INTERNATIONAL STANDARD

ISO/IEC 14496-3

Fourth edition
2009-09-01

AMENDMENT 4
2013-12-15

Information technology — Coding of audio-visual objects —

Part 3: **Audio**

AMENDMENT 4: New levels for AAC profiles

Technologies de l'information — Codage des objets audiovisuels —

Partie 3: Codage audio

AMENDEMENT 4: Nouveaux niveaux pour profils AAC

Reference number
ISO/IEC 14496-3:2009/Amd.4:2013(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

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 4 to ISO/IEC 14496-3:2009 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Coding of audio-visual objects —

Part 3:
Audio

AMENDMENT 4: New levels for AAC profiles

1 Notes from the editor

Some modifications with respect to ISO/IEC 14496-3:2009 are highlighted by colored background as follows. These highlights shall be removed upon integration of the amendment into a future edition of ISO/IEC 14496-3.

YELLOW Cross-references and subclause numbers that need to be checked and aligned with ISO/IEC 14496-3:2009.

GRAY Changes relative to ISO/IEC 14496-3:2009 that are highlighted for better visibility.

2 Changes to the text of ISO/IEC 14496-3:2009

In 1.2 Normative references, add:

ISO/IEC 23001-8, Information technology — MPEG systems technologies — Part 8: Coding-independent code points

In 1.3 Terms and Definitions, add:

1.3.z **SAOC-DE**: Spatial Audio Object Coding Dialogue Enhancement

and increase the index-number of subsequent entries.

In 1.5.1.1 Audio object type definition, amend Table 1.1 with the updates in the table below:

Object Type ID	Audio Object Type	gain control	[...]	Remark
0	Null			
[..]	[...]			
43	SAOC			
44	LD MPEG Surround			
45	SAOC-DE			
46 - 95	(reserved)			

After 1.5.1.2.39 add the following new subclauses:

1.5.1.2.40 SAOC-DE object type

The SAOC-DE object type conveys Spatial Audio Object Coding Dialogue Enhancement side information (see ISO/IEC 23003-2:2010/Amd.3) in the MPEG-4 Audio framework.

In 1.5.2.1 (Profiles), Table 1.3 (Audio Profiles definition), add:

Object Type ID	Audio Object Type	…
…	…	…
43	SAOC	
44	LD MPEG Surround	
45	SAOC-DE	

In 1.5.2.3 Levels within the profiles replace:

Table 1.10 – Levels for the AAC Profile

Level	Max. channels/ object	Max. sampling rate [kHz]	Max. PCU	Max. RCU
1	2	24	3	5
2	2	48	6	5
3	NA	NA	NA	NA
4	5	48	19	15
5	5	96	38	15

With:

Table 1.10 – Levels for the AAC Profile

Level	Max. channels/ object	Max. sampling rate [kHz]	Max. PCU	Max. RCU
1	2	24	3	5
2	2	48	6	5
3	NA	NA	NA	NA
4	5	48	19	15
5	5	96	38	15
6	7	48	25	19
7	7	96	50	19

Replace:

Table 1.11 – Levels for the High Efficiency AAC Profile

Level	Max. channels/ object	Max. AAC sampling rate, SBR not present [kHz]	Max. AAC sampling rate, SBR present [kHz]	Max. SBR sampling rate [kHz] (in/out)	Max. PCU	Max. RCU	Max. PCU Low power SBR	Max. RCU Low power SBR
1	NA	NA	NA	NA	NA	NA	NA	NA
2	2	48	24	24/48	9	10	7	8

3	2	48	48	48/48 (Note 1)	15	10	12	8
4	5	48	24/48 (Note 2)	48/48 (Note 1)	25	28	20	23
5	5	96	48	48/96	49	28	39	23

Note 1: For level 3 and level 4 decoders, it is mandatory to operate the SBR tool in downsampled mode if the sampling rate of the AAC core is higher than 24kHz. Hence, if the SBR tool operates on a 48kHz AAC signal, the internal sampling rate of the SBR tool will be 96kHz, however, the output signal will be downsampled by the SBR tool to 48kHz.

Note 2: For one or two channels the maximum AAC sampling rate, with SBR present, is 48kHz. For more than two channels the maximum AAC sampling rate, with SBR present, is 24kHz.

With:

Table 1.11 – Levels for the High Efficiency AAC Profile

Level	Max. channels/ object	Max. AAC sampling rate, SBR not present [kHz]	Max. AAC sampling rate, SBR present [kHz]	Max. SBR sampling rate [kHz] (in/out)	Max. PCU	Max. RCU	Max. PCU Low power SBR	Max. RCU Low power SBR
1	NA	NA	NA	NA	NA	NA	NA	NA
2	2	48	24	24/48	9	10	7	8
3	2	48	48	48/48 (Note 1)	15	10	12	8
4	5	48	24/48 (Note 2)	48/48 (Note 1)	25	28	20	23
5	5	96	48	48/96	49	28	39	23
6	7	48	24/48 (Note 2)	48/48 (Note 1)	34	37	27	30
7	7	96	48	48/96	67	37	53	30

Note 1: For level 3, level 4 and level 6 decoders, it is mandatory to operate the SBR tool in downsampled mode if the sampling rate of the AAC core is higher than 24kHz. Hence, if the SBR tool operates on a 48kHz AAC signal, the internal sampling rate of the SBR tool will be 96kHz, however, the output signal will be downsampled by the SBR tool to 48kHz.

Note 2: For one or two channels the maximum AAC sampling rate, with SBR present, is 48kHz. For more than two channels the maximum AAC sampling rate, with SBR present, is 24kHz.

Replace:

Table 1.12 – Levels for the High Efficiency AAC v2 Profile

Level (Note 1)	Max. channels/object	Max. AAC sampling rate, SBR not present [kHz]	Max. AAC sampling rate, SBR present [kHz]	Max. SBR sampling rate [kHz] (in/out)	Max. PCU	Max. RCU	Max. PCU HQ / LP SBR (Note 5)	Max. RCU HQ / LP SBR (Note 5)
1	NA	NA	NA	NA	NA	NA	NA	NA
2	2	48	24	24/48	9	10	9	10
3	2	48	24/48 (Note 3)	48/48 (Note 2)	15	10	15	10
4	5	48	24/48 (Note 4)	48/48 (Note 2)	25	28	20	23
5	5	96	48	48/96	49	28	39	23

Note 1: Level 2,3, and 4 HE AAC v2 Profile decoders implements the baseline version of the parametric stereo tool. A level 5 decoder shall not be limited to the baseline version of the parametric stereo tool.

Note 2: For level 3 and level 4 decoders, it is mandatory to operate the SBR tool in downsampled mode if the sampling rate of the AAC core is higher than 24kHz. Hence, if the SBR tool operates on a 48kHz AAC signal, the internal sampling rate of the SBR tool will be 96kHz, however, the output signal will be downsampled by the SBR tool to 48kHz.

Note 3: If Parametric Stereo data is present the maximum AAC sampling rate is 24kHz, if Parametric Stereo data is not present the maximum AAC sampling rate is 48kHz.

Note 4: For one or two channels the maximum AAC sampling rate, with SBR present, is 48kHz. For more than two channels the maximum AAC sampling rate, with SBR present, is 24kHz.

Note 5: The PCU/RCU number are given for a decoder operating the LP SBR tool whenever applicable.

With:

Table 1.12 – Levels for the High Efficiency AAC v2 Profile

Level (Note 1)	Max. channels/object	Max. AAC sampling rate, SBR not present [kHz]	Max. AAC sampling rate, SBR present [kHz]	Max. SBR sampling rate [kHz] (in/out)	Max. PCU	Max. RCU	Max. PCU HQ / LP SBR (Note 5)	Max. RCU HQ / LP SBR (Note 5)
1	NA	NA	NA	NA	NA	NA	NA	NA
2	2	48	24	24/48	9	10	9	10
3	2	48	24/48 (Note 3)	48/48 (Note 2)	15	10	15	10
4	5	48	24/48 (Note 4)	48/48 (Note 2)	25	28	20	23
5	5	96	48	48/96	49	28	39	23
6	7	48	24/48 (Note 4)	48/48 (Note 2)	34	37	27	30
7	7	96	48	48/96	67	37	53	30

Note 1: Level 2, 3, 4, 6 and 7 HE AAC v2 Profile decoders implement the baseline version of the parametric stereo tool. A level 5 decoder shall not be limited to the baseline version of the parametric stereo tool.

Note 2: For level 3, level 4 and level 6 decoders, it is mandatory to operate the SBR tool in downsampled mode if the sampling rate of the AAC core is higher than 24kHz. Hence, if the SBR tool operates on a 48kHz AAC signal, the internal sampling rate of the SBR tool will be 96kHz, however, the output signal will be downsampled by the SBR tool to 48kHz.

Note 3: If Parametric Stereo data is present the maximum AAC sampling rate is 24kHz, if Parametric Stereo data is not present the maximum AAC sampling rate is 48kHz.

Note 4: For one or two channels the maximum AAC sampling rate, with SBR present, is 48kHz. For more than two channels the maximum AAC sampling rate, with SBR present, is 24kHz.

Note 5: The PCU/RCU number are given for a decoder operating the LP SBR tool whenever applicable.

In 1.5.2.4 audioProfileLevelIndication

Insert the following new entries into Table 1.14 “audioProfileLevelIndication values” and adapt the “reserved for ISO use” range accordingly:

0x50	AAC Profile	L6
0x51	AAC Profile	L7
0x52	HE-AAC Profile	L6
0x53	HE-AAC Profile	L7
0x54	HE-AACv2 Profile	L6
0x55	HE-AACv2 Profile	L7
0x56	Extended HE-AAC Profile	L6
0x57	Extended HE-AAC Profile	L7
0x58 - 0x7F	reserved for ISO use	-

In 1.5.2.3 after:

"The NA (Not Applicable) levels are introduced to emphasize the hierarchical structure of the AAC Profile and the High Efficiency AAC Profile. Hence, a decoder supporting the High Efficiency AAC Profile at a given level can decode an AAC Profile stream of the same or a lower level. The NA levels are not indicated in the audioProfileLevelIndication table (Table 1.14).

Add:

NOTE: A Level 6 or 7 decoder is not required to decode a Level 5 stream."

In 1.5.2.3 "1.5.2.3 Levels within the profiles",in the AAC, HE-AAC and HE-AACv2 Profile definition after each occurrence of the sentence:

"For the audio object type 2 (AAC LC), mono or stereo mixdown elements are not permitted."

Add:

"For Levels 1 to 5 the height extension element is not permitted."

In 1.6.2.1 extend Table 1.15 "AudioSpecificConfig()"as follows:

Table 1.15 – Syntax of AudioSpecificConfig()

Syntax	No. of bits	Mnemonic
AudioSpecificConfig ()		
{		
...		
saocPresentFlag = -1;		
ldmpsPresentFlag = -1;		
saocDePresentFlag = -1;		
if (audioObjectType == 5 || audioObjectType == 29) {		
...		
case 43:		
saocPresentFlag = 1;		
saocPayloadEmbedding;	**1**	**uimsbf**
SaocSpecificConfig();		
break;		
case 44:		
ldmpsPresentFlag = 1;		
ldsacPayloadEmbedding;	**1**	**uimsbf**
LDSpatialSpecificConfig();		
break;		
case 45:		
saocDePresentFlag = 1;		
saocDePayloadEmbedding;	**1**	**uimsbf**
SaocDeSpecificConfig();		
break;		
default:		
/* reserved */		
}		
...		
if (extensionIdentifier == -1 && bits_to_decode() >= 11) {		
extensionIdentifier;	**11**	**bslbf**
}		
if (extensionIdentifier == 0x7cb) {		
extensionIdentifier = -1;		
if (audioObjectType != 43 && bits_to_decode() >= 1) {		
saocPresentFlag;	**1**	**uimsbf**
if (saocPresentFlag == 1) {		
saocPayloadEmbedding = 1;		
saocscLen;	**8**	**uimsbf**
if (saocscLen == 0xff) {		
saocscLenExt;	**16**	**uimsbf**
saocscLen += saocscLenExt;		
}		
SaocSpecificConfig();		
}		
}		

```
    }
    if (extensionIdentifier == -1 && bits_to_decode() >= 11 ) {
        extensionIdentifier;                                    11      bslbf
    }
    if ( extensionIdentifier == 0x7cc ) {
        extensionIdentifier = -1;
        if ( audioObjectType != 44 && bits_to_decode() >= 1 ) {
            ldmpsPresentFlag;                                   1       uimsbf
            if ( ldmpsPresentFlag == 1 ) {
                ldsacPayloadEmbedding = 1;
                ldsscLen;                                       8       uimsbf
                if ( ldsscLen == 0xff ) {
                    ldsscLenExt;                                16      uimsbf
                    ldsscLen += ldsscLenExt;
                }
                LDSpatialSpecificConfig();
            }
        }
    }
    if (extensionIdentifier == -1 && bits_to_decode() >= 11 ) {
        extensionIdentifier;                                    11      bslbf
    }
    if ( extensionIdentifier == 0x7cd ) {
        extensionIdentifier = -1;
        if ( audioObjectType != 45 && bits_to_decode() >= 1 ) {
            saocDePresentFlag;                                  1       uimsbf
            if ( saocDePresentFlag == 1 ) {
                saocDePayloadEmbedding = 1;
                saocDescLen;                                    8       uimsbf
                if ( saocDescLen == 0xff ) {
                    saocDescLenExt;                             16      uimsbf
                    saocDescLen += saocDescLenExt;
                }
                SaocDeSpecificConfig();
            }
        }
    }
}
```

After 1.6.2.1.19 add the new subclause as follows:

1.6.2.1.20 SaocDeSpecificConfig

Defined in ISO/IEC 23003-2:2010/Amd.3.

In 1.6.2.2.1 extend Table 1.17 "Audio Object Types" as follows:

Table 1.17 — Audio Object Types

Object Type ID	Audio Object Type	definition of elementary stream payloads and detailed syntax	Mapping of audio payloads to access units and elementary streams
0	NULL		
…			
43	SAOC	ISO/IEC 23003-2	
44	LD MPEG Surround	ISO/IEC 23003-2	
45	SAOC-DE	ISO/IEC 23003-2:2010/Amd.3	

In 1.6.3.5 channelConfiguration replace:

A four bit field indicating the audio output channel configuration:

Table 1.19 – Channel Configuration

value	number of channels	audio syntactic elements, listed in order received	channel to speaker mapping
0	-	-	defined in AOT related SpecificConfig
1	1	single_channel_element()	center front speaker
2	2	channel_pair_element()	left, right front speakers
3	3	single_channel_element(), channel_pair_element()	center front speaker, left, right front speakers
4	4	single_channel_element(), channel_pair_element(), single_channel_element()	center front speaker, left, right center front speakers, rear surround speakers
5	5	single_channel_element(), channel_pair_element(), channel_pair_element()	center front speaker, left, right front speakers, left surround, right surround rear speakers
6	5+1	single_channel_element(), channel_pair_element(), channel_pair_element(), lfe _element()	center front speaker, left, right front speakers, left surround, right surround rear speakers, front low frequency effects speaker
7	7+1	single_channel_element(), channel_pair_element(), channel_pair_element(), channel_pair_element(), lfe_element()	center front speaker left, right center front speakers, left, right outside front speakers, left surround, right surround rear speakers, front low frequency effects speaker
8-15	-	-	reserved

With:

A four bit field that carries a ChannelConfiguration value as defined in ISO/IEC 23001-8, “Coding Independent Code Points”. Table 1.19 lists the valid channel configurations for use in MPEG-4 Audio and documents the association of channel elements to speaker positions: the third column lists the audio syntactic elements in the order in which they occur in each access unit, and the fourth column lists the channel to speaker mapping where the first channel element listed maps to the first speaker listed. In the case of channel_pair_elements(), the first channel element that occurs within the channel pair maps to the next speaker listed. The columns of Table 1.19 that duplicate information from 23001-8 are informative, as 23001-8 is normative. The mapping of loudspeaker names to loudspeaker positions is as defined in by OutputChannelPosition in ISO/IEC 23001-8.

Table 1.19 – Channel Configuration

value	number of channels	audio syntactic elements, listed in order received	channel to speaker mapping
0	-	-	defined in AOT related SpecificConfig
1	1	single_channel_element()	center front speaker
2	2	channel_pair_element()	left, right front speakers
3	3	single_channel_element(), channel_pair_element()	center front speaker, left, right front speakers
4	4	single_channel_element(), channel_pair_element(), single_channel_element()	center front speaker, left, right front speakers, rear center speakers
5	5	single_channel_element(), channel_pair_element(), channel_pair_element()	center front speaker, left, right front speakers, left surround, right surround speakers
6	5.1	single_channel_element(), channel_pair_element(), channel_pair_element(), lfe _element()	center front speaker, left, right front speakers, left surround, right surround speakers, low frequency enhancement peaker
7	7.1 Front	single_channel_element(), channel_pair_element(), channel_pair_element(), channel_pair_element(), lfe_element()	center front speaker left, right front center speakers, left, right front speakers, left surround, right surround speakers, low frequency enhancement speaker
8-10	-	-	reserved
11	6.1	single_channel_element(), channel_pair_element(), channel_pair_element(), single_channel_element(), lfe_element()	center front speaker, left, right front speakers, left surround, right surround speakers, rear center speaker, low frequency enhancement speaker
12	7.1 Back	single_channel_element(), channel_pair_element(), channel_pair_element(), channel_pair_element(), lfe_element()	center front speaker left, right front speakers, left surround, right surround speakers, rear surround left, right speakers, low frequency enhancement speaker
13	22.2	single_channel_element(), channel_pair_element(), channel_pair_element(), channel_pair_element(), channel_pair_element(), single_channel_element(), lfe_element(), lfe_element(), single_channel_element(), channel_pair_element(), channel_pair_element(), single_channel_element(), channel_pair_element(), single_channel_element(), single_channel_element(), channel_pair_element()	center front speaker, left, right front center speakers, left, right front speakers, left, right side surround speakers, rear left, right surround speakers, rear center speaker, left front low freq. enhancement speaker, right front low freq. enhancement speaker, center front vertical height speaker, left, right front vertical height speakers, left, right vertical height side surr. speakers, top center surround speaker, left, right surr. vertical height rear speakers, center vertical height rear speaker, center front vertical bottom speaker, left, right front vertical bottom speakers

14	7.1 Top	single_channel_element(), channel_pair_element(), channel_pair_element(), lfe_element(), channel_pair_element()	center front speaker, left, right front speakers, left surround, right surround speakers, low frequency enhancement speaker, left, right front vertical height speakers
15	-	-	reserved

After 1.6.3.25 add the new text as follows:

1.6.3.26 saocDePayloadEmbedding

The audio Object Type ID 45 SAOC-DE is used to convey spatial audio object coding side information for SAOC-DE decoding as defined in ISO/IEC 23003-2:2010/Amd.3. Depending on this flag, the SAOC-DE data payload, i.e., SaocDeFrame(), is available by different means:

Table AMD4.13 – saocDePayloadEmbedding

saocDePayloadEmbedding	Meaning
0	One SaocDeFrame() is mapped into one access unit. Subsequent access units form one elementary stream. That elementary stream will always depend on another elementary stream that contains the underlying (downmixed) audio data.
1	The top level payload is multiplexed into the underlying (downmixed) audio data. The actual multiplexing details depend on the presentation of the audio data (i.e., usually on the AOT). Note that this leads to an elementary stream with no real payload. That elementary stream will always depend on another elementary stream that contains both, the underlying (downmixed) audio data and the multiplexed spatial audio data.

1.6.3.27 saocDePresentFlag

A one bit field indicating the presence or absence of SAOC-DE data. The value –1 indicates that the saocDePresentFlag was not conveyed in the AudioSpecificConfig().

1.6.3.28 saocDescLen

A helper variable indicating the number of bytes of the subsequent SaocSpecificConfig() data function including possible fill bits.

1.6.3.29 saocDescLenExt

A helper variable indicating the additional number of bytes of the subsequent SaocSpecificConfig() data function including possible fill bits.

After 4.4.1.1 Program config element, after Table 4.2 – Syntax of program_config_element(), add the following subclause:

4.4.1.2 Height extension element

Table AMD4.1 - Syntax of height extension element

Syntax	No. of Bits	Mnemonic
height_extension_element(comment_field_bytes) {		
PCE_HEIGHT_EXTENSION_SYNC;	**8**	**bslfb**
for (i = 0; i < num_front_channel_elements; i++){		
front_element_height_info[i];	**2**	**bslbf**
}		
for (i = 0; i < num_side_channel_elements; i++){		
side_element_height_info[i];	**2**	**bslbf**
}		
for (i = 0; i < num_back_channel_elements; i++){		
back_element_height_info[i];	**2**	**bslbf**
}		
byte_alignment();		
height_info_crc_check	**8**	**rpchof**
}		
Note 1: height_extension_element() may be embedded in comment_field_data[i] of program_config_element().		

The height extension element should not be used if it is possible to signal presence of height channels via channelConfiguration.

The height extension elements must not change the channel configuration as defined via the channelConfiguration element.

In 4.4.2.7 extend Table 4.57 "Syntax of extension_payload()" as follows:

Table 4.57 – Syntax of extension_payload()

Syntax	No. of bits	Mnemonic
extension_payload(cnt)		
{		
extension_type;	**4**	**uimsbf**
align = 4;		
switch(extension_type) {		
case EXT_DYNAMIC_RANGE:		
return dynamic_range_info();		
case EXT_SAC_DATA:		
return sac_extension_data(cnt);		
case EXT_SAOC_DATA:		
return saoc_extension_data(cnt);		
case EXT_LDSAC_DATA:		
return ldsac_extension_data(cnt);		
case EXT_SBR_DATA:		
return sbr_extension_data(id_aac, 0);		Note 1
case EXT_SBR_DATA_CRC:		
return sbr_extension_data(id_aac, 1);		Note 1

```
        case EXT_SAOC_DE_DATA:
            return saoc_de_extension_data(cnt);
        case EXT_DATA_LENGTH:
            …
```

In 4.4.2.7 after Table 4.61 add new Table AMD4.14 "saoc_de_extension_data()" as given below:

Table AMD4.14 – Syntax of saoc_de_extension_data()

Syntax	No. of bits	Mnemonic
saoc_de_extension_data(cnt)		
{		
ancType;	**2**	**uimsbf**
ancStart;	**1**	**uimsbf**
ancStop;	**1**	**uimsbf**
for (i=0; i<cnt-1; i++) {		
ancDataSegmentByte[i];	**8**	**bslbf**
}		
return (cnt);		
}		

After 4.5.1.2 Program config element (PCE),add the following subclause:

4.5.1.3 Height extension element

4.5.1.3.1 Definitions

4.5.1.3.1.1 Data elements

PCE_HEIGHT_EXTENSION_SYNC 0xac

front_element_height_info a two-bit field containing height information

side_element_height_info a two-bit field containing height information

back_element_height_info a two-bit field containing height information

Table AMD4.2 - Definition of height information

front_element_height_info **side_element_height_info** **back_element_height_info**	**height information**
0	Normal height
1	Top speaker
2	Bottom speaker
3	Reserved

If no height extension element is transmitted, height information is set to "Normal height".

In the PCE the order of channel elements within the front, side and back channel elements shall follow the order as implied by Table AMD4.2 i.e. normal height elements first followed by Top and Bottom height speakers.

height_info_crc_check 8 bit CRC error detection data generated as described in 7.3.4.6. of ISO/IEC23003-1, CRC Polynom: G(X) = X8 + X2 + X + 1, with initial shift register state 0xFF.

4.5.1.3.1.2 Help elements

crc_calc() Function that returns CRC value calculated by data of all front_element_height_info, side_element_height_info and back_element_height_info.

The following bits are protected and fed into the CRC algorithm in order of their appearance:

1) 8-bit height info sync word
2) front_element_height_info
3) side_element_height_info
4) back_element_height_info
5) byte_alignment bits

NOTE: Channel configurations containing other than normal height speakers should be signaled through explicit channel configurations as defined in Table 1.19, if possible.

In 4.4.2.1 Payloads for the audio object types AAC main, AAC SSR, AAC LC and AAC LTP, after Table 4.10 – Syntax of data_stream_element(), insert the following tables:

Table AMD4.3 - Syntax of MPEG4_ancillary_data

Syntax	No. of Bits	Mnemonic
MPEG4_ancillary_data() {		
ancillary_data_sync;	8	**bslbf**
mpeg_audio_type	**2**	**bslbf**
dolby_surround_mode	**2**	**bslbf**
drc_presentation_mode	**2**	**bslbf**
stereo_downmix_mode;	**1**	**bslbf**
bs_info_reserved, set to “0”	**1**	**bslbf**
anc_data_status_reserved, set to “000”	**3**	**bslbf**
downmixing_levels_MPEG4_status	**1**	**bslbf**
ancillary_data_extension_status;	**1**	**bslbf**
audio_coding_and_compression_status	**1**	**bslbf**
coarse_grain_timecode_status	**1**	**bslbf**
fine_grain_timecode_status	**1**	**bslbf**

if (downmixing_levels_MPEG4_status == 1) {		
center_mix_level_on	**1**	**bslbf**
center_mix_level_value	**3**	**bslbf**
surround_mix_level_on	**1**	**bslbf**
surround_mix_level_value	**3**	**bslbf**
}		
If (audio_coding_mode_and_compression_status == 1) {		
audio_coding_mode_reserved	**7**	**bslbf**
compression_on	**1**	**bslbf**
compression_value	**8**	**bslbf**
}		
if(coarse_grain_timecode_status == 1)		
coarse_grain_timecode	**16**	**bslbf**
if(fine_grain_timecode_status == 1)		
fine_grain_timecode	**16**	**bslbf**
if (ancillary_data_extension_status == 1){		
reserved, set to "0"	**1**	**bslbf**
ext_downmixing_levels_status;	**1**	**bslbf**
ext_downmixing_global_gains_status;	**1**	**bslbf**
ext_downmixing_lfe_level_status;	**1**	**bslbf**
reserved, set to "0000"	**4**	**bslbf**
if (ext_downmixing_levels_status == 1){		
dmix_a_idx;	**3**	**bslbf**
dmix_b_idx;	**3**	**bslbf**
reserved, set to "00"	**2**	**bslbf**
}		
if (ext_downmixing_global_gains_status == 1) {		
dmx_gain_5_sign;	**1**	**bslbf**
dmx_gain_5_idx;	**6**	**bslbf**
reserved, set to "0"	**1**	**bslbf**
dmx_gain_2_sign;	**1**	**bslbf**
dmx_gain_2_idx;	**6**	**bslbf**
reserved, set to "0"	**1**	**bslbf**
}		
if (ext_downmixing_lfe_level_status == 1) {		
dmix_lfe_idx;	**4**	**bslbf**
reserved, set to "0000"	**4**	**bslbf**
}		
}		
}		

In 4.5.2.1.1 after::

data_stream_element()	Abbreviation DSE. Syntactic element that contains data. Again, there are 16 element_instance_tags. There is, however, no restriction on the number of data_stream_element()'s with any one instance tag, as a single data stream may continue across multiple data_stream_element()'s with the same instance tag.

Add:

The DSE may contain a MPEG4_ancillary_data element as described in Table AMD4.3.

In 4.5.2.1.4 Data stream element (DSE) after:

See ISO/IEC13818-7 (13818-7:2005, subclause 8.6 "Data Stream Element (DSE)").

Add:

A DSE containing a MPEG4_ancillary_data element shall have the same **element_instance_tag** as the first channel element it is associated to. It should follow after the last channel or SBR element it is associated to.

In 4.5.2.9.3 extend Table 4.121 "Values of the extension_type field" as follows:

Table 4.121 – Values of the extension_type field

Symbol	Value of extension_type	Purpose
EXT_FILL	'0000'	bitstream payload filler
EXT_FILL_DATA	'0001'	bitstream payload data as filler
EXT_DATA_ELEMENT	'0010'	data element
EXT_DATA_LENGTH	'0011'	container with explicit length for extension_payload()
EXT_LDSAC_DATA	'1001'	LD MPEG Surround
EXT_SAOC_DATA	'1010'	SAOC
EXT_DYNAMIC_RANGE	'1011'	dynamic range control
EXT_SAC_DATA	'1100'	MPEG Surround
EXT_SBR_DATA	'1101'	SBR enhancement
EXT_SBR_DATA_CRC	'1110'	SBR enhancement with CRC
EXT_SAOC_DE_DATA	'1111'	SAOC-DE
-	all other values	Reserved: These values can be used for a further extension of the syntax in a compatible way.
Note: Extension payloads of the type EXT_FILL or EXT_FILL_DATA have to be added to the bitstream payload if the total bits for all audio data together with all additional data are lower than the minimum allowed number of bits in this frame necessary to reach the target bitrate. Those extension payloads are avoided under normal conditions and free bits are used to fill up the bit reservoir. Those extension payloads are written only if the bit reservoir is full.		

In 4.5.2. "Decoding of the GA bitstream payloads", after 4.5.2.13, insert the following subclause:

4.5.2.14 MPEG4 ancillary data

4.5.2.14.1 Definitions

4.5.2.14.1.1 Data elements

ancillary_data_sync Indicates the presence of MPEG-4 Audio ancillary data and shall be set to 0xBC

mpeg_audio_type See Table AMD4.4

Table AMD4.4 - Definition of mpeg_audio_type

mpeg_audio_type	Description
“00”	Reserved
“01”	Reserved
“10”	Reserved
“11”	MPEG4 Audio data

dolby_surround_mode indicates whether a 2-ch stereo audio signal has been encoded in Dolby Surround. This field shall be set to “00” if the audio stream is not 2-channel stereo.

Table AMD4.5 - Definition of dolby_surround_mode

dolby_surround_mode	Description
“00”	Dolby surround mode not indicated
“01”	2-ch audio part is not Dolby surround encoded
“10”	2-ch audio part is Dolby surround encoded
“11”	Reserved

drc_presentation_mode Indicates whether **dynamic_range_info()** as defined in 4.5.2.7 shall be prioritized over dynamic range control transmitted in the **compression_value** field. The DRC presentation mode should not be changed within one elementary stream.

Table AMD4.6 - Definition of drc_presentation_mode

drc_presentation_mode	Description
"00"	DRC presentation mode not indicated
"01"	DRC presentation mode 1
"10"	DRC presentation mode 2
"11"	Reserved

stereo_downmix_mode indicates a downmixing procedure

Table AMD4.7 - Definition of downmix procedure

stereo_downmix_mode	downmix procedure
0	Lo/Ro
1	Lt/Rt

downmixing_levels_MPEG4_status indicates if downmixing_levels_MPEG4 exists

ancillary_data_extension_status indicates if MPEG4_ext_ancillary_data() exists

audio_coding_mode_and_compression_status indicates if the associated fields exist

coarse_grain_timecode_status indicates if coarse grain timecode information exists

fine_grain_timecode_status	indicates if coarse grain timecode information exists
center_mix_level_on	Indicates if the center_mix_level_value field carries information. If this field is set to "1" the center_mix_value field shall indicate the matrix mix down level of the center channel with respect to the left and right front channels. If this field is set to “0”, the center_mix_level_value field shall be set to “000”.
center_mix_level_value	indicates the multiplication factor for center channel if creating a 2-channel stereo downmix. See 4.5.2.1.12.2.1 for details.
surround_mix_level_on	Indicates if the surround_mix_level_value field carries information. If this field is set to "1" the surround_mix_value shall indicate the matrix mix-down level of the surround channels with respect to the left and right front channels. If this field is set to “0”, the surround_mix_level_value field shall be set to “000”.
surround_mix_level_value	indicates the multiplication factor for surround channels if creating a 2-channel stereo downmix. See 4.5.2.1.12.2.1 for details.
compression_on	indicates if the compression_value field carries information. If this field is set to "1" the compression_value field indicates the heavy compression factor. If this field is set to “0”, the compression_value field shall be set to “00000000”.
compression_value	This field consists of a value X in the four msb's and a value Y in the four lsb's. The actual compression value is 48.164 - 6.0206 × X - 0.4014 × Y dB. The compression_value field indicates a heavy compression factor which may be applied instead of light compression data in dynamic_range_info() as defined in 4.5.2.7.
coarse_grain_timecode	carries coarse grain time code information
fine_grain_timecode	carries fine grain time code information
ext_downmixing_levels_status	indicates if ext_downmixing_levels() exists
ext_downmixing_global_gains_status	indicates if ext_downmixing_global_gains() exists
ext_downmixing_lfe_level_status	indicates if ext_downmixing_lfe_level() exists
dmix_a_idx	indicates an index for the generation of a 5-channel downmix as shown in Tables AMD4.8 and AMD4.10
dmix_b_idx	indicates an index for the generation of a 5-channel downmix as shown in Tables AMD4.8 and AMD4.10
dmix_lfe_idx	indicates an index for LFE handling during downmixing as shown in Table AMD4.9
dmx_gain_5_sign	Downmix loudness compensation gain sign information. One bit indicating the sign of the global downmix gain factor for a 5-channel downmix. (0 if positive, 1 if negative)

dmx_gain_5_idx	Downmix loudness compensation gain magnitude information for a 5-channel downmix
dmx_gain_2_sign	Downmix loudness compensation gain sign information. One bit indicating the sign of the global downmix gain factor for a 2-ch stereo downmix
dmx_gain_2_idx	Downmix loudness compensation gain magnitude information for a 2-ch stereo downmix

dmx_gain_5_idx and **dmx_gain_2_idx** are quantized in 0.25 dB steps using a 6-bit unsigned integer, and therefore, in association with **dmx_gain_5_sign** and **dmx_gain_2_sign,** have a range of +/-15.75 dB. They are interpreted as a gain value that shall be applied to the decoded audio output samples of all channels including LFE channels of the current frame.

Downmix indices should be present when multi-channel audio is transmitted, i.e. for 5-channel audio, **center_mix_level_value** and **surround_mix_level_value,** and for 7-channel audio in addition **dmix_a_idx** and **dmix_b_idx** are recommended.

4.5.2.14.1.2

MPEG4 ancillary data as defined in this subclause shall be placed into a single data_stream_element() as defined in table 4.10. The element_instance_tag of this data_stream_element() shall have the same value as the element_instance_tag of the first audio element to which the ancillary data applies.

4.5.2.14.2 Decoding process

Speaker abbreviations follow outputChannelPosition as defined in ISO/IEC 23001-8.

The decoder may use the **ancillary_data_sync** and **mpeg_audio_type** fields to verify the presence of MPEG-4 ancillary data.

The decoder should parse the **dolby_surround_mode** field and provide the information to the framework.

It is strongly recommended that the decoder parses the **drc_presentation_mode** field and applies the information as specified in 4.5.2.14.2.4.

If a 2-channel stereo down-mix is to be derived from a 5.0- or 5.1-channel audio signal, it is strongly recommended that the decoder applies the fields **center_mix_level_on**, **center_mix_level_value**, **surround_mix_level_on**, **surround_mix_level_value**, **dmix_lfe_idx** and **pseudo_surround_enable** as specified in 4.5.2.14.2.1. It is strongly recommended that the downmixing levels in these fields have priority over the matrix mix-down method specified in 4.5.1.2.2.

If a 5.0- or 5.1-channel down-mix is to be derived, it is strongly recommended that the decoder parses and applies the fields **dmix_a_idx**, **dmix_b_idx**.as specified in 4.5.2.14.2.2.

If a 2-channel stereo downmix is to be derived from a 6.1 or 7.1 stream, it is strongly recommended that this is generated by downmixing the 5.0- or 5.1-channel down-mix to 2-channel stereo, i.e. it is strongly recommended that the downmixing procedures for 5 and 2 channels as specified in 4.5.2.14.2.2 and 4.5.2.14.2.1 are applied subsequently. It is strongly recommended that the global downmix gains as specified in 4.5.2.14.2.3 are used to compensate for loudness deviations after downmixing.

It is strongly recommended that the decoder parses the **compression_on** and **compression_value** fields. If **compression_on** is set to “1”, the decoder shall apply these values as required according to the DRC Presentation mode specified in 4.5.2.14.2.4.

4.5.2.14.2.1 Downmixing from 5.1 to Stereo

if **stereo_downmix_mode** is 0,

L' = L + C×b + Ls×a + LFE×c

R' = R + C×b + Rs×a + LFE×c

else if **stereo_downmix_mode** is 1,

L' = L + C×b - (Ls+Rs)×a + LFE×c

R' = R+ C×b + (Ls+Rs)×a +LFE×c

where **surround_mix_level**, "a" and **center_mix_level**, "b" are shown as "Multiplication factor" in Table AMD4.8. C, L, R, Ls, Rs are the source signals and L' and R' are the derived stereo signals. LFE channels should be omitted from the mixdown (i.e. c is equal to zero) if **ext_downmixing_lfe_level_status** is "0". If **ext_downmixing_lfe_level_status** is "1", the LFE mix level "c" shall be derived as shown in Table AMD4.9.

Table AMD4.8 - Definition of mix level value

dmix_a_idx, dmix_b_idx surround_mix_level, "a" center_mix_level, "b"	Multiplication factor
000	1.0 (0dB)
001	0.841 (-1.5dB)
010	0.707 (-3dB)
011	0.596 (-4.5dB)
100	0.500 (-6dB)
101	0.422 (-7.5dB)
110	0.355 (-9dB)
111	0.000 (-•dB)

Table AMD4.9 - Definition of LFE mix level value

dmix_lfe_idx	Multiplication factor
0000	3.162 (+10dB)
0001	2.000 (+6dB)
0010	1.679 (+4.5dB)
0011	1.413 (+3 dB)
0100	1.189 (+1.5dB)
0101	1.0 (0dB)
0110	0.841 (-1.5dB)
0111	0.707 (-3dB)
1000	0.596 (-4.5dB)
1001	0.500 (-6dB)
1010	0.316 (-10 dB)
1011	0.178 (-15dB)
1100	0.100 (-20dB)
1101	0.032 (-30dB)
1110	0.010 (-40dB)
1111	0.000 (-•dB)

4.5.2.14.2.2 Downmixing from 6.1/7.1 to 5.1

Table AMD4.10 - Definition of downmixing_levels

Channel Configuration	dmix_a_idx	dmix_b_idx
7.1 Back, 6.1	d1	d2
7.1 Front	e1	e2
7.1 Top	f1	f2

4.5.2.14.2.2.1 Downmixing from 7.1 Back to 5.1

C' = C

L' = L

R' = R

Ls' = Ls × d1+ Lsr × d2

Rs' = Rs × d1 + Rsr' × d2

LFE' = LFE

where

surround mix level, "d1" and rear surround mix level, "d2" are shown in Table AMD4.10.

C, L, R, Ls, Rs, Lsr, Rsr, LFE are the source signals and C', L', R', Ls', Rs', LFE' are the derived 5.1 channel signals.

4.5.2.14.2.2.2 Downmixing from 7.1 Front to 5.1

C' = C + (Lc+Rc) × e1

L' = L + Lc × e2

R'= R + Rc × e2

Ls' = Ls

Rs' = Rs

LFE' = LFE

where

center front mix level, "e1" and "e2" are shown in Table AMD4.10.

C, L, R, Lc, Rc ,Ls, Rs, LFE are the source signals and C', L', R', Ls', Rs', LFE' are the derived 5.1 channel signals.