INTERNATIONAL STANDARD

ISO/IEC 14496-3

Third edition
2005-12-01

AMENDMENT 5
2007-07-15

Information technology — Coding of audio-visual objects —

Part 3: **Audio**

AMENDMENT 5: BSAC extensions and transport of MPEG Surround

Technologies de l'information — Codage des objets audiovisuels —

Partie 3: Codage audio

AMENDEMENT 5: Extensions BSAC et transport d'ambiance MPEG

Reference number
ISO/IEC 14496-3:2005/Amd.5:2007(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 5 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 normative syntax of the integration of ER BSAC and the SBR tool and the corresponding decoding process. An informative encoder description is given in the Annex. Furthermore, this Amendment specifies the transport and embedding of MPEG Surround (ISO/IEC 23003-1) side information in MPEG-4 AAC bitstreams and MPEG-4 Audio environments.

Information technology — Coding of audio-visual objects —

Part 3:
Audio

AMENDMENT 5: BSAC extensions and transport of MPEG Surround

In the following, changes to existing text and Tables are highlighted by gray background.

In the fourth paragraph of subclause 0.3.2.2 General audio coding tools, replace:

It operates at up to 48 kHz sampling rate and uses a frame length of 512 or 480 samples, compared to the 1024 or 960 samples used in standard MPEG-2/4 AAC to enable coding of general audio signals with an algorithmic delay not exceeding 20 ms.

with:

To enable coding of general audio signals with an algorithmic delay not exceeding 20 ms at 48 kHz, it uses a frame length of 512 or 480 samples (compared to the 1024 or 960 samples used in standard MPEG-2/4 AAC).

In Part 3: Audio, Subpart 1, in subclause 1.2 Normative references, add:

ISO/IEC 14496-23, *Information technology — Coding of audio-visual objects — Part 23: Symbolic Music Representation*

ISO/IEC 23003-1*, Information technology — MPEG audio technologies — Part 1: MPEG Surround*

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

PS: Parametric Stereo

SMR: Symbolic Music Representation

and increase the index-number of subsequent entries.

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

Tools/ Modules Audio Object Type	Remark	Object Type ID
…		
MPEG Surround		30
…		
(escape)	X	31
…		
(reserved)		39
SMR Simple		40
SMR Main		41
...		

In Part 3: Audio, Subpart 1, after subclause 1.5.1.2.32 add the following subclauses:

1.5.1.2.33 PS object type

The PS object type contains the PS tool and can be combined with the SBR tool.

1.5.1.2.34 MPEG Surround object type

The MPEG Surround object type conveys MPEG Surround side information (see ISO/IEC 23003-1) in the MPEG-4 Audio framework.

1.5.1.2.35 SMR Simple object type

The SMR simple object type it is used to transport musical scores to be rendered both audibly and visually. The encoded data contains information on the main score, the parts (i.e. instrument specific scores), the possible multilingual lyrics associated with the parts, the visual formatting rules to be used for visual rendering, fonts to be used for custom musical symbols and synchronization information. The fonts and the synchronization information are represented as binary data, all the other as XML data. The SMR simple object type can convey the XML data as plain XML text or as gzip-ped XML (see ISO/IEC 14496-23).

1.5.1.2.36 SMR Main object type

The SMR main object type can transport the musical scores exactly in the same way as the SMR Simple object type but in this case the XML data can be encoded also using the MPEG-B tools defined in ISO/IEC 23001-1 (see ISO/IEC 14496-23).

In Part 3: Audio, Subpart 1, in subclause 1.5.1.2.6 SBR object type, replace Table 1.2 with the table below:

Table 1.2 — Audio object types that can be combined with SBR

Audio Object Type	Object Type ID
AAC main	1
AAC LC	2
AAC SSR	3
AAC LTP	4
AAC Scalable	6
ER AAC LC	17
ER AAC LTP	19
ER AAC scalable	20
ER BSAC	22

In subclause 1.5.2.1, replace Table 1.3 Audio Profiles definition with the table below:

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	X*			X*							
15	General MIDI	X*			X*							
16	Algorithmic Synthesis and Audio FX	X*			X*							
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	MPEG Surround											
31	(escape)											
32	Layer-1											
33	Layer-2											
34	Layer-3											
35	DST											

36	ALS											
37	SLS											
38	SLS non-core											
39	(reserved)											
40	SMR Simple											
41	SMR Main											

*) Subset of Main Synthetic

In subclause 1.6.2.1, extend Table 1.13 — AudioSpecificConfig() as follows:

Table 1.13 — Syntax of AudioSpecificConfig()

Syntax	No. of bits	Mnemonic
AudioSpecificConfig ()		
{		
audioObjectType = GetAudioObjectType();		
samplingFrequencyIndex;	**4**	**uimsbf**
if (samplingFrequencyIndex==0xf)		
samplingFrequency;	**24**	**uimsbf**
channelConfiguration;	**4**	**uimsbf**
sbrPresentFlag = -1;		
psPresentFlag = -1;		
if (audioObjectType == 5 ||		
audioObjectType == 29) {		
extensionAudioObjectType = audioObjectType;		
sbrPresentFlag = 1;		
if (audioObjectType == 29) {		
psPresentFlag = 1;		
}		
extensionSamplingFrequencyIndex;	**4**	**uimsbf**
if (extensionSamplingFrequencyIndex==0xf)		
extensionSamplingFrequency;	**24**	**uimsbf**
audioObjectType = GetAudioObjectType();		
if (audioObjectType == 22)		
extensionChannelConfiguration;	**4**	**uimsbf**
}		
else {		
extensionAudioObjectType = 0;		
}		
switch (audioObjectType) {		
...		
case 28:		
SSCSpecificConfig();		
break;		
case 30:		
sacPayloadEmbedding;	**1**	**uimsbf**
SpatialSpecificConfig();		
break;		
case 32:		
case 33:		
case 34:		
...		

```
    case 40:
    case 41:
        SymbolicMusicSpecificConfig()
        break;
    default:
        /* reserved */
    }

    ...

    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
                        }
                    }
                }
            }
            if ( extensionAudioObjectType == 22 ) {
                sbrPresentFlag;                             1       uimsbf
                if (sbrPresentFlag == 1) {
                    extensionSamplingFrequencyIndex;        4       uimsbf
                    if ( extensionSamplingFrequencyIndex == 0xf )
                        extensionSamplingFrequency;         24      uimsbf
                }
                extensionChannelConfiguration;              4       uimsbf
            }
        }
    }
}
```


After subclause 1.6.2.1.13, add the following new subclauses:

1.6.2.1.14 SpatialSpecificConfig

Defined in ISO/IEC 23003-1:2007, 5.1.

1.6.2.1.15 SymbolicMusicSpecificConfig

Defined in ISO/IEC 14496-23:—[1)], 7.2.1.

1) To be published.

In subclause 1.6.2.2.1, extend Table 1.15 — Audio Object Types as follows:

Table 1.15 — 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		
…			
29	PS	ISO/IEC 14496-3 subpart 8	
30	MPEG Surround	ISO/IEC 23003-1	
31	(escape)		
…			
39	(reserved)		
40	SMR Simple	ISO/IEC 14496-23	
41	SMR Main	ISO/IEC 14496-23	

After subclause 1.6.3.15 fillBits, add the following new subclauses:

1.6.3.16 extensionChannelConfiguration

A four bit field indicating the channel configuration of the BSAC extensions. This configuration data is available in case of explicit signalling for the BSAC extensions. The number of audio output channels is determined according to Table 1.17 — Channel Configuration.

1.6.3.17 sacPayloadEmbedding

The audio Object Type ID 30 MPEG Surround is used to convey spatial audio coding side information for MPEG Surround decoding as defined in ISO/IEC 23003-1. Depending on this flag, the MPEG surround data payload, i.e. SpatialFrame(), is available by different means:

Table 1.19A — sacPayloadEmbedding

sacPayloadEmbedding	Meaning
0	One SpatialFrame() 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.

In Part 3: Audio, Subpart 1, after 1.6.6 Signaling of Parametric Stereo (PS), add the following new sublcause:

1.6.7 Signalling of BSAC Extension payloads

The implicit signalling method for BSAC extension payloads is similar to that of SBR tool. The BSAC decoder that can decode the BSAC extension payloads checks if there is an extension type for the SBR tool such as 'EXT_BSAC_SBR_DATA' in the bsac_raw_data_block(). The sampling frequency should be updated, when the extension type is detected and the SBR tool is operated in dual-rate mode.

The BSAC extension decoder checks if there is the extension type for the BSAC channel extension such as 'EXT_BSAC_CHANNEL' in the bsac_raw_data_block(). In case where the extension type for multichannel, the number of channel from the AudioSpecificConfig() for BSAC Audio Object Type is updated depending on the 'channel_configu-ration_index' of each extended_bsac_base_element().

When explicit signalling is used, implicit signalling shall not occur. Two different types of explicit signalling are available:

1. Explicit Signalling Method 1: hierarchical signalling
   If the first audioObjectType (AOT) signalled is the SBR AOT, a second audio object type is signalled which indicates the ER BSAC AOT. The extensionChannelConfiguration indicates the total number of channels in the bsac_raw_data_block().

2. Explicit Signalling Method 2 : backward compatible signalling
   The extensionAudioObjectType is signalled at the end of the AudioSpecificConfig(). If the extensionAudioObjectType is the ER BSAC AOT, the extensionChannelConfiguration indicates the total number of channels in the bsac_raw_data_block(). 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.

Table 1.22B explains the decoder behaviour with SBR and BSAC channel extension signalling.