 – 0x1ff	Variable position between left and middle
0x200	Middle
0x201 – 0x3fe	Variable position between middle and right
0x3ff	Right wall

5.5.3.5 **y_pos**

This field specifies the Y coordinate of the object's position within the cube at the current moment in time. It takes values in the range 1 – 0x3ff, representing points on and within the cube in increments of 1/1023 as shown in Table 10.

Table 10 - Audio Object Y Coordinate

Y Coordinate	Meaning
0x0	RESERVED
0x1	Back wall
0x2 – 0x1ff	Variable position between back and middle
0x200	Middle
0x201 – 0x3fe	Variable position between middle and front
0x3ff	Front wall

5.5.3.6 **z_pos**

This field specifies the Z coordinate of the object's position within the cube at the current moment in time. It takes values in the range 1 – 0x3ff, representing points on and within the cube in increments of 1/1023 as shown in Table 11.

Table 11 - Audio Object Z Coordinate

Z Coordinate	Meaning
0x0	RESERVED
0x1	Bottom
0x02 – 0x1ff	Variable position between bottom and middle
0x200	Middle
0x201 – 0x3fe	Variable position between middle and top
0x3ff	Top

5.5.3.7 **size**

This field specifies the amount of spread applied to an object. It takes values 0 – 31 as shown in Table 12.

Table 12 - Audio Object Size

size	Meaning
0x0	No spread (point source)
…	…
0x1f	Maximum spread (entire field)

5.5.3.8 **size_3d**

This field specifies whether or not the object (when its size is greater than zero) is flat *i.e.,* only contributes energy into a 2-D horizontal plane or it contributes energy into 3-D horizontal and vertical planes. It takes values as shown in Table 13.

Table 13 - Audio Object Size Vertical

size_3d	Meaning
0x0	2-D Horizontal plane only
0x1	3-D Horizontal and vertical planes

5.5.3.9 **diverge**

This field specifies whether to implement divergence by cloning the object into its mirror front quadrant on the X axis and reducing the gain of each object by -3dB to maintain the correct energy level. The Y axis position is the same for both the original and cloned objects. The parameter takes values as shown in**Error! Reference source not found.** Table 14**Error! Reference source not found.**.

Table 14 - Audio Object Diverge

diverge	Meaning
0x0	No divergence
0x1	Clone object in mirror-front quadrant

5.5.3.10 **source**

This field specifies the index (track or channel number) of the audio essence signal that contains the source audio data for the target object.

5.5.3.11 **source_gain**

This field specifies the gain that should be applied to the audio signal source before the element is mixed into the final presentation. It takes the range -25.0 dB to +6.0 dB in steps of 0.5 dB plus a setting for muting the audio signal source, as illustrated in Table 15.

Table 15 - Audio Object Source Gain

source_gain	Meaning
0x0	Infinite attenuation (i.e., muted)
0x1	-25.0 dB
0x2 - 0x32	-24.5dB to -0.5dB
0x33	0 dB
0x34	0.5 dB
0x35 - 0x3e	1dB to 5.5dB
0x3f	6.0 dB

5.6 Headphone Element Description (HED)

5.6.1 Headphone Element Description General

The Headphone Element Description payload is a supplemental set of parameters that are used to further refine the behaviour of a previously declared audio elements that are to be included in binaural processing delivery workflows for headphone reproduction.

5.6.2 Headphone Element Description Payload Syntax

Syntax	No. of Bits
headphone_element_description_payload() {	
do {	
audio_element_id	12
head_tracking_enabled	1
render_mode	7
If(audio_element_id::type == ABD){	
channel_mask	16
}	
} while (bytes remain)	
}	

5.6.3 Headphone Element Description Payload Fields

5.6.3.1 **audio_element_id**

This field specifies the audio element for which the following headphone parameter data will be associated. Allowed values are 1-4095, 0 is reserved.

5.6.3.2 **head_tracking_enabled**

This field specifies whether the audio object position is relative to the listener's head (head tracking is disabled) or the audio object position is relative to the world outside the listener (head tracking is enabled).

Allowed values are 0–1, the value 0 indicates head tracking is disabled, the value 1 indicates head tracking is enabled.

5.6.3.3 **render_mode**

This field specifies the audio element room reverberation amount. It takes values in the range 0–127, where 0 represents anechoic (no room reverb present) and 127 represents maximum room reverb present.

5.6.3.4 **channel_mask**

This field, if present, specifies which channels of an audio bed are to be processed. Each bit represents an individual speaker designation that when the value 1 is added aligns with the channel designations detailed in Table 5, e.g., bit 0 = Left, bit 8 = Left Top Front, bit 15 = Right Front Wide. Allowed values for each bitfield are 0–1, 0 indicates that processing is disabled, 1 indicates that processing is enabled.

5.7 Dynamic Position Update (XYZ)

5.7.1 Dynamic Position Update General

The Dynamic Position Update payload describes updates to the positions of audio objects.

5.7.2 Dynamic Position Update Payload Syntax

Syntax	No. of Bits
dynamic_position_update_payload() {	
sample_time	6
do {	
audio_element_id	12
x_pos	10
y_pos	10
z_pos	10
} while (bytes remain)	
}	

5.7.3 Dynamic Position Update Payload Fields

5.7.3.1 **sample_time**

This field specifies the sample time at which the position update occurs. Sample time is indicated in units of 32 samples relative to the time of the most recent AOD payload that contains position information for the specified audio object.

5.7.3.2 **audio_element_id**

This field specifies the object which is being updated via its object identifier, Allowed values are 1-4095, 0 is reserved.

5.7.3.3 **x_pos**

This field specifies the X coordinate of the object's position at the specified sample time. The value of x_pos is as defined in Section 5.5.3.4.

5.7.3.4 **y_pos**

This field specifies the Y coordinate of the object's position at the specified sample time. The value of y_pos is as defined in Section 5.5.3.5.

5.7.3.5 **z_pos**

This field specifies the Z coordinate of the object's position at the specified sample time. The value of z_pos is as defined in Section 5.5.3.6.

5.8 Audio Element Names (AEN)

5.8.1 Audio Element Names General

The Audio Element Names payload describes human-readable informative descriptions of audio beds and objects.

5.8.2 Audio Element Names Payload Syntax

Syntax	No. of Bits
audio_element_names_payload() {	
do {	
audio_element_id	12
do {	
charval	8
} while (charval != 0)	
} while (bytes remain)	
}	

5.8.3 Audio Element Names Payload Fields

5.8.3.1 **audio_element_id**

This field specifies the identifier to which the following name will be associated. Allowed values are 1-4095, 0 is reserved.

5.8.3.2 **charval**

This field specifies a series of UTF-8 encoded characters that will be appended to this string to form the complete name. To terminate and complete the string this value is set to 0.

5.9 ED2 Substream Description (ESD)

5.9.1 ED2 Substream Description General

The ED2 Substream Description payload is only present in PMD audio metadata that is transmitted as part of ED2 and describes the properties of the ED2 substream that it is found within.

5.9.2 ED2 Substream Description Payload Syntax

Syntax	No. of Bits
ed2_substream_description_payload() {	
stream_count_minus1	4
substream_index	4
stream_frame_rate	4
stream_config	5
internal_reserved	3
}	

5.9.3 ED2 Substream Description Payload Fields

5.9.3.1 stream_count_minus1

This field specifies the total number of individual ED2 substreams that comprise the entire ED2 system stream. Each ED2 substream is capable of carrying up to 8 channels of audio essence. Add one to the value of this field to get the total number.

5.9.3.2 substream_index

This field specifies a 0-based index of the particular ED2 substream this payload is found within in relation to the entire set of ED2 substreams that comprise the ED2 system stream. Allowed values are 0-15.

5.9.3.3 stream_frame_rate

This field specifies the underlying Dolby E frame rate. High frame rates (those above 30 fps) are not supported. This field takes values as shown in Table 16.

Table 16 - ED2 Substream Description Stream Rate

stream_frame_rate	Meaning
0x0	RESERVED
0x1	23.98 fps
0x2	24 fps
0x3	25 fps
0x4	29.97 fps
0x5	30 fps
0x6 – 0xf	RESERVED

5.9.3.4 stream_config

This field specifies the Dolby E Program Configuration. This field takes values as shown in Table 17.

Table 17 - ED2 Substream Description Stream Config

stream_config	Meaning
0x0	5.1+2
0x1	5.1+1+1
0x2	4+4
0x3	4+2+2
0x4	4+2+1+1

0x5	4+1+1+1+1
0x6	2+2+2+2
0x7	2+2+2+1+1
0x8	2+2+1+1+1+1
0x9	2+1+1+1+1+1+1
0xa	1+1+1+1+1+1+1+1
0xb	5.1
0xc	4+2
0xd	4+1+1
0xe	2+2+2
0xf	2+2+1+1
0x10	2+1+1+1+1
0x11	1+1+1+1+1+1
0x12	4
0x13	2+2
0x14	2+1+1
0x15	1+1+1+1
0x16	7.1
0x17	7.1 screen
0x18 – 0x1f	RESERVED

5.9.3.5 **reserved**

This field is reserved.

5.10 ED2 Substream Names (ESN)

The ED2 Substream Names payload is only present in PMD audio metadata that is transmitted as part of ED2 and describes a human-readable textual description of the ED2 substream that it is found within.

5.10.1 ED2 Substream Names Payload Syntax

Syntax	No. of Bits
ed2_substreams_names_payload() {	
do {	
substream_id	4
do {	
charval	8
} while (charval != 0)	
} while (bytes remain)	
}	

5.10.2 ED2 Substream Names Payload Fields

5.10.2.1 **substream_id**

This field specifies the substream identifier to which the following name will be associated. Allowed values are 1-15, 0 is reserved.

5.10.2.2 **charval**

This field specifies a series of UTF-8 encoded characters that will be appended to this string to form the complete name. To terminate and complete the string this value is set to 0

5.11 ED2 Turnaround Description (ETD)

5.11.1 ED2 Turnaround Description General

The ED2 Turnaround Description payload describes how to repackage an ED2 system stream into a deliverable containing a subset of audio essence and metadata for delivery in either ED2 or Dolby E formats.

5.11.2 ED2 Turnaround Description Payload Syntax

Syntax	No. of Bits
ed2_turnaround_description_payload() {	
do {	
ed2_turnaround_id	8
ed2_exists	1
If (ed2_exists == 1) {	
stream_frame_rate_code_1	4
do {	
audio_presentation_id_1	9
eac3_enc_params_id_1	8
} while (0 != audio_presentation_id_1)	
}	
dolbye_exists	1
if (dolbye_exists == 1) {	
stream_frame_rate_code_2	4
pgmconfig	5
do {	
audio_presentation_id_2	9
eac3_enc_params_id_2	8
} while (0 != audio_presentation_id_2)	
}	
} while (bytes remain)	
}	

5.11.3 ED2 Turnaround Description Payload Fields

5.11.3.1 **ed2_turnaround_id**

This field specifies a unique identifier for each individual turnaround configuration, 1 – 255, 0 is reserved.

5.11.3.2 **ed2_exists**

This field indicates whether ED2 turnaround parameters are present in the payload. If so, the ED2 metadata will occur immediately after the bit.

5.11.3.3 **stream_frame_rate_code_1**

This field, if present, specifies the required ED2 frame rate for transmission. It takes values shown in Table 16.

5.11.3.4 **audio_presentation_id_1**

This field, if present, specifies an audio presentation description and its associated un-rendered audio essence to be included as part of the output ED2 system stream. Multiple presentations can be included. Allowed values are 1-511, the value 0 indicates that the presentation list has ended.

5.11.3.5 **eac3_enc_params_id_1**

This field, if present, specifies a set of E-AC-3 encoding parameters to include with the associated audio presentation description. Allowed values are 1–255, the value 0 indicates that the ED2/Dolby E transcoder should use a set of predefined default E-AC-3 encoding parameters.

5.11.3.6 **dolbye_exists**

This field indicates whether Dolby E turnaround parameters are present in the payload. If so, the Dolby E metadata will occur immediately after the bit.

5.11.3.7 **pgmconfig**

This field, if present, specifies the required Dolby E program config. It takes values shown in Table 17.

5.11.3.8 **stream_frame_rate_code_2**

This field, if present, specifies the required Dolby E frame rate for transmission. It takes values shown in Table 16.

5.11.3.9 **audio_presentation_id_2**

This field, if present, specifies an audio presentation to be rendered and the subsequent audio essence to be included in the current Dolby E program. Allowed values are 1-511, the value 0 indicates that the presentation list has ended.

5.11.3.10 **eac3_enc_params_id_2**

This field, if present, specifies a set of E-AC-3 encoding parameters to include with the associated Dolby E program. Allowed values are 1–255, the value 0 indicates that the Dolby E encoder should use a set of predefined default E-AC-3 encoding parameters.

5.12 E-AC-3 Encoding Parameters (EEP)

5.12.1 E-AC-3 Encoding Parameters General

The E-AC-3 Encoding Parameters payload describes metadata that configures E-AC-3 encoder parameters (parameters that are not conveyed to a decoder) and parameters that are used to optimize the content based upon decoder capabilities.

5.12.2 E-AC-3 Encoding Parameters Payload Syntax

Syntax	No. of Bits
eac3_encoding_parameters_payload() {	
do {	
eac3_enc_params_id	8
encoder_params_exist	1
if (encoder_params_exist == 1) {	
dynrng_prof	3
compr_prof	3
surround90	1
hmixlev	5
}	
bitstream_params_exist	1
if (bitstream_params_exist == 1) {	
bsmod	3
dsurmod	2
dialnorm	5
dmixmod	2
ltrtcmixlev	3
ltrtsurmixlev	3
lorocmixlev	3
lorosurmixlev	3
}	
drc_exists	1
if (drc_exists == 1) {	
drc_port_spkr	3
drc_port_hphon	3
drc_flat_panl	3
drc_home_thtr	3
drc_ddplus	3
}	
do {	
audio_presentation_id	9
} while (audio_presentation_id != 0)	
} while (bytes remain)	
}	

5.12.3 E-AC-3 Encoding Parameters Payload Fields

5.12.3.1 **eac3_enc_params_id**

This field specifies a unique identifier for each individual set of E-AC-3 encoding parameters, 1 -255.

5.12.3.2 **encoder_params_exist**

This field specifies whether this payload includes encoder information. If its value is 0, there is no encoder information.

5.12.3.3 **dynrng_prof**

This field, if present, specifies the required compression profile the E-AC-3 encoder must use to calculate the dynrng DRC gain words of the output AC-3 or E-AC-3 bitstream. It takes values as listed in Table 18.

Table 18 - Encoder Configuration DRC Type

drc type	Meaning
0x0	None
0x1	Film Standard
0x2	Film Light
0x3	Music Standard
0x4	Music Light
0x5	Speech
0x6 – 0x7	RESERVED

5.12.3.4 **compr_prof**

This field, if present, specifies the required AC-3 RF mode value the DD encoder must insert into its metadata. It takes values as listed in Table 18.

5.12.3.5 **surround90**

This field, if present, specifies whether or not the encoder should phase-shift the surrounds with respect to the fronts. It takes values shown in Table 19.

Table 19 - Encoder Configuration Surround90

surround90	Meaning
0x0	Surrounds are in-phase with fronts
0x1	Surrounds are phase-shifted 90 degrees from fronts

5.12.3.6 **hmixlev**

This field, if present, specifies how much attenuation should be applied to height channels when downmixing into a 5.1 core. It takes values shown in Table 20.

Table 20 - Encoder Configuration Hmixlev

Hmixlev	Meaning
N = 0x0 – 0x1e	*N* dB attenuation
0x1f	Infinite attenuation (i.e., ignore)

5.12.3.7 **bitstream_params_exist**

This field specifies whether this payload includes E-AC-3 bitstream information. If its value is 0, there is no AC-3 bitstream information.

5.12.3.8 **bsmod**

This field, if present, specifies the required AC-3 bitstream mode of the audio essence. It takes values as shown in Table 21.

Table 21 - Encoder Configuration AC-3 Bsmod

bsmod	Meaning
0x0	Complete Main

0x1	Music and Effects
0x2	Visually Impaired
0x3	Hearing Impaired
0x4	Dialogue
0x5	Commentary
0x6	Emergency
0x7	Voice Over

5.12.3.9 **dsurmod**

This field, if present, specifies the required dsurmod value the E-AC-3 encoder must insert into its metadata. It takes values as listed in Table 22. This parameter is only valid if the presentation identified by the audio_presentation_id value is stereo only.

Table 22 - Encoder Configuration AC-3 Dsurmod

dsurmod	Meaning
0x0	Not Indicated
0x1	NOT Dolby surround encoded
0x2	Dolby surround encoded
0x3	RESERVED

5.12.3.10 **dialnorm**

This field, if present, specifies the AC-3 dialogue normalization level for the content. This in turn indicates how much lower the dialogue is than digital 100% in decibels. It takes values listed in Table 23.

Table 23 - Encoder Configuration AC-3 Dialnorm

dialnorm	Meaning
0	RESERVED
N = 1 - 31	-*N* dB

5.12.3.11 **dmixmod**

This field, if present, specifies the preferred downmix mode, LtRt or LoRo. It takes values shown in Table 24.

Table 24 - Encoder Configuration Dmixmod

prefdmixmod	Meaning
0x0	Not indicated
0x1	Prefer LtRt downmix
0x2	Prefer LoRo downmix
0x3	PLII

5.12.3.12 **ltrtcmixlev**

This field, if present, specifies how much attenuation should be applied to the center channel when creating an LtRt downmix. It takes values shown in Table 25.

Table 25 - Encoder Configuration LtRtcmixlev

ltrtcmixlev	Meaning
0x0	+3.0 dB
0x1	+1.5 dB

0x2	0.0 dB
0x3	-1.5 dB
0x4	-3.0 dB
0x5	-4.5 dB
0x6	-6.0 dB
0x7	-∞ dB

5.12.3.13 **ltrtsurmixlev**

This field, if present, specifies how much attenuation should be applied to the LtRt surround channel or channels when downmixing to LtRt. It takes values shown in Table 26.

Table 26 - Encoder Configuration LtRtsurmixlev

ltrtsurmixlev	Meaning
0x0	RESERVED
0x1	RESERVED
0x2	RESERVED
0x3	-1.5 dB
0x4	-3.0 dB
0x5	-4.5 dB
0x6	-6.0 dB
0x7	-∞ dB

5.12.3.14 **lorocmixlev**

This field, if present, specifies indicates how much attenuation should be applied to the center channel when creating an LoRo downmix. It takes values shown in Table 25.

5.12.3.15 **lorosurmixlev**

This field, if present, specifies how much attenuation should be applied to the LtRt surround channel or channels when downmixing to LoRo. It takes values shown in Table 26.

5.12.3.16 **drc_exists**

This field specifies whether the payload contains extended DRC information for more advanced delivery codecs. If it does, it occurs immediately after this bit.

5.12.3.17 **drc_port_spkr**

This field takes values as shown in Table 18.

5.12.3.18 **drc_port_hphon**

This field takes values as shown in Table 18.

5.12.3.19 **drc_flat_panl**

This field takes values as shown in Table 18.

5.12.3.20 **drc_home_thtr**

This field takes values as shown in Table 18.

5.12.3.21 **drc_ddplus**

This field takes values as shown in Table 18.

5.12.3.22 **audio_presentation_id**

This field specifies an audio presentation identifier to which this set of E-AC-3 encoder parameters applies. A set of E-AC-3 encoder parameters may apply to multiple audio presentations and multiple audio presentation identifiers may be present. Allowed values are 1-511, the value 0 indicates that the list has ended.

5.13 Identity And Timing (IAT)

5.13.1 Identity and Timing General

The Identity and Timing payload describes additional information that uniquely identifies a program and provides timing information. The payload can also carry generic user data.

5.13.2 Identity And Timing Payload Syntax

Syntax	No. of Bits
Identity_and_timing_payload() {	
version	2
If (version == 0x3) {	
version += **extended_version**	4
}	
b_content_id	1
if (b_content_id == 1) {	
content_id_type	5
content_id_size_minus1	5
content_id	(content_id_size_minus1+1)*8
}	
b_distribution_id	1
if (b_distribution_id == 1) {	
distribution_id_type	3
distribution_id_size_minus1	4
distribution_id	(distribution_id_size_minus1+1)*8
}	
timestamp	35
b_offset	1
if (b_offset == 1) {	
offset	11
}	
b_validity_duration	1
if (b_validity_duration == 1) {	
validity_duration	11
}	
b_user_data	1
if (b_user_data == 1) {	

	user_data_size_minus1	8
	user_data	(user_data_size_minus1+1)*8
}		
b_extension		1
if (b_extension == 1) {		
	extension_size_minus1	8
	extension_data	(extension_data_size_minus1+1)*8
}		
byte_padding		0..7
}		

5.13.3 Identity and Timing Payload Fields

5.13.3.1 **version**

This 2-bit field shall indicate the version of the IAT message. For IAT messages that conform to this specification, the version field shall be set to '0', and the extended_version field shall not be present.

5.13.3.2 **extended_version**

Per Section 5.13.3.1, this field is not present.

5.13.3.3 **b_content_id**

This 1-bit boolean flag indicates the presence of the content ID fields. When the b_content_id flag is set to '1', the content_id_type, content_id_size, and content_id fields shall be present in the message. When the b_content_id flag is set to '0', the content_id_type, content_id_size, and content_id fields shall not be present in the message.

5.13.3.4 **content_id_type**

This 5-bit field specifies the type of content ID that the content_id field represents. The content_id_type field is coded according to Table 27.

Table 27 - IAT Content Id Type

content_id_type	Meaning
0x0	UUID
0x1	EIDR
0x2	Ad-ID
0x3 – 0x1e	RESERVED
0x1f	Unspecified

5.13.3.5 **content_id_size_minus1**

This field specifies the size of the content_id field that immediately follows this field, in number of bytes minus one. For example, a content_id_size_minus1 value of '11' indicates a content_id of 12 bytes.

5.13.3.6 **content_id**

This field specifies the content ID associated with the current content as specified by the content_id_type.

- For a UUID content ID type, the content_id field shall contain a 128-bit UUID as defined by IETF RFC 4122.

- For an EIDR content ID type, the content_id field shall contain a 96-bit identifier, formatted according to the Compact Binary Representation defined in Section 11.2 of SMPTE RP 2079-1, that represents an identifier registered with EIDR (http://eidr.org).
- For an Ad-ID content ID type, the content_id field shall contain an Ad-ID String, formatted according to SMPTE RP 2092-1, that represents an identifier registered with Ad-ID (http://www.ad-id.org/).

5.13.3.7 **b_distribution_id**

This 1-bit boolean flag indicates the presence of the Distribution Channel ID fields. When the b_distribution_id flag is set to '1', the distribution_id_size, and distribution_id fields shall be present in the message. When the b_distribution_id flag is set to '0', the distribution_id_size, and distribution_id fields shall not be present in the message.

5.13.3.8 **distribution_id_type**

This 3-bit field specifies the type of Distribution Channel ID that the distribution_id field represents. The distribution_id_type field is coded according to Table 28.

Table 28 - IAT Distribution Id Type

distribution_id_type	Meaning
0x0	ATSC 3.0
0x1 – 0x6	RESERVED
0x7	Unspecified type

5.13.3.9 **distribution_id**

This field specifies the Distribution Channel ID associated with the current content.

- For an ATSC 3.0 Distribution Channel ID type, the distribution_id field shall contain an ATSC 3.0 VP1 Channel ID as defined in ATSC A/336.

5.13.3.10 **timestamp**

This field specifies the time stamp value for the IAT payload. The timestamp is measured in ticks, where the duration of a tick is 1/240,000 of a second.

5.13.3.11 **b_offset**

This 1-bit Boolean flag indicates the presence of the offset field. When the b_offset flag is set to '1', the offset field shall be present in the message. When the b_offset flag is set to '0', the offset field shall not be present in the message.

5.13.3.12 **offset**

This field when present, specifies the number of samples into the future, relative to the indicated timestamp, where the validity of this IAT message begins.

5.13.3.13 **b_validity_duration**

This 1-bit Boolean flag indicates the presence of the validity_duration field. When the b_validity_duration flag is set to '1', the validity_duration field shall be present in the message. When the b_validity_duration flag is set to '0', the validity_duration field shall not be present in the message.

5.13.3.14 **validity_duration**

This field when present, specifies the number of samples into the future that the IAT payload is valid, relative to the timestamp and offset, if present.

5.13.3.15 b_user_data

This 1-bit Boolean flag indicates the presence of the user_data instance. When the b_user_data flag is set to '1', the user_data_size and user_data fields shall be present in the message. When the b_user_data flag is set to '0', the user_data_size and user_data fields shall not be present in the message.

5.13.3.16 user_data_size_minus1

This field specifies the size of the user_data field that immediately follows this field, in number of bytes minus one. For example, a user_data_size_minus1 value of '7' indicates a user_data size of 8 bytes.

5.13.3.17 user_data

Generic data field with a maximum size of user_data_size_minus1 + 1 bytes.

5.13.3.18 b_extension

This 1-bit Boolean flag indicates that extension data is present.

5.13.3.19 extension_size_minus1

This field specifies the size of the extension_data field that immediately follows this field, in number of bytes minus one. For example, an extension_data_size_minus1 value of '7' indicates an extension_data size of 8 bytes.

5.13.3.20 extension_data

Extension data field with a maximum size of extension_data_size_minus1 + 1 bytes. The content of the extension_data field is not defined in the present document.

5.14 PMD Version

5.14.1 PMD Version General

The PMD Version payload is used to track the implementation version of the PMD bitstream..

5.14.2 PMD Version Payload Syntax

Syntax	No. of Bits
pmd_version_payload() {	
major_version	8
minor_version	8
}	

5.14.3 PMD Version Payload Fields

5.14.3.1 major_version

This field specifies the major version number of the PMD bitstream.

5.14.3.2 minor_version

This field specifies the minor version number of the PMD bitstream..

Appendix A

SMPTE Universal Labels for PMD Payloads

Each PMD payload is uniquely identified with a SMPTE Universal Label (UL). The assigned ULs for PMD payloads are as shown in Table 29.

Table 29 - SMPTE UL Assignments for PMD

SMPTE UL	Payload Name
06.0E.2B.34.04.01.01.0D.04.04.02.04.00.00.00.00	PMD Version
06.0E.2B.34.04.01.01.0D.04.04.02.05.00.00.00.00	Audio Bed Description
06.0E.2B.34.04.01.01.0D.04.04.02.06.00.00.00.00	Audio Object Description
06.0E.2B.34.04.01.01.0D.04.04.02.07.00.00.00.00	Audio Presentation Description
06.0E.2B.34.04.01.01.0D.04.04.02.08.00.00.00.00	Audio Presentation Names
06.0E.2B.34.04.01.01.0D.04.04.02.09.00.00.00.00	Audio Element Names
06.0E.2B.34.04.01.01.0D.04.04.02.0A.00.00.00.00	ED2 Substream Description
06.0E.2B.34.04.01.01.0D.04.04.02.0B.00.00.00.00	ED2 Substream Names
06.0E.2B.34.04.01.01.0D.04.04.02.0C.00.00.00.00	EAC3 Encoding Parameters
06.0E.2B.34.04.01.01.0D.04.04.02.0D.00.00.00.00	Dynamic Position Update
06.0E.2B.34.04.01.01.0D.04.04.02.0E.00.00.00.00	Identity And Timing
06.0E.2B.34.04.01.01.0D.04.04.02.0F.00.00.00.00	Presentation Loudness Description
06.0E.2B.34.04.01.01.0D.04.04.02.10.00.00.00.00	ED2 Turnaround Description
06.0E.2B.34.04.01.01.0D.04.04.02.11.00.00.00.00	Headphone Element Description

EDITOR'S NOTE: UL values defined in this table are temporary and will be replaced by their final values prior to publication, at which point this note will be removed.

Bibliography (Informative)

Dolby E High-Level Frame Description (https://www.dolby.com/us/en/technologies/dolby-e-high-level-frame-description.pdf)

Dolby ED2 - Next Generation Audio Mezzanine Coder (https://www.dolby.com/us/en/technologies/dolby-atmos/Dolby-ED2.pdf)