INTERNATIONAL STANDARD

ISO/IEC 14496-3

Third edition
2005-12-01

AMENDMENT 2
2006-03-15

Information technology — Coding of audio-visual objects —

Part 3:
Audio

AMENDMENT 2: Audio Lossless Coding (ALS), new audio profiles and BSAC extensions

Technologies de l'information — Codage des objets audiovisuels —
Partie 3: Codage audio

AMENDEMENT 2: Codage audio sans perte (ALS), nouveaux profils audio et extensions BSAC

Reference number
ISO/IEC 14496-3:2005/Amd.2:2006(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 2 to ISO/IEC 14496-3:2005 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This amendment specifies the Audio Lossless Coding (ALS) scheme. The amendment further defines a new profile, the High Efficiency AAC v2 Profile, that incorporates all the features of the High Efficiency AAC Profile and in addition the Parametric Stereo tool. The amendment also specifies the way in which the audio object type ER BSAC is extended to support multi-channel format, providing backward compatibility.

Information technology — Coding of audio-visual objects —

Part 3:
Audio

AMENDMENT 2: Audio Lossless Coding (ALS), new audio profiles and BSAC extensions

In the Introduction, at the end of subclause "Lossless Audio Coding Tools", add:

MPEG-4 ALS (Audio Lossless Coding) provides lossless coding of digital audio signals. Input signals can be integer PCM data with 8 to 32-bit word length or 32-bit IEEE floating-point data. Up to 65536 channels are supported.

In Part 3: Audio, Subpart 1, in subclause 1.3 Terms and Definitions, add:

ALS: Audio Lossless Coding

and increase the index-number of subsequent entries.

In Part 3: Audio, Subpart 1, in subclause 1.5.1.1 Audio object type definition, replace table 1.1 with the table below:

Table 1.1 — Audio Object Type definition based on Tools/Modules

Object Type ID	Audio Object Type	gain control	block switching	window shapes - standard	window shapes – AAC LD	filterbank - standard	filterbank - SSR	TNS	LTP	intensity	coupling	frequency deomain prediction	PNS	MS	SIAQ	FSS	upsampling filter tool	quantisation&coding - AAC	quantisation&coding – TwinVQ	quantisation&coding - BSAC	AAC ER Tools	ER payload syntax	EP Tool 1)	CELP	Silence Compression	HVXC	HVXC 4kbit/s VR	SA tools	SASBF	MIDI	HILN	TTSI	SBR	Layer-1	Layer-2	Layer-3	SSC (Transient, Sinusoid, Noise)	Parametric stereo	DST	ALS	Remark
0	Null																																								
1	AAC main		X	X		X		X		X	X	X	X	X				X																							2)
2	AAC LC		X	X		X		X		X	X		X	X				X																							2)
3	AAC SSR	X	X	X			X	X		X	X		X	X				X																							
4	AAC LTP		X	X		X		X	X	X	X		X	X				X																							2)
5	SBR																																X								
6	AAC Scalable		X	X		X		X	X	X			X	X	X	X	X	X																							6)
7	TwinVQ		X	X		X		X	X					X					X																						
8	CELP																							X																	
9	HVXC																									X															
10	(reserved)																																								
11	(reserved)																																								
12	TTSI																															X									
13	Main synthetic																											X	X	X											3)
14	Wavetable synthesis																												X	X											4)
15	General MIDI																													X											
16	Algorithmic Synthesis and Audio FX																											X													
17	ER AAC LC		X	X		X		X		X			X	X				X			X	X	X																	
18	(reserved)																																								
19	ER AAC LTP		X	X		X		X	X	X			X	X				X			X	X	X																	5)
20	ER AAC scalable		X	X		X		X		X			X	X	X	X	X	X			X	X	X																	6)
21	ER TwinVQ		X	X		X		X						X					X			X	X																	
22	ER BSAC		X	X		X		X		X			X	X						X		X	X																	
23	ER AAC LD				X	X		X	X	X			X	X				X			X	X	X																	
24	ER CELP																					X	X	X	X															
25	ER HVXC																					X	X			X	X													
26	ER HILN																					X	X								X									
27	ER Parametric																					X	X			X	X				X									
28	SSC																																				X	X			
29	PS																																X					X			
30	(reserved)																																								
31	(escape)																																								
32	Layer-1																																	X							
33	Layer-2																																		X						
34	Layer-3																																			X					
35	DST																																					X			
36	ALS																																							X	
37 - 95	(reserved)																																								

In Part 3: Audio, Subpart 1, in subclause 1.5.1.2 Description, add:

1.5.1.2.30 ALS object type

The ALS object type is the counterpart of the Audio Lossless Coding (ALS) scheme and contains the corresponding ALS tools.

In Part 3: Audio, Subpart 1, replace Table 1.3 (Audio Profiles definition) with the following table:

Table 1.3 – Audio Profiles definition

Object Type ID	Audio Object Type	Main Audio Profile	Scalable Audio Profile	Speech Audio Profile	Synthetic Audio Profile	High Quality Audio Profile	Low Delay Audio Profile	Natural Audio Profile	Mobile Audio Internet-working Profile	AAC Profile	High Efficiency AAC Profile	High Efficiency AAC v2 Profile
0	Null											
1	AAC main	X						X				
2	AAC LC	X	X			X		X		X	X	X
3	AAC SSR	X						X				
4	AAC LTP	X	X			X		X				
5	SBR										X	X
6	AAC Scalable	X	X			X		X				
7	TwinVQ	X	X					X				
8	CELP	X	X	X		X	X	X				
9	HVXC	X	X	X			X	X				
10	(reserved)											
11	(reserved)											
12	TTSI	X	X	X	X		X	X				
13	Main synthetic	X			X							
14	Wavetable synthesis											
15	General MIDI											
16	Algorithmic Synthesis and Audio FX											
17	ER AAC LC					X		X	X			
18	(reserved)											
19	ER AAC LTP					X		X				
20	ER AAC Scalable					X		X	X			
21	ER TwinVQ							X	X			
22	ER BSAC							X	X			
23	ER AAC LD						X	X	X			
24	ER CELP					X	X	X				
25	ER HVXC						X	X				
26	ER HILN							X				
27	ER Parametric							X				
28	SSC											
29	PS											X
30	(reserved)											
31	(escape)											
32	Layer-1											
33	Layer-2											
34	Layer-3											
35	DST											
36	ALS											

In Part 3: Audio, Subpart 1, subclause 1.5.2.3 (Levels within the profiles), add at the end:

- **Levels for the High Efficiency AAC v2 Profile**

Table 1.11A - Levels for the High Efficiency AAC v2 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 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 (Note 1)	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: A level 2 HE AAC v2 Profile decoder implements the baseline version of the parametric stereo tool. Higher level decoders 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.

A HE AAC v2 Profile decoder of a certain level shall operate the HQ SBR tool for streams containing Parametric Stereo data. For streams not containing Parametric Stereo data, the HE AAC v2 Profile decoder may operate the HQ SBR tool, or the LP SBR tool.

In Part 3: Audio, Subpart 1, subclause 1.5.2.4 (Table 1.12 - audioProfileLevelIndication Values), replace the row:

0x30-0x7F	reserved for ISO use	-

with:

0x28	AAC Profile	L1
0x29	AAC Profile	L2
0x2A	AAC Profile	L4
0x2B	AAC Profile	L5
0x2C	High Efficiency AAC Profile	L2
0x2D	High Efficiency AAC Profile	L3
0x2E	High Efficiency AAC Profile	L4
0x2F	High Efficiency AAC Profile	L5
0x30	High Efficiency AAC v2 Profile	L2
0x31	High Efficiency AAC v2 Profile	L3
0x32	High Efficiency AAC v2 Profile	L4
0x33	High Efficiency AAC v2 Profile	L5
0x34-0x7F	reserved for ISO use	-

In Part 3: Audio, Subpart 1, in subclause 1.6.2.1 AudioSpecificConfig, replace table 1.13 with the table below:

Table 1.13 — Syntax of AudioSpecificConfig()

Syntax	No. of bits	Mnemonic
AudioSpecificConfig ()		
{		
audioObjectType = GetAudioObjectType();		
samplingFrequencyIndex;	**4**	**bslbf**
if (samplingFrequencyIndex == 0xf) {		
samplingFrequency;	**24**	**uimsbf**
}		
channelConfiguration;	**4**	**bslbf**
sbrPresentFlag = -1;		
psPresentFlag = -1;		
if (audioObjectType == 5 ||		
audioObjectType == 29) {		
extensionAudioObjectType = 5;		
sbrPresentFlag = 1;		
if (audioObjectType == 29) {		
psPresentFlag = 1;		
}		
extensionSamplingFrequencyIndex;	**4**	**uimsbf**
if (extensionSamplingFrequencyIndex == 0xf) {		
extensionSamplingFrequency;	**24**	**uimsbf**
}		
audioObjectType = GetAudioObjectType();		
}		
else {		
extensionAudioObjectType = 0;		
}		
switch (audioObjectType) {		
case 1:		
case 2:		
case 3:		
case 4:		
case 6:		
case 7:		
case 17:		
case 19:		
case 20:		
case 21:		
case 22:		
case 23:		
GASpecificConfig();		
break:		
case 8:		
CelpSpecificConfig();		
break;		
case 9:		
HvxcSpecificConfig();		
break:		
case 12:		
TTSSpecificConfig();		
break;		

```
case 13:
case 14:
case 15:
case 16:
    StructuredAudioSpecificConfig();
    break;
case 24:
    ErrorResilientCelpSpecificConfig();
    break;
case 25:
    ErrorResilientHvxcSpecificConfig();
    break;
case 26:
case 27:
    ParametricSpecificConfig();
    break;
case 28:
    SSCSpecificConfig();
    break;
case 32:
case 33:
case 34:
    MPEG_1_2_SpecificConfig();
    break;
case 35:
    DSTSpecificConfig();
    break;
case 36:
    ALSSpecificConfig();
    break;
default:
    /* reserved */
}
switch (audioObjectType) {
case 17:
case 19:
case 20:
case 21:
case 22:
case 23:
case 24:
case 25:
case 26:
case 27:
    epConfig;                                          2          bslbf
    if ( epConfig == 2 || epConfig == 3 ) {
        ErrorProtectionSpecificConfig();
    }
    if ( epConfig == 3 ) {
        directMapping;                                 1          bslbf
        if ( ! directMapping ) {
            /* tbd */
        }
    }
}
if ( extensionAudioObjectType != 5 && bits_to_decode() >= 16 ) {
    syncExtensionType;                                 11         bslbf
    if (syncExtensionType == 0x2b7) {
    extensionAudioObjectType = GetAudioObjectType();
```

```
            if ( extensionAudioObjectType == 5 ) {
                sbrPresentFlag;                                  1          uimsbf
                if (sbrPresentFlag == 1) {
                    extensionSamplingFrequencyIndex;             4          uimsbf
                    if ( extensionSamplingFrequencyIndex == 0xf ) {
                        extensionSamplingFrequency;              24         uimsbf
                    }
                    if ( bits_to_decode() >= 12 ) {
                        syncExtensionType;                       11         bslbf
                        if (syncExtensionType == 0x548) {
                            psPresentFlag;                       1          uimsbf
                        }
                    }
                }
            }
        }
    }
}
```

In Part 3: Audio, Subpart 1, in subclause 1.6.2.1 AudioSpecificConfig, add:

1.6.2.1.12 ALSSpecificConfig

Defined in ISO/IEC 14496-3 subpart 11.

In Part 3: Audio, Subpart 1, in subclause 1.6.2.1 Overview, replace table 1.15 by the following table:

Table 1.15 – Audio Object Types

Audio Object Type	Object Type ID	definition of elementary stream payloads and detailed syntax	Mapping of audio payloads to access units and elementary streams
AAC MAIN	1	ISO/IEC 14496-3 subpart 4	see subclause 1.6.2.2.2.1.2
AAC LC	2	ISO/IEC 14496-3 subpart 4	see subclause 1.6.2.2.2.1.2
AAC SSR	3	ISO/IEC 14496-3 subpart 4	see subclause 1.6.2.2.2.1.2
AAC LTP	4	ISO/IEC 14496-3 subpart 4	see subclause 1.6.2.2.2.1.2
SBR	5	ISO/IEC 14496-3 subpart 4	
AAC scalable	6	ISO/IEC 14496-3 subpart 4	see subclause 1.6.2.2.2.1.3
TwinVQ	7	ISO/IEC 14496-3 subpart 4	
CELP	8	ISO/IEC 14496-3 subpart 3	
HVXC	9	ISO/IEC 14496-3 subpart 2	
TTSI	12	ISO/IEC 14496-3 subpart 6	
Main synthetic	13	ISO/IEC 14496-3 subpart 5	
Wavetable synthesis	14	ISO/IEC 14496-3 subpart 5	
General MIDI	15	ISO/IEC 14496-3 subpart 5	
Algorithmic Synthesis and Audio FX	16	ISO/IEC 14496-3 subpart 5	
ER AAC LC	17	ISO/IEC 14496-3 subpart 4	see subclause 1.6.2.2.2.1.4
ER AAC LTP	19	ISO/IEC 14496-3 subpart 4	see subclause 1.6.2.2.2.1.4
ER AAC scalable	20	ISO/IEC 14496-3 subpart 4	see subclause 1.6.2.2.2.1.4
ER Twin VQ	21	ISO/IEC 14496-3 subpart 4	
ER BSAC	22	ISO/IEC 14496-3 subpart 4	

ER AAC LD	23	ISO/IEC 14496-3 subpart 4	see subclause 1.6.2.2.2.1.4
ER CELP	24	ISO/IEC 14496-3 subpart 3	
ER HVXC	25	ISO/IEC 14496-3 subpart 2	
ER HILN	26	ISO/IEC 14496-3 subpart 7	
ER Parametric	27	ISO/IEC 14496-3 subpart 2 and 7	
SSC	28	ISO/IEC 14496-3 subpart 8	
PS	29	ISO/IEC 14496-3 subpart 8	
(reserved)	30		
(escape)	31		
Layer-1	32	ISO/IEC 14496-3 subpart 9	
Layer-2	33	ISO/IEC 14496-3 subpart 9	
Layer-3	34	ISO/IEC 14496-3 subpart 9	
DST	35	ISO/IEC 14496-3 subpart 10	
ALS	36	ISO/IEC 14496-3 subpart 11	

In Part 3: Audio, Subpart 1, under 1.6.3 Semantics, after 1.6.3.13 extensionAudioObjectType add:

1.6.3.14 psPresentFlag

A one bit field indicating the presence or absence of Parametric Stereo data. The value –1 indicates that the psPresentFlag was not conveyed in the AudioSpecificConfig(). In this case, a High Efficiency AAC v2 Profile decoder shall support implicit signaling (see subclause 1.6.6).

In Part 3: Audio, Subpart 1, after 1.6.5 Signaling of SBR, add the following subclause:

1.6.6 Signaling of Parametric Stereo (PS)

1.6.6.1 Generating and Signaling HE AAC + PS Content

The PS tool in combination with the HE AAC coder enables good stereo quality at very low bitrates. At the same time it allows for compatibility with existing HE AAC-only decoders. However, the output from a HE AAC decoder will only be mono for a HE AAC v2 stream carrying PS data.

Therefore, depending on the application, a content provider or content creator may want to choose between the two alternatives given below. In general, the PS data is always embedded in the HE AAC stream in a HE AAC compatible way (in the sbr_extension element), and PS is a pure post processing step in the decoder. Therefore, compatibility can be achieved. However, by means of different signaling the content creator can select between the full-quality mode and the backward compatibility mode as outlined in 1.6.6.1.1 and 1.6.6.1.2.

For the hierarchical profiles, a profile higher in the profile hierarchy is of course able to decode the content of a profile lower in the profile hierarchy. In Figure 1.0A the hierarchical structure of the AAC, HE AAC and HE AAC v2 Profile is displayed. The figure shows that a HE AAC Profile decoder is fully capable of decoding any AAC-Profile stream, given that the HE AAC Profile decoder is of the same or a higher level as indicated in the AAC Profile stream. Similarly the HE AAC v2 decoder can handle all HE AAC Profile streams as well as all AAC Profile streams.

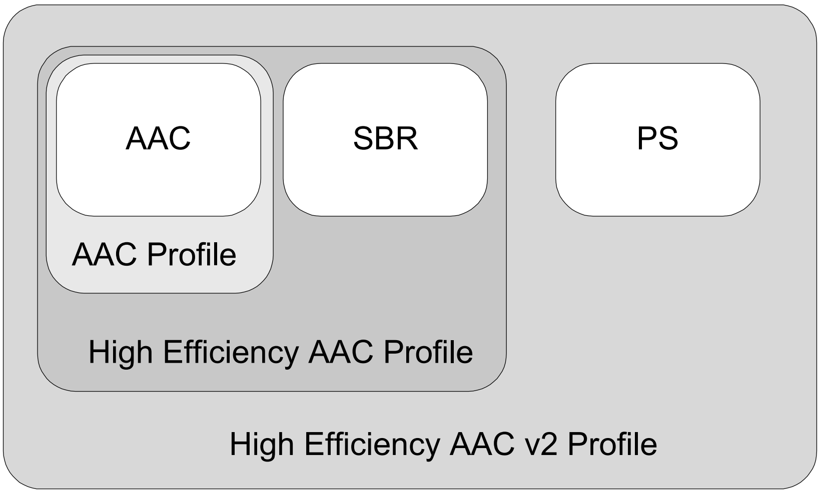

Figure 1.0A – Hierarchical structure of AAC, HE AAC and HE AAC v2 Profile, and compatibility between them.

1.6.6.1.1 Ensuring Full Audio Quality of AAC+SBR+PS for the Listener

To ensure that listeners get the full audio quality of AAC+SBR+PS, the stream should indicate the HE AAC v2 Profile and use the explicit, hierarchical signaling (signaling 2.A. as described below), so that it is played by HE AAC v2 Profile decoders, i.e., PS capable decoders. With regard to HE AAC-only streams or AAC-only streams, an HE AAC v2 Profile decoder will decode all HE AAC Profile streams and AAC Profile streams of the appropriate level, as the HE AAC v2 Profile is a superset of the HE AAC Profile and the AAC Profile.

1.6.6.1.2 Achieving Backward Compatibility with Existing HE AAC and AAC Decoders

The aim of this mode is to get all AAC-based and HE AAC-based decoders to play the stream, even if they do not support the PS tool. Compatible streams can be created using the following two signaling methods:

a) indicate a profile containing SBR (e.g. the HE AAC Profile), but not the HE AAC v2 Profile, and use the explicit backward compatible signalling (2.B. as described below). This method is recommended for all MPEG-4 based systems in which the length of the AudioSpecificConfig() is known in the decoder. As this is not the case for LATM with audioMuxVersion==0 (see clause 1.7), this method cannot be used for LATM with audioMuxVersion==0. In explicit backward compatible signaling, PS-specific configuration data is added at the end of the AudioSpecificConfig(). Decoders that do not know about PS will ignore these parts, while HE AAC v2 Profile decoders will detect its presence and configure the decoder accordingly.

b) indicate a profile containing SBR (e.g. the HE AAC Profile), but not the HE AAC v2 Profile, and use implicit signalling. In this mode, there is no explicit indication of the presence of PS data. Instead, HE AAC v2 Profile decoders shall open two output channels for a stream containing SBR data with channelConfiguration==1, i.e., a mono stream using a single channel element, and check the presence of PS data while decoding the stream and use the PS tool if PS data is found. This is possible because PS can be decoded without PS-specific configuration data if a certain way of handling decoder number of output channels is obeyed, as described below for HE AAC v2 Profile decoders.

Both methods lead to the result that, provided that the profile indication indicates a profile supported by the decoder, the AAC+SBR part of an AAC+SBR+PS streams will be decoded by HE AAC-only decoders, and the AAC part of an AAC+SBR+PS stream will be decoded by AAC-only decoders. HE AAC v2 decoders will detect the presence of PS and decode the full quality AAC+SBR+PS stream.

1.6.6.2 Implicit and Explicit Signaling of Parametric Stereo

This subclause outlines the different signaling methods of PS, and the decoder behavior for different types of signaling.

There are several ways to signal the presence of PS data:

1. **implicit signaling:** If bs_extension_id equals EXTENSION_ID_PS, PS data is present in the sbr_extension element, and this implicitly signals the presence of PS data. The ability to detect and decode implicitly signaled PS is mandatory for all High Efficiency AAC v2 Profile (HE AAC v2 Profile) decoders.

2. **explicit signaling:** The presence of PS data is signaled explicitly by means of the PS Audio Object Type and the psPresentFlag in the AudioSpecificConfig(). When explicit signaling of PS is used, implicit signaling of PS shall not occur. Two different types of explicit signaling are available:

   2.A. **hierarchical signaling:** If the first audioObjectType (AOT) signaled is the PS AOT, the extensionAudioObjectType is set to SBR, and a second audio object type is signaled which indicates the underlying audio object type. This signaling method is not backward compatible. This method may be needed in systems that do not convey the length of the AudioSpecificConfig(), such as LATM with audioMuxVersion==0, and content authors are encouraged to use it only when thus needed.

   2.B. **backward compatible signaling:** If the extensionAudioObjectType SBR is signaled at the end of the AudioSpecificConfig(), a psPresentFlag is transmitted at the end of the backward compatible explicit SBR signaling, indicating the presence or absence of PS data. This method shall only be used in systems that convey the length of the AudioSpecificConfig(). Hence, it shall not be used for LATM with audioMuxVersion==0.

For all types of parametric stereo signaling, the channelConfiguration in the audioSpecifcConfig indicates the number of channels of the underlying AAC coded stream. Hence, if parametric stereo data is available, the channelConfiguration will be one, indicating a single channel element, while the parametric stereo tool will produce two output channels based on the single channel element and the parametric stereo data.

Table 1.22A shows the decoder behavior depending on profile and audio object type indication when implicit or explicit signaling is used.

Table 1.22A – PS Signaling and Corresponding Decoder Behavior

Bitstream characteristics				Decoder behavior	
Profile indication	**PS signaling**	**psPresent Flag**	**raw_data_block**	**HE AAC Profile Decoders**	**HE AAC v2 Profile Decoders**
High Efficiency AAC Profile	signaling 1, implicit signaling (first AOT != PS)	-1	AAC+SBR	Play AAC+SBR	Play AAC+SBR (Note 1)
			AAC+SBR+PS	Play AAC+SBR	Play at least AAC+SBR, should play AAC+SBR+PS (Note 1)
	signaling 2.B, backwards compatible explicit signaling (second AOT == SBR)	0	AAC+SBR	Play AAC+SBR	Play AAC+SBR (Note 2)
		1	AAC+SBR+PS	Play AAC+SBR	Play at least AAC+SBR, should play AAC+SBR+PS (Note 3)
High Efficiency AAC v2 Profile	signaling 2.A, non-backwards compatible signaling (first AOT == PS)	1	AAC+SBR+PS	Undefined	Play AAC+SBR+PS (Note 3)
	signaling 2.B, backwards compatible signaling (second AOT == SBR)	1	AAC+SBR+PS	Undefined	Play AAC+SBR+PS (Note 3)
Note 1: Implicit signaling, assume the presence of PS data in the payload, giving two output channels for a single channel element. Note 2: Explicitly signals that there is no PS data, hence no implicit signaling is present. Note 3: Number of output channels is two for a single channel element containing AAC+SBR+PS data.					

The upper part of Table 1.22A displays bitstream characteristics and decoder behavior if the profile indication is the High Efficiency AAC Profile. The lower part displays bitstream characteristics and decoder behavior if the profile indication is the High Efficiency AAC v2 Profile.

1.6.6.3 HE AAC v2 Profile Decoder Behavior in Case of Implicit Signaling

If the presence of PS data is backward compatible implicitly signaled (signaling 1, in the list above) the first AudioObjectType signaled is not the PS AOT, and the psPresentFlag is not read from the AudioSpecificConfig(). Hence, the psPresentFlag is set to –1, indicating that implicit signaling of parametric stereo may occur.

Since a received mono stream will result in a stereo output if Parametric Stereo data is present in the stream, the HE AAC v2 Profile decoder shall assume that PS data is available and decide the number of output channels to be two for a single channel element containing SBR data, and thus also possibly PS data. If no PS data is found the mono output shall be mapped to the two opened channels for every single channel element.