



International
Standard

ISO/IEC 14496-26

Information technology — Coding of audio-visual objects —

Part 26: Audio conformance

*Technologies de l'information — Codage des objets
audiovisuels —*

Partie 26: Conformité audio

Second edition
2024-11

Reference number
ISO/IEC 14496-26:2024(en)

Full document details: ISO IEC 14496-26-2024

 **Documents**
© ISO/IEC 2024 Document Sharing Hub



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2024

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword.....	v
Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms and definitions	2
3.1 Conformance Data.....	2
3.2 Conformance Tools.....	2
3.3 Conformance Test Sequences.....	2
3.4 Compressed Data.....	2
3.5 Reference Waveforms.....	2
4 Conformance Points	2
5 Profiles.....	4
6 Conformance data	4
6.1 File name conventions.....	4
6.2 Content	6
7 Audio Object Types	7
7.1 General	7
7.2 Null.....	13
7.3 AAC-based scalable configurations	14
7.4 AAC (main, LC, ER LC, SSR, LTP, ER LTP, ER LD, scalable, ER scalable)	15
7.5 TwinVQ and ER_TwinVQ	40
7.6 ER BSAC	43
7.7 CELP	51
7.8 ER CELP	55
7.9 HVXC.....	60
7.10 ER HVXC.....	71
7.11 ER HILN and ER Parametric	74
7.12 TTSI.....	89
7.13 General MIDI.....	91
7.14 Wavetable Synthesis	92
7.15 Algorithmic Synthesis and AudioFX.....	93
7.16 Main Synthetic.....	100
7.17 SBR.....	101
7.18 PS (Parametric Stereo).....	113
7.19 SSC (Sinusoidal Coding)	115
7.20 DST (Lossless coding of oversampled audio).....	120
7.21 Layer-3.....	123
7.22 ALS (Audio lossless coding)	125
7.23 SLS (Scalable Lossless Coding).....	127
7.24 Layer-1 and Layer 2	130
7.25 Low Delay SBR.....	132
8 Audio EP tool	143
8.1 Compressed data	143
8.2 Decoders	145
9 Audio Composition	151
9.1 AudioBIFS v1.....	151
9.2 Advanced Audio BIFS nodes	162
9.3 AudioBIFS v3 Nodes.....	188

10	MPEG-4 audio transport stream	205
10.1	General	205
10.2	Compressed Data	205
10.3	Decoders	206
11	Upstream	207
11.1	Compressed data	207
11.2	Decoders	207
12	Conformance test sequence assignment to profiles and levels	207
12.1	Overview	207
12.2	Audio	208
12.3	Systems	224
Annex A (informative)	Complexity measurement criteria and tool for level definitions of algorithmic synthesis and AudioFX Object Type	229
Annex B (informative)	Test bitstreams for the CELP object type	250

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and <https://patents.iec.ch>. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This second edition cancels and replaces the first edition (ISO/IEC 14496-26:2010), which has been technically revised. It also incorporates the Amendments ISO/IEC 14496-26:2010/Amd 2:2010, ISO/IEC 14496-26:2010/Amd 3:2014, ISO/IEC 14496-26:2010/Amd 4:2016, ISO/IEC 14496-26:2010/Amd 5:2018 and the Technical Corrigenda ISO/IEC 14496-26:2010/Cor 2:2011, ISO/IEC 14496-26:2010/Cor 3:2011, ISO/IEC 14496-26:2010/Cor 4:2011, ISO/IEC 14496-26:2010/Cor 5:2012, ISO/IEC 14496-26:2010/Cor 6:2013, ISO/IEC 14496-26:2010/Cor 7:2013 and ISO/IEC 14496-26:2010/Cor 8:2015.

The main changes are as follows:

- Introduced additional BSAC conformance bitstreams to assist in implementing Terrestrial-DMB products
- Correction according to bitstreams and wave files for BSAC
- Correction according to `channel_pair_element()`
- Correction for MPEG-4 ALS floating point bitstreams
- Correction for MPEG-4 SLS test sequences

- Correction for BSAC and HE-AAC V2 profile
- Correction for ER AAC test sequences
- New conformance bitstreams for Low Delay AAC V2 profile
- Correction for AAC block length parameter and test sequences
- Additional multichannel conformance data for AAC and HE-AAC
- Additional levels of ALS simple profile and SBR enhancements
- Additional levels of MPEG-4 ALS simple profile supporting high-resolution audio

A list of all parts in the ISO/IEC 14496 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

ISO/IEC 14496-3 specifies coded representations of audio information. ISO/IEC 14496-3 allows for large flexibility, achieving suitability of ISO/IEC 14496 for many different applications. The flexibility is obtained by including parameters in the bitstream that define the characteristics of coded bitstreams. Examples are the audio sampling frequency, bitrate parameters, synchronisation timestamps, the association of bitstreams and synthetic objects within objects.

Characteristics of compressed data and decoders are defined for ISO/IEC 14496-3. The compressed data characteristics define the subset of the standard that is exploited in the compressed data. Examples are the applied values or range of the sampling rate and bitrate parameters. Decoder characteristics define the properties and capabilities of the applied decoding process. An example of a property is the applied arithmetic accuracy. The capabilities of a decoder specify which compressed data the decoder can decode and reconstruct, by defining the subset of the standard that may be exploited in the decodable compressed data. Compressed data can be decoded by a decoder if the characteristics of the compressed data are within the subset of the standard specified by the decoder capabilities.

The tests described in this document can be used for various purposes such as:

- manufacturers of encoders, and their customers, can use the tests to verify whether the encoder produces bitstreams compliant with ISO/IEC 14496-3.
- manufacturers of decoders and their customers can use the tests to verify whether the decoder meets the requirements specified in ISO/IEC 14496-3 for the claimed decoder capabilities.
- manufacturers and customers of terminals supporting interactive, broadcast and local sessions over a multitude of transport protocols and networks, can use the tests to verify whether the claimed functionalities are compliant with ISO/IEC 14496-6.
- manufacturers of test equipments, and their customers can use the tests to verify compliance with ISO/IEC 14496-3.

Information technology — Coding of audio-visual objects —

Part 26: Audio conformance

1 Scope

This document specifies how tests can be designed to verify whether compressed data and decoders meet requirements specified by ISO/IEC 14496-3. Encoders are not addressed specifically. An ISO/IEC 14496 encoder generates compressed data compliant with the syntactic and semantic bitstream payload requirements specified in ISO/IEC 14496-3.

This document summarises the requirements, cross references them to characteristics, and defines how conformance with them can be tested. Guidelines are given on constructing tests to verify decoder conformance. Some examples of compressed data implemented according to these guidelines are provided as an electronic annex to this document usually together with their uncompressed counterparts (reference waveforms).

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies..

ISO/IEC 11172-3, *Information technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s — Part 3: Audio*

ISO/IEC 11172-4, *Information technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s — Part 4: Compliance testing*

ISO/IEC 13818-3, *Information technology — Generic coding of moving pictures and associated audio information — Part 3: Audio*

ISO/IEC 13818-4, *Information technology — Generic coding of moving pictures and associated audio information — Part 4: Conformance testing*

ISO/IEC 13818-7, *Information technology — Generic coding of moving pictures and associated audio information — Part 7: Advanced Audio Coding (AAC)*

ISO/IEC 14496-1, *Information technology — Coding of audio-visual objects — Part 1: Systems*

ISO/IEC 14496-3, *Information technology — Coding of audio-visual objects — Part 3: Audio*

ISO/IEC 14496-3:2019, *Information technology — Coding of audio-visual objects — Part 3: Audio*

ISO/IEC 14496-11, *Information technology — Coding of audio-visual objects — Part 11: Scene description and application engine*

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

ISO/IEC 23003-2:2018, *Information technology — MPEG audio technologies — Part 2: Spatial Audio Object Coding (SAOC)*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 14496-1, ISO/IEC 14496-3 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at <https://www.iso.org/obp>

— IEC Electropedia: available at <https://www.electropedia.org/>

3.1 conformance data

conformance test sequences and conformance tools

3.2 conformance tools

tools which are provided within an electronic annex of this document to check certain conformance criteria

3.3 conformance test sequences

set of compressed data and its reference waveforms

Note 1 to entry: The conformance test sequences are provided as examples at: <https://standards.iso.org/iso-iec/14496/-26/ed-2/en/>.

3.4 compressed data

encoded data according to ISO/IEC 14496-3

3.5 reference waveforms

decoded counterparts of the compressed data

4 Conformance points

All audio decoders except the LATM-based decoders are part of the MPEG-4 framework. Table 1 gives an overview about the interfaces that have to be provided to test the audio decoders using the MPEG-4 System.

Table 1 — Conformance points

conformance point / interface	data flow direction	description / reference
AudioSpecificConfig	in	audio related decoder specific information, see ISO/IEC 14496-3:2019, (1.6.2.1 AudioSpecificConfig)
audio access units	in	audio related bitstream payload, see ISO/IEC 14496-1:2004 (7.1.2.3 Access Units (AU))
BIFS/AudioSource node	in	see ISO/IEC 14496-11: 2005 (7.2.2.15 Audio Source)
private test info	in	to control some elements which are usually generated by random number generators
audio composition units	out	see ISO/IEC 14496-1: 2004 (7.2.8 Composition Units (CU))

Figure 1 gives an overview about the test bench (MPEG-4 System), the system under test (Audio decoder), and the interfaces between them. Figure 2 gives a more detailed view on the audio decoder, consisting of error protection (EP) decoder and audio core decoder.

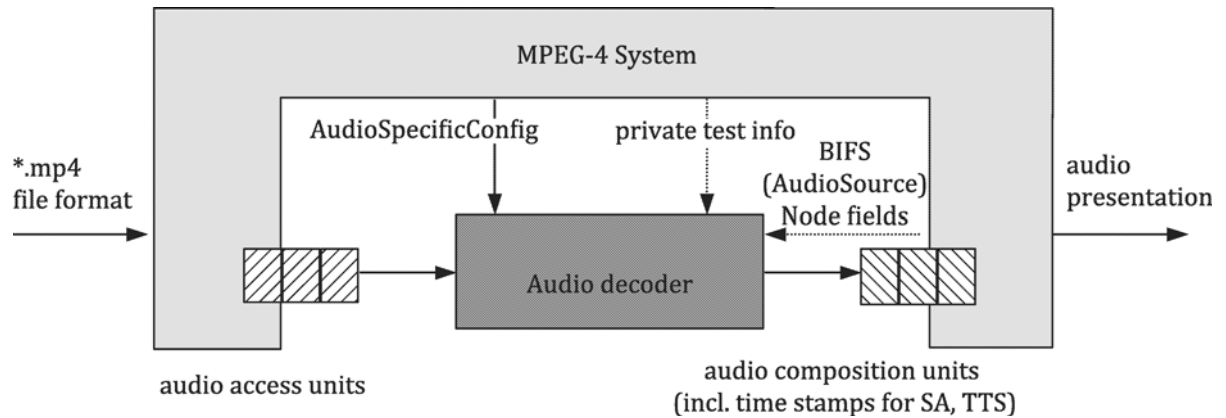


Figure 1 — Audio conformance points

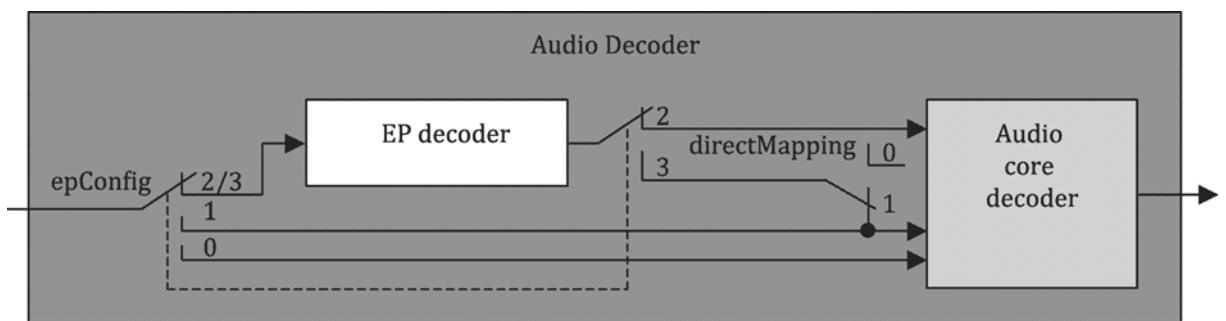


Figure 2 — Audio decoder structure

Clause 7 describes:

The conformance criteria of the audio core decoder.

The conformance criteria of the compressed data not requiring the EP decoder ($\text{epConfig} == 0 \parallel \text{epConfig} == 1$).

The properties of the examples of compressed data with ($\text{epConfig} == 0 \parallel \text{epConfig} == 1$).

Clause 8 describes:

The conformance criteria of the EP decoder

The conformance criteria of the compressed data requiring the EP decoder ($\text{epConfig} == 2 \parallel \text{epConfig} == 3$).

The properties of the examples of compressed data with ($\text{epConfig} == 2 \parallel \text{epConfig} == 3$).

Compressed data with different epConfig settings are available referring to the same reference waveforms. Here, the output of a conforming decoder shall be equal, independently of the used epConfig setting.

For some of the compressed data containing scalable configurations, conformance points are defined at the PCM output of the decoder for m layers being decoded from an n -layer input, where m is an integer in the range 0 (base layer conformance) to $n-1$. The reference PCM decoder output signals corresponding to these conformance points are listed in the respective conformance tables.

5 Profiles

ISO/IEC 14496-3 defines several profiles and several levels within each profile. Conformance is always tested against a certain level within a certain profile. Audio profiles always comprise a set of audio object types. Nevertheless the conformance criteria as described within this document are based on audio object types. The assignment of object types to profiles as well as the level definitions can be found in ISO/IEC 14496-3. The conformance of a certain level within a certain profile is fulfilled, if the conformance of each object type belonging to this profile is fulfilled. The assignment of the provided test sequences to profiles and levels can be found in Clause 12.

6 Conformance data

6.1 File name conventions

For all conformance test sequences, the file name convention given in Table 2 is used.

Table 2 — File name conventions

object type name/ tool name	File Name (compressed)	File Name (uncompressed)
AdvancedAudioBIFS - perceptual approach	aabper<coreSetup>	-- not applicable --
AdvancedAudioBIFS - physical approach	aabphy<coreSetup>	-- not applicable --
AudioBIFS	ab<coreSetup>_<coder>	ab<coreSetup>_<coder>
AudioBIFS v3	ABv3_<nodeAbbrev><coreSetup>	-- not applicable --
AAC scalable	ac<coreSetup>	ac<coreSetup>[_lay<highestLay>]
AAC LC	al<coreSetup>_<fs>	al<coreSetup>_<fs>[_cut<fac>_boost<facr>] [_level<lvl>][_<chan>]
AAC main	am<coreSetup>_<fs>	am<coreSetup>_<fs>[_cut<fac>_boost<facr>] [_level<lvl>][_<chan>]
AAC LTP	ap<coreSetup>_<fs>	ap<coreSetup>_<fs>
AAC SSR	as<coreSetup>_<fs>	as<coreSetup>_<fs>[_<chan>]
CELP	ce<coreSetup>	ce<coreSetup>[_lay<highestLay>]
ER AAC scalable	er_ac<coreSetup>_ep<epConfig> [<epSetup>]	er_ac<coreSetup>[_lay<highestLay>]
ER AAC LD	er_ad<coreSetup>_<fs>_ep<epConfig> [<epSetup>]	er_ad<coreSetup>_<fs>
ER AAC LC	er_al<coreSetup>_<fs>_ep<epConfig> [<epSetup>]	er_al<coreSetup>_<fs>
ER AAC LTP	er_ap<coreSetup>_<fs>_ep<epConfig> [<epSetup>]	er_ap<coreSetup>_<fs>
SBR (+AAC LC)	al_sbr_<tool>_<fs>_<nchan>[_fsaac<fs>][_sig<sig>]	al_sbr_<mode>_<tool>_<fs>_<nchan> [_fsaac<fs>][_sig<sig>][_<chan>]
SBR (+AAC LC with 960 samples per frame)	al960_sbr_<tool>_<fs>_<nchan> [_fsaac<fs>][_sig<sig>]	al960_sbr_<mode>_<tool>_<fs>_<nchan> [_fsaac<fs>][_sig<sig>][_<chan>]
PS (+SBR+AAC LC)	al_sbr_ps_<coreSetup>	al_sbr_ps_<coreSetup>[_<version>]
SSC	ssc_<tool>_<nchan>[_sig<sig>]	ssc_<mode>_<tool>_<nchan>[_sig<sig>][_<chan>]
DST	dst_<tool>_<nchan>[_sig<sig>]	dst_<mode>_<tool>_<nchan>[_sig<sig>][_<chan>]
Layer-3	l3_<coreSetup>	l3_<coreSetup>

object type name/ tool name	File Name (compressed)	File Name (uncompressed)
ER BSAC	er_bs<coreSetup>_<fs>_ep<epConfig>[<epSetup>]	er_bs<coreSetup>_<fs>[_lay<highestLay>]
ER CELP	er_ce<coreSetup>_ep<epConfig>[<epSetup>]	er_ce<coreSetup>[_lay<highestLay>]
ER HILN	er_hi<coreSetup>_ep<epConfig>[<epSetup>]	er_hi<coreSetup>[_lay<highestLay>][_s<speedFac>][_p<pitchFac>]
ER HVXC	er_hv<coreSetup>_ep<epConfig>[<epSetup>]	er_hv<coreSetup>[_lay<highestLay>]_<delay>
ER Parametric	er_pa<coreSetup>_ep<epConfig>[<epSetup>]	er_pa<coreSetup>[_lay<highestLay>]_<delay>
ER Twin VQ	er_tv<coreSetup>_ep<epConfig>[<epSetup>]	er_tv<coreSetup>[_lay<highestLay>]
HVXC	hv<coreSetup>	hv<coreSetup>[_lay<highestLay>]_ref<decCfg>
Algorithmic Synthesis and Audio FX	sy<coreSetup>	sy<coreSetup>
TTSI	tts<coreSetup>	tts<coreSetup>
TwinVQ	tv<coreSetup>	tv<coreSetup>[_lay<highestLay>]
ALS	als_<tool>_<coreSetup>	als_<tool>_<coreSetup>
SLS	sls<coreSetup>_<fs>_<bitres>	sls<coreSetup>_<fs>_<bitres>
Layer-1	l1_<coreSetup>	l1_<coreSetup>
Layer-2	l2_<coreSetup>	l2_<coreSetup>
ER AAC ELD	er_eld<coreSetup>_<fs>_ep<epConfig>[<epSetup>]	er_eld<coreSetup>_<fs>
LD MPEG Surround (+ER AAC LD)	er_ad_ldmps_<coreSetup>	er_ad_ldmps_<coreSetup>
LD MPEG Surround (+ER AAC ELD)	er_eld_ldmps_<coreSetup>	er_eld_ldmps_<coreSetup>

<bitres> can be 16 or 24 and indicates the bit resolution of the coded wavefile

<chan> indicates the channel for multi-channel sequences (f<number> - number of the front channel, b<number>- number of the back channel, s<number> - number of the side channel, l<number> - number of the LSF channel).

<coder> indicates the coder used to encode the content (ce – CELP, sa – Structured Audio, pcm – PCM)

<coreSetup> refers to a certain audio coder setup. It is most likely a number, but may also contain characters.

<delay> refers to the decoder delay, it can become "ld" (low delay) or "nd" (normal delay).

<epConfig> can be 0, 1, 2 or 3, depending on epConfig (defined in AudioSpecificConfig).

<epSetup> is required if (epConfig==2 || epConfig==3). It refers to a certain error protection setup.

<fs> sampling frequency (08, 11, 12, 16, 22, 24, 32, 44, 48, 64, 88 or 96).

_level<lvl> refers to the level with regard to DRC.

_cut<fac>_boost<fac> refers to the cut and boost factors with regard to DRC.

_lay<highestLay> is required for any scalable configuration. It marks the highest layer of the scalable configuration used for decoding (starting with 0 for the core layer).

_p<pitchfac> is a number referring to the decoder configuration with regard to the pitch factor.

_ref<decCfg> is a number referring to the decoder configuration with regard to delay mode, speed and pitch change.

_s<speedfac> is a number referring to the decoder configuration with regard to the speed factor.

<tool> indicates the SBR module mainly targeted by the test sequence. Possible values are "e" for testing the envelope adjuster "s" for testing sine addition, "gh" for testing time-grid transitions in combination with changes of SBR header data, "i" for testing inverse filtering, "qmf" for testing the QMF implementation, "cm" and "gen" for testing various channel modes, "sig" for testing SBR signalling, "twi" for QMF identification, "sr" for testing various combinations of sampling rates, and "esbr" for testing SBR enhancements.

<nodeAbbrev> is the abbreviation of one of the AudioBIFS v3 node names.

<nchan> corresponds to the number of channels present in the conformance test sequence. It is either a single integer, in which case it refers to the number of main audio channels, or two integers separated by a '.', in which case the first integer equals the number of main audio channels, while the second number equals the number of low frequency enhancement channels.

fsaac<fs> corresponds to the sampling rate of the underlying AAC-LC data. If it is omitted, it is half the sampling rate given as output sampling rate.

<sig> is an integer describing the kind of signalling used according to Table 3. If this value is omitted, backwards compatible explicit signalling of SBR is used.

Table 3 – File name conventions for SBR signaling

sig	Signalling method used
0	Implicit signalling of SBR
1	Hierarchical explicit signalling of SBR
2	Backwards compatible explicit signalling of SBR

<mode> is either "hq" or "lp" for the high quality or the low power version of the SBR decoding algorithm respectively.

<version> is either "bl" or "ur" for the baseline or the unrestricted version of the parametric stereo decoding algorithm respectively.

With respect to file extensions, the following rules are applied as shown in Table 4

Table 4 – File extensions

Compressed	MPEG-4 file format	.mp4
Compressed	native MPEG-1/2 Audio storage format	.mpg
Compressed	Audio data interchange format	.adif
Compressed	Audio data transport stream	.adts
Compressed	AudioSyncStream	.ass
Compressed	EPASyncStream	.ess
Compressed	AudioPointerStream	.aps
Uncompressed	HILN Conformance Test Parameters	.ctp
Uncompressed	WAVE format (uncompressed PCM format)	.wav
Uncompressed	TTSI decoded text and control digits	.txt

6.2 Content

The test set includes a set of sine sweeps, a set of musical/speech test sequences and a set of noise-like test sequences. The supplied sine sweeps with an amplitude of -20dB relative to full scale have an absolute amplitude of +/- 0.1.

7 Audio object types

7.1 General

This clause lists all audio object types. It starts with a general description, which may be related to more than one object type.

This clause contains general descriptions for conformance testing on compressed data and decoders. Unless explicitly restricted, these descriptions are related to all object types.

7.1.1 Compressed data

7.1.1.1 Characteristics

Characteristics of compressed data specify the constraints that are applied by the encoder in generating the compressed data. These syntactic and semantic constraints may, for example, restrict the range or the values of parameters that are encoded directly or indirectly in the compressed data. The constraints applied to a given compressed data may or may not be known a priori.

Decoder relevant compressed data may consist of the following parts:

decoder specific information (AudioSpecificConfig)

BIFS/AudioSource node (field information)

audio access units (establishing the bitstream payload)

7.1.1.1.1 ESC instance configuration

In case of `epConfig=1`, each instance of each sensitivity category belonging to one frame is stored separately within a single access unit, i.e. there exist as many elementary streams as instances defined within a frame as shown in Table 6.

Note: In case of `epConfig=3`, the mapping between EP classes and ESC instances is signaled by the data element `directMapping`. In case of `directMapping=1`, the restrictions regarding the ESC instance configuration apply accordingly to the EP class configuration.

Table 5 gives an overview about the valid configurations:

Table 5 — Number of ESC instances that build a frame in case of `epConfig==1`

Audio object type	number of ESC instances to build a frame
ER AAC	see Table 6
ER Twin VQ	non-scalable or base layer: 2 any enhancement layer: 2
ER BSAC	base layer: 2 any large-step enhancement layer: 1
ER CELP	base layer: 5 any enhancement layer: 1
ER HVXC	2 kbit/s, non-scalable or base layer: 4 4 kbit/s, non-scalable: 5 any enhancement layer: 3
ER HILN	base layer: 5 any enhancement/extension layer: 1
ER Parametric	PARAMode==0,1 base layer: 5 PARAMode==2,3 base layer: 15 any enhancement/extension layer: 1

Table 6 — Number of ESC instances that build elements/layers of an ER AAC frame in the case of epConfig==1

	aacScalefactorDataResilienceFlag	
	0	1
single channel element (SCE) / mono layer	3	4
channel pair element (CPE) / stereo layer	7	9
extension payload (EPL)	2	

Depending on the value of the data element channelConfiguration, an AAC frame may cover several instances of SCE, CPE or EPL. This leads to the following valid configurations presented in Table 7.

Table 7 — Number of ESC instances that build an ER AAC frame/layer in the case of epConfig==1

AOT				channelConfiguration	aacScalefactorDataResilienceFlag		N extension payloads
					0	1	
17	19	20	23		main payload		
x	x	x	x	1	3	4	+2*N
x	x	x	x	2	7	9	
x	x		x	3	3+7	4+9	
x	x		x	4	3+7+3	4+9+4	
x	x		x	5	3+7+7	4+9+9	
x	x		x	6	3+7+7+3	4+9+9+4	
x	x		x	7	3+7+7+7+3	4+9+9+9+4	

7.1.1.2 Test procedure

Each compressed data shall meet the syntactic and semantic requirements specified in ISO/IEC 14496-3. For each audio object type a set of semantic tests to be performed on the compressed data is described. To verify whether the syntax is correct is straightforward and therefore not defined herein after. In the description of the semantic tests it is assumed that the tested compressed data contains no errors due to transmission or other causes. For each test the condition or conditions that shall be satisfied are given, as well as the prerequisites or conditions in which the test can be applied.

7.1.2 Decoders

7.1.2.1 Characteristics

The decoder characteristics are defined by the profiles and levels being tested.

7.1.2.2 Test procedure

To test audio decoders, ISO/IEC JTC 1/SC 29/WG 11 supplies a number of test sequences. Supplied sequences cover all profile decoders. For a supplied test sequence, testing can be done by comparing the output of a decoder under test with a reference output also supplied by ISO/IEC JTC 1/SC 29/WG 11. In cases where the decoder under test is followed by additional operations (e.g. quantizing a signal to a 16 bit output signal) the conformance point is prior to such additional operations, i.e. it is permitted to use the actual decoder output (e.g. with more than 16 bit) for conformance testing.

Measurements are carried out relative to full scale where the output signals of the decoders are normalized to be in the range between -1.0 and +1.0.

The following subclauses define a set of test methods. A particular test method for a certain test sequence is specified in the object type specific subclauses.

For elements producing output that cannot be tested with the methods described below, specific conformance testing procedures are described in the object type specific subclauses.

7.1.2.2.1 RMS/LSB measurement

To fulfill the “RMS/LSB Measurement” test at an accuracy level of “K bit”, an ISO/IEC 14496-3 decoder shall provide an output waveform such that the RMS level of the difference signal between the output of the decoder under test and the supplied reference output is less than $2^{-(K-1)}/\sqrt{12}$. In addition, the difference signal shall have a maximum absolute value of at most $2^{-(K-2)}$ relative to full-scale. The “RMS/LSB Measurement” test shall be carried out for an accuracy level of K=16 bit unless a different accuracy level is explicitly stated.

7.1.2.2.1.1 Calculation of RMS

For the calculation of the RMS level, all measurements are carried out relative to full scale where the output signals of the decoder and supplied test sequences are normalized to be in the range between -1.0 and +1.0.

The supplied reference waveforms have a precision (P) of 24 bits, where the most significant bit (MSB) will be labeled bit 0 and the least-significant bit (LSB) will be labeled bit 23. The most significant bit (bit 0) represents the value of -1, the second most significant bit (bit 1) represents the value of +1/2, etc.

$$\begin{aligned}
 \text{value of bit 0 (MSB)} &= -\frac{1}{2^0} = -1 \\
 \text{value of bit 1} &= \frac{1}{2^1} = \frac{1}{2} \\
 \text{value of bit 2} &= \frac{1}{2^2} = \frac{1}{4} \\
 &\vdots \\
 \text{value of bit 23 (LSB)} &= \frac{1}{2^{23}} = \frac{1}{8,388,608}
 \end{aligned}$$

The output waveform of the decoder under test is required to be in the same format. In the case that the output of the decoder has a precision of P' bits and if P' is smaller than 24, then the output is extended to 24 bits by setting bit P' through bit 23 to zero. In the next step, the difference (*diff*) of the samples of these signals has to be calculated. Every channel of a multichannel waveform shall be tested. The total number of samples for each channel is N.

$$diff(n) = \text{'output signal of decoder under test (n)'} - \text{'supplied test sequence (n)'}, \text{ for } n = 1 \text{ to } N$$

The values of all difference samples shall be squared, summed, divided by N and then the square-root shall be calculated. This calculation finally gives the RMS level.

$$rms = \sqrt{\frac{1}{N} \sum_{n=1}^N diff(n)^2}$$

This test only verifies the computational accuracy of an implementation.

Software is provided for performing this verification procedure.

7.1.2.2.2 Segmental SNR

This criterion is designed to test decoders decoding the object types CELP, ER CELP, HVXC, ER HVXC, TwinVQ, ER TwinVQ and ER HILN.

Definition:

$x_a(i)$: i^{th} sample of reference output signal (normalized in a range between -1.0 and 1.0).

$x_b(i)$: i^{th} sample of output signal of a decoder under test normalized in a range between -1.0 and 1.0.

L : the length of segment

N : the total number of segments

$SS(k)$: SNR of k^{th} segment

$SSNR$: segmental SNR

$$SS(k) = \log_{10} \left(1 + \frac{\sum_{i=0}^{L-1} x_a(k \times L + i)^2}{10^{-13} L + \sum_{i=0}^{L-1} (x_a(k \times L + i) - x_b(k \times L + i))^2} \right)$$

$$SSNR = 10 \times \log_{10} \left(10^{\sum_{k=0}^{N-1} SS(k) / N} - 1.0 \right)$$

7.1.2.2.3 Frequency domain criterion based on cepstrum analysis

This criterion is designed to test decoders decoding the object types CELP, ER CELP, TwinVQ, ER TwinVQ and ER HILN.

The cepstrum analysis procedure is defined by means of the functions `lpc2cepstrum` and `calculate_lpc` provided in pseudo C code below.

```
#define LPC_ORDER          16 /* LPC order */
#define CEPSTRUM_ORDER     32 /* Cepstrum order */
#define BW                 0.0125F /* Bandwidth scalefactor */

void lpc2cepstrum (float lpc_coef[], /* in: LPC coefficients (a-parameters) */
                  float C[], /* out: LPC cepstrum */
                  {
    float ss;
    int i, m;

    /* it is assumed that lpc_coef[0] is 1 ! */

    C[1] = -lpc_coef[1];

    for (m = 2; m <= LPC_ORDER; m++)
    {
        ss = -lpc_coef[m] * m;
        for (i = 1; i < m; i++)
        {
```

```

        ss -= lpc_coef[i] * C[m-i];
    }
    C[m] = ss;
}

for (m = LPC_ORDER + 1; m <= CEPSTRUM_ORDER; m++)
{
    ss = 0.0F;
    for (i = 1; i <= LPC_ORDER; i++)
    {
        ss -= lpc_coef[i] * C[m-i];
    }
    C[m] = ss;
}

for (m = 2; m <= CEPSTRUM_ORDER; m++)
{
    C[m] /= m;
}
}

void calculate_lpc (float    *in,          /* in:  input PCM audio data          */
                  int      frame_size,    /* in:  analysis frame length in samples */
                  float    *lpc_coef)     /* out: LPC coefficients              */
{
    int      ip;
    float    wvpowfr, cor[LPC_ORDER + 1];
    float    wlag [LPC_ORDER + 1];
    float    *wdw;

    wdw = (float*) malloc (sizeof (float) * frame_size);

    if (wdw == NULL)
    {
        printf ("Memory allocation error in calculate_lpc.\n");
        exit (1);
    }

    hamwdw (wdw, frame_size);

    for (ip = 0; ip < frame_size; ip++)
    {
        in[ip] *= wdw[ip];
    }

    sigcor (in, frame_size, &wvpowfr, cor, LPC_ORDER);

    lagwdw (wlag, LPC_ORDER, BW);

    for (ip = 1; ip <= LPC_ORDER; ip++)
    {
        cor[ip] *= wlag[ip];
    }

    corref (LPC_ORDER, cor, lpc_coef);

    free (wdw);
}

void hamwdw (float    wdw[],
            int      n)
{
    int      i;
    float    d, pi = 3.141592653589793F;

    d = (float) (2.0 * pi/n);

    for (i = 0; i < n; i++)
    {
        wdw[i] = (float) (0.54 - 0.46 * cos (d * i));
    }
}

void lagwdw (float    wdw[],
            int      n,

```

```

        float    h)
{
    int          i;
    float        pi = 3.141592653589793F;
    float        a, b, w;

    a = (float) (log (0.5) * 0.5 / log (cos (0.5 * pi * h)));
    a = (float) ((int) a);
    w = 1.0F;
    b = a;
    wdw[0] = 1.0F;

    for (i = 1; i <= n; i++)
    {
        b += 1.0F;
        w *= a / b;
        wdw[i] = w;
        a -= 1.0F;
    }
}

void sigcor (float    *sig,
            int      n,
            float    *_pow,
            float    cor[],
            int      p)
{
    int      k, ij;
    float    c, dsqsum;
    float    sqsum = 1.0e-35F;

    if (n > 0)
    {
        for (ij = 0; ij < n; ij++)
        {
            sqsum += (sig[ij] * sig[ij]);
        }
        dsqsum = (float) (1.0 / sqsum);

        for (k = 1; k <= p; k++)
        {
            c = 0.0;
            for(ij = k; ij < n; ij++)
            {
                c += (sig[ij - k] * sig[ij]);
            }
            cor[k] = c * dsqsum;
        }
        k = p;
    }
    *_pow = (float) ((sqsum - 1.e-35) / (float)n);

    cor[0] = 1.0F;
}

void corref (int      p,          /* in:  LPC analysis order          */
            float    cor[],      /* in:  correlation coefficients    */
            float    alf[],      /* out: linear predictive coefficients */

            {
    int      i, j, k;
    float    resid, r, a;
    float    ref[LPC_ORDER + 1];

    ref[1] = cor[1];
    alf[1] = -ref[1];
    resid = (float) ((1.0 - ref[1]) * (1.0 + ref[1]));

    for (i = 2; i <= p; i++)
    {
        r = cor[i];

        for (j = 1; j < i; j++)
        {
            r += alf[j] * cor[i-j];

```

```

    }

    alf[i] = -(ref[i] = (r /= resid));
    j = 0;
    k = i;

    while (++j <= --k)
    {
        a = alf[j];
        alf[j] -= r * alf[k];

        if (j < k)
        {
            alf[k] -= r * a;
        }
    }
    resid = (float) (resid * (1.0 - r) * (1.0 + r));
}

```

7.1.2.2.4 PNS conformance criteria

Two tests based on spectral waveform analysis and one test based on temporal waveform analysis shall be applied.

Spectral PNS conformance analysis:

[PNS-1] Both the decoded output and the reference output signal are analyzed by means of an N-point DFT ($N=2*\text{number_of_spectral_lines_per_frame}$, e. g. 2048-point for AAC LC) with a Hann window and 50 % overlap between subsequent windows. For both signals, the DFT lines are grouped corresponding to scalefactor bands and the accumulated squared absolute values are computed for each scalefactor band. As the first test criterion, the ratio between the energies of both signals averaged over time shall be within the interval [-0.4 dB; 0.4 dB] for each scalefactor band. As the second test criterion, the ratio between the standard deviations (over time) of the energies of both signals shall be within the interval [-0.8 dB; 0.8 dB] for each scalefactor band. For this test, sequences are supplied containing a static spectrum generated by a single PNS codebook section covering all scalefactor bands (i.e. each frame carries the same spectral "envelope", long blocks only, no other codebooks).

[PNS-2] The same type of analysis and the same thresholds are used as in test [PNS-1], but with a window size of $N/8$ and grouping corresponding to scalefactor bands for a SHORT_WINDOW. For this test, sequences are supplied containing a periodic repetition of PNS and Null codebook sections within grouped short blocks ({1;1;1;1;2;2} grouping with PNS switched on in subblocks 0,2,4).

Temporal PNS conformance analysis:

[PNS-3] Starting at the first available decoder frame boundary, the sum of the squared output samples is computed for blocks of $\text{number_of_spectral_lines_per_frame}/16$ samples for both decoded signal and reference signal. As the test criterion, the ratio between the energies of both signals shall be within the interval [-5 dB; 5 dB] for 91 % of the blocks and within the interval [-10 dB; 10 dB] for 99 % of the blocks. For this test, the same sequences as provided for [PNS-2] shall be used.

7.2 Null

The NULL object type provides the possibility to feed raw PCM data directly into the audio compositor. No decoding is involved. The sampling rate and the audio channel configuration is specified by the AudioSpecificConfig.

7.3 AAC-based scalable configurations

7.3.1 Compressed data

7.3.1.1 Characteristics

Encoders may apply restrictions to the following parameters of the compressed data.

7.3.1.1.1 Layer configuration

number of non-AAC layers

number of AAC layers

7.3.1.1.2 AudioSpecificConfig

See description for individual object types.

7.3.1.1.3 Bitstream payload

See description for individual object types.

7.3.1.2 Test procedure

Each compressed data shall meet the syntactic and semantic requirements specified in ISO/IEC 14496-3. This subclause describes a set of semantic tests to be performed on decoder relevant data. The procedure to verify whether the syntax is correct is straightforward and therefore not defined in this subclause. In the description of the semantic tests it is assumed that the tested compressed data contains no errors due to transmission or other causes. For each test the condition or conditions that shall be satisfied are given, as well as the prerequisites or conditions in which the test can be applied.

7.3.1.2.1 Layer configuration

The number of AAC layers shall not exceed 8 in any scalable configuration.

The number of CELP layers (object type 8 or 24) shall not exceed 2, if CELP is used as base layer coder within an AAC based scalable configuration.

The number of TwinVQ layers (object type 7 or 21) shall not exceed 1, if TwinVQ is used as base layer coder within an AAC based scalable configuration.

Only those object type combinations shown in Table 8 are valid.

Table 8 — Valid object type combinations within an AAC-based scalable configuration

audio object type for the base layer coder	audio object type for the AAC enhancement layers
6 (AAC scalable)	6 (AAC scalable)
8 (CELP)	6 (AAC scalable)
7 (TwinVQ)	6 (AAC scalable)
20 (ER AAC scalable)	20 (ER AAC scalable)
24 (ER CELP)	20 (ER AAC scalable)
21 (ER TwinVQ)	20 (ER AAC scalable)

If CELP is used as base layer coder within an AAC based scalable configuration, its `samplingFrequencyIndex` shall be 0xc (7350 Hz) or 0xb (8000 Hz).

7.3.1.2.2 AudioSpecificConfig

7.3.1.2.2.1 AudioSpecificConfig()

channelConfiguration: Shall be 1 in case of `audioObjectType` 7 (TwinVQ) or 21 (ER TwinVQ).

7.3.1.2.3 Bitstream payload

7.3.1.2.3.1 tvq_scalable_main_header()

tns_data_present: Shall be 0.

7.3.1.2.3.2 aac_scalable_main_header()

max_sfb: Shall not be smaller than `last_max_sfb` (helper variable specified in ISO/IEC 14496-3).

7.3.1.2.3.3 aac_scalable_extension_header()

max_sfb: Shall not be smaller than `last_max_sfb` (helper variable specified in ISO/IEC 14496-3).

7.4 AAC (main, LC, ER LC, SSR, LTP, ER LTP, ER LD, scalable, ER scalable)

7.4.1 Compressed data

7.4.1.1 Characteristics

Encoders may apply restrictions to the following parameters of the compressed data:

7.4.1.1.1 AudioSpecificConfig

- a) `samplingFrequencyIndex`
- b) `samplingFrequency`
- c) `channelConfiguration`
- d) `program_config_element()`

- e) `frameLengthFlag`
- f) `dependsOnCoreCoder`
- g) `extensionFlag`
- h) `epConfig`
- i) `ErrorProtectionSpecificConfig()`
- j) `aacSectionDataResilienceFlag`
- k) `aacScalefactorDataResilienceFlag`
- l) `aacSpectralDataResilienceFlag`

7.4.1.1.2 Bitstream payload

- a) use of prediction in main profile
- b) `pulse_data`
- c) `window_shape`
- d) M/S stereo
- e) intensity stereo
- f) TNS
- g) `data_stream_element()`
- h) dependently switched coupling channel
- i) independently switched coupling channel
- j) LFE channel
- k) matrix-downmix

7.4.1.2 Test procedure

Each compressed data shall meet the syntactic and semantic requirements specified in ISO/IEC 14496-3. This subclause describes a set of semantic tests to be performed on decoder relevant data. The procedure to verify whether the syntax is correct is straightforward and therefore not defined in this subclause. In the description of the semantic tests it is assumed that the tested compressed data contains no errors due to transmission or other causes. For each test the condition or conditions that shall be satisfied are given, as well as the prerequisites or conditions in which the test can be applied.

7.4.1.2.1 AudioSpecificConfig

7.4.1.2.1.1 AudioSpecificConfig()

audioObjectType: Shall be encoded according to the AAC object type (see Table 21 and Table 22).

samplingFrequencyIndex: Shall be encoded with the values specified in Table 9, Table 10, Table 11, Table 12, Table 13 and Table 14.

samplingFrequency: Shall be encoded with the values specified in Table 9.

extensionAudioObjectType: Shall be the Audio Object Type SBR (AOT == 5).

extensionSamplingFrequency: Shall be encoded with a value listed in Table 9, and the value shall be the same as `samplingFrequency`, or twice the value of `samplingFrequency`.

extensionSamplingFrequencyIndex: Shall be encoded with a value listed in Table 9, and the value shall indicate an extensionSamplingFrequency being the same as samplingFrequency as indicated by samplingFrequencyIndex, or the value shall indicate an extensionSamplingFrequency being twice the value of samplingFrequency.

sbrPresentFlag: Shall be encoded with the value zero if no SBR data is contained in the compressed MPEG-4 data. If SBR data is present in the compressed MPEG-4 data the parameter shall be encoded with the value one.

Table 9 — Specification of samplingFrequencyIndex and samplingFrequency for Scalable and Main Profile

SamplingFrequencyIndex / SamplingFrequency	Level 1	Level 2	Level 3	Level 4
Scalable Profile	0x6..0xc, 0xf / ≤ 24000		0x3..0xc, 0xf / ≤ 48000	
Main Profile	0x0..0xc, 0xf / no limitation			

Table 10 - Specification of samplingFrequencyIndex and samplingFrequency for High Quality Audio and Low Delay Audio Profile

SamplingFrequencyIndex / SamplingFrequency	Level 1,5	Level 2,6	Level 3,7	Level 4,8
High Quality Audio Profile	0x7..0xc, 0xf / ≤ 22050	0x3..0xc, 0xf / ≤ 48000		
Low Delay Audio Profile	0xb..0xc, 0xf / ≤ 8000	0x8..0xc, 0xf / ≤ 16000	0x3..0xc 0xf / ≤ 48000	

Table 11 - Specification of samplingFrequencyIndex and samplingFrequency for Natural Audio Profile

SamplingFrequencyIndex/ SamplingFrequency	Level 1,3	Level 2,6
Natural Audio Profile	0x3..0xc 0xf / ≤ 48000	0x0..0xc, 0xf / ≤ 96000

Table 12 - Specification of samplingFrequencyIndex and samplingFrequency for Mobile Audio Internet Working Profile

SamplingFrequencyIndex/ SamplingFrequency	Level 1,4	Level 2,5	Level 3,6
Mobile Audio Internet Working Profile	0x6..0xc, 0xf / ≤ 24000	0x3..0xc, 0xf / ≤ 48000	

Table 13 - Specification of samplingFrequencyIndex and samplingFrequency for AAC Profile

SamplingFrequencyIndex / SamplingFrequency	Level 1	Level 2	Level 3	Level 4	Level 5
AAC Profile	0x6..0xc, 0xf / ≤ 24000	0x3..0xc, 0xf / ≤ 48000	NA	0x3..0xc, 0xf / ≤ 48000	0x0..0xc, 0xf / ≤ 96000

Table 14 - Specification of samplingFrequencyIndex and samplingFrequency for High Efficiency AAC Profile

samplingFrequencyIndex / samplingFrequency		Level 1	Level 2	Level 3	Level 4, Level 6	Level 5, Level 7
High Efficiency AAC Profile	SBR present	NA	0x6..0xc, 0xf / ≤ 24000	0x3..0xc, 0xf / ≤ 48000	0x3..0xc, 0xf / ≤ 48000 ^a	0x3..0xc, 0xf / ≤ 48000
	SBR not present	NA	0x3..0xc, 0xf / ≤ 48000	0x3..0xc, 0xf / ≤ 48000	0x3..0xc, 0xf / ≤ 48000	0x0..0xc, 0xf / ≤ 96000

^a For Level 4, for one or two channels the maximum AAC sampling rate, with SBR present, is 48 kHz. For more than two channels the maximum AAC sampling rate, with SBR present, is 24 kHz. (0x6..0xc, 0xf / ≤ 24000)

Table 15 - Specification of extensionSamplingFrequencyIndex and extensionSamplingFrequency for High Efficiency AAC Profile

extensionSamplingFrequencyIndex / extensionSamplingFrequency	Level 1	Level 2	Level 3,4,6	Level 5,7
High Efficiency AAC Profile	NA	0x6..0xc, 0xf / ≤ 24000	0x3..0xc, 0xf / ≤ 48000	0x0..0xc, 0xf / ≤ 96000

channelConfiguration: shall be encoded with the values specified in Table 16. In the case of channelConfiguration=0, the following restrictions apply to the number of syntactic elements specified in the program_config_element():

- the number of main audio channels (represented by SCE and CPE) shall not exceed the maximum number specified for a certain profile and level.
- the number of remaining audio channels (represented by LFE and CCE) shall not exceed the maximum number specified for a certain AudioObjectType and number of main audio channels (see ISO/IEC 14496-3:2019, subclause 1.5.2.3).

Table 16 — Specification of ChannelConfiguration for Scalable and Main Profile

ChannelConfiguration	Level 1	Level 2	Level 3	Level 4
Scalable Profile	0, 1	0..2		0..7
Main Profile	0..7			

Table 17 – Specification of ChannelConfiguration for High Quality and Low Delay Audio Profile

ChannelConfiguration	Level 1, 5	Level 2, 6	Level 3, 7	Level 4, 8
High Quality Audio Profile	0,1	0..2	0..6	
Low Delay Audio Profile	0,1			0..2

Table 18 - Specification of ChannelConfiguration for Natural Audio Profile

ChannelConfiguration	Level 1, 2, 3, 4
Natural Audio Profile	0..7

Table 19 - Specification of ChannelConfiguration for Mobile Audio Internet Working Profile

ChannelConfiguration	Level 1, 4	Level 2, 5	Level 3, 6
Mobile Audio Internet Working Profile	0, 1	0..2	0..6

Table 20 - Specification of ChannelConfiguration for AAC and High Efficiency AAC Profile

ChannelConfiguration	Level 1	Level 2	Level 3	Level 4,5	Level 6,7
AAC Profile	0..2	0..2	NA	0..6	0..7, 11, 12, 14
High Efficiency AAC Profile	NA	0..2	0..2	0..6	0..7, 11, 12, 14

In addition to Table 16, Table 17, Table 18, Table 19 and Table 20 the following audioObjectType based restrictions apply:

- channelConfiguration=0 is permitted only for the audioObjectTypes 1 (AAC main), 2 (AAC LC), 3 (AAC SSR) and 4 (AAC LTP), but not for the audioObjectTypes 6 (AAC scalable), 17 (ER AAC LC), 19 (ER AAC LTP), 20 (ER AAC scalable) and 23 (ER AAC LD).
- channelConfiguration>2 is not permitted for audioObjectTypes 6 (AAC scalable) and 20 (ER AAC scalable).

epConfig: No restrictions apply.

directMapping: Shall be 1.

7.4.1.2.1.2 GASpecificConfig()

frameLengthFlag: Shall be zero for the following audio object types: 1, 2, 3, 4, 17, 19, when used in Scalable Audio Profile, Main Audio Profile, High Quality Audio Profile, Natural Audio Profile or Mobile Audio Internet Working Profile. No restrictions apply otherwise.

dependsOnCoreCoder: Shall be encoded with the value 1 in the first AAC scalable coding layer (audio object type 6 or 20) if a core coder is used in the underlying base layer of a scalable AAC configuration; shall be encoded with the value 0 otherwise.

coreCoderDelay: no restrictions apply.

extensionFlag: shall be encoded with the value 0 in the case of the audioObjectTypes 1, 2, 3, 4, 6. Shall be encoded with the value 1 in the case of the audioObjectTypes 17, 19, 20, 23.

extensionFlag3: Shall be encoded with the value 0.

7.4.1.2.1.3 ELDSpecificConfig()

frameLengthFlag: no restrictions apply.

aacSectionDataResilienceFlag: no restrictions apply

aacScalefactorDataResilienceFlag: no restrictions apply

aacSpectralDataResilienceFlag: no restrictions apply

ldSbrPresentFlag: no restrictions apply

ldSbrSamplingRate: no restrictions apply

IdSbrCrcFlag: no restrictions apply

eldExtType: no restrictions apply

eldExtLenAdd: no restrictions apply

eldExtLenAddAdd: no restrictions apply

other_byte: no restrictions apply

7.4.1.2.1.4 **program_config_element()**

No program may contain more main audio channels, LFE channels, independent coupling_channel_element()'s and dependent coupling_channel_element()'s than specified by the profile and level.

The following restrictions apply to the elements of program_config_element():

element_instance_tag: no restrictions

object_type: shall match the AudioObjectType within AudioSpecificConfig

sampling_frequency_index: shall match the samplingFrequencyIndex within AudioSpecificConfig

num_front_channel_elements: see restriction regarding the number of channels as stated above.

num_side_channel_elements: see restriction regarding the number of channels as stated above.

num_back_channel_elements: see restriction regarding the number of channels as stated above.

num_lfe_channel_elements: see restriction regarding the number of channels as stated above.

num_assoc_data_elements: no restrictions apply.

num_valid_cc_elements: see restriction regarding the number of channels as stated above.

mono_mixdown_present: shall be 0 for the audio object types 1 (AAC main), 2 (AAC LC), 3 (AAC SSR) and 4 (AAC LTP) when used in Scalable Audio Profile, Main Audio Profile, High Quality Audio Profile or Natural Audio Profile.

mono_mixdown_element_number: shall be encoded with the element_instance_tag of a single_channel_element().

stereo_mixdown_present: shall be 0 for the audio object types 1 (AAC main), 2 (AAC LC), 3 (AAC SSR) and 4 (AAC LTP) when used in Scalable Audio Profile, Main Audio Profile, High Quality Audio Profile or Natural Audio Profile.

stereo_mixdown_element_number: shall be encoded with the element_instance_tag of a channel_pair_element.

matrix_mixdown_idx_present: may only be encoded with a value of 1 if a 3 front/2 rear 5-channel program (with or without LFE) is indicated for this PCE.

matrix_mixdown_idx: no restrictions apply.

pseudo_surround_enable: no restrictions apply.

front_element_is_cpe[i]: no restrictions apply.

front_element_tag_select: shall be encoded with the element_instance_tag of either a single_channel_element or a channel_pair_element.

side_element_is_cpe[i]: no restrictions apply.

side_element_tag_select: shall be encoded with the element_instance_tag of either a single_channel_element or a channel_pair_element.

back_element_is_cpe[i]: no restrictions apply.

back_element_tag_select: shall be encoded with the element_instance_tag of either a single_channel_element or a channel_pair_element.

lfe_element_tag_select: shall be encoded with the element_instance_tag of a lfe_channel_element.

assoc_data_element_tag_select: shall be encoded with the element_instance_tag of a data_stream_element.

cc_element_is_ind_sw: shall be encoded with the same value as the ind_sw_cce_flag field of the coupling_channel_element corresponding to valid_cc_element_tag_select.

valid_cce_element_tag_select: shall be encoded with the element_instance_tag of a coupling_channel_element.

comment_field_bytes: no restrictions apply.

comment_field_data[i]: AAC, HE-AAC, HE-AACv2 and Extended HE-AAC Profile decoders of Level 6 and 7 shall support parsing the comment_field_data and interpret a **height_extension_element()** embedded therein, else no restrictions apply.

7.4.1.2.1.5 ErrorProtectionSpecificConfig()

number_of_concatenated_frame: Shall be one.

For details see also Clause 8.

7.4.1.2.2 Bitstream payload

7.4.1.2.2.1 raw_data_block()

id_syn_ele: if a program_config_element() (PCE) is present, it shall be the first syntactic element in a raw_data_block(), indicated by id_syn_ele encoded with a value of ID_PCE

7.4.1.2.2.2 Any syntactic element

element_instance_tag: ensure that element_instance_tag numbers within each element type are unique within each frame. This restriction does not apply to data_stream_element()'s (DSE), which may have duplicated element_instance_tags.

7.4.1.2.2.3 channel_pair_element()

common_window: no restrictions apply.

ms_mask_present: shall not be encoded with the binary value 11. Shall not be encoded with the binary value 10 if intensity stereo is used, i.e., if the Huffman codebook INTENSITY_HCB or INTENSITY_HCB2 is signaled in the same channel pair of the same frame.

ms_used: no restrictions apply.

7.4.1.2.2.4 ics_info()

ics_reserved_bit: shall be set to zero.

window_sequence: Shall be zero (ONLY_LONG_SEQUENCE) if audioObjectType == 23, no such restriction applies for the remaining object types. The meaningful window_sequence transitions are as follows:

from ONLY_LONG_SEQUENCE to	{ ONLY_LONG_SEQUENCE LONG_START_SEQUENCE
from LONG_START_SEQUENCE to	{ EIGHT_SHORT_SEQUENCE LONG_STOP_SEQUENCE
from LONG_STOP_SEQUENCE to	{ ONLY_LONG_SEQUENCE LONG_START_SEQUENCE
from EIGHT_SHORT_SEQUENCE to	{ EIGHT_SHORT_SEQUENCE LONG_STOP_SEQUENCE

Other, non-meaningful, window_sequence transitions are also possible:

from ONLY_LONG_SEQUENCE to	{ EIGHT_SHORT_SEQUENCE LONG_STOP_SEQUENCE
from LONG_START_SEQUENCE to	{ ONLY_LONG_SEQUENCE LONG_START_SEQUENCE
from LONG_STOP_SEQUENCE to	{ EIGHT_SHORT_SEQUENCE LONG_STOP_SEQUENCE
from EIGHT_SHORT_SEQUENCE to	{ ONLY_LONG_SEQUENCE LONG_START_SEQUENCE

A conformant bitstream shall consist of only meaningful window_sequence transitions. However, decoders are required to handle non-meaningful window_sequence transitions as well. Test sequences al03 and as17 are provided respectively for AOT 2 (AAC LC) and AOT 3 (AAC SSR) to test decoder performance on non-meaningful window sequence transitions (AOT 1 (AAC Main) and AOT 4 (AAC LTP) decoders also shall fulfil conformance for AOT 2) (see 7.4.1.2.2.1). The performance requirements for non-meaningful window_sequence transitions are the same as for the meaningful transitions.

window_shape: no restrictions apply.

max_sfb: shall be $\leq$ num_swb_long or num_swb_short as appropriate for window_sequence and sampling frequency.

scale_factor_grouping: no restrictions apply.

predictor_data_present: shall be encoded with the value 0 for the audioObjectTypes 2 (AAC LC), 3 (AAC SSR) and 17 (ER AAC LC); shall be encoded with the value 0 when used in the Low Delay AAC profile; no restrictions apply otherwise.

predictor_reset: shall be encoded with the binary value of 1 sufficiently often so that normative behaviour is achieved (AAC main).

predictor_reset_group_number: shall not be encoded with the binary values 00000 or 11111 (AAC main).

prediction_data_used: no restrictions apply.

ltp_data_present: No restrictions apply.

7.4.1.2.2.5 pulse_data()

number_pulse: no restrictions apply.

pulse_start_sfb: shall be smaller than num_swb_long_window[fs_index].

pulse_offset[i]: swb_offset_long_window[pulse_start_sfb] + pulse_offset[0] + ... + pulse_offset[number_pulse] shall not be greater than 1023.

pulse_amp[i]: shall be encoded with a value small enough such that the compensated quantized spectral coefficient is not greater than 8191.

7.4.1.2.2.6 coupling_channel_element()

The number of dependently-switched and independently-switched coupling channel elements shall not exceed the allowed numbers specified by the profile and level. No coupling channel shall target a given single_channel_element() or channel_pair_element() more than once per frame. Dependently switched coupling channels are not permitted for audio object type 4 (AAC LTP).

ind_sw_cce_flag: shall not be encoded with the binary value of 1 if independently-switched coupling channel elements are not specified by the level and profile.

num_coupled_elements: shall not be encoded with a value greater than the total number of single_channel_elements and channel_pair_elements.

cc_target_is_cpe: shall be encoded with the binary value 1 if the syntactic element with element_instance_tag of cc_target_tag_select is a channel_pair_element; otherwise, it shall be encoded with the binary value of 0.

cc_target_tag_select: shall only be encoded with a binary value equal to the element_instance_tag of a single_channel_element or a channel_pair_element of the current frame.

cc_l: no restrictions apply.

cc_r: no restrictions apply.

cc_domain: no restrictions apply.

gain_element_sign: no restrictions apply.

gain_element_scale: no restrictions apply.

common_gain_element_present: no restrictions apply.

hcod_sf: see 7.4.1.2.2.17.

7.4.1.2.2.7 lfe_channel_element()

The number of LFEs shall not exceed the allowed number specified by the profile & level.

The **window_shape** field of any LFE shall always be encoded with a value of 0 (sine window).

The **window_sequence** field of any LFE shall always be encoded with a value of ONLY_LONG_SEQUENCE.

Only the lowest 12 spectral coefficients of any LFE may be non-zero.

The **predictor_data_present_flag** of any LFE shall be encoded with a value of 0.

Temporal noise shaping shall not be used in any LFE.

7.4.1.2.2.8 data_stream_element()**data_byte_align_flag:** no restrictions apply.**count:** no restrictions apply.**esc_count:** no restrictions apply.**dat_stream_byte:** no restrictions apply.**7.4.1.2.2.9 fill_element()****count:** no restrictions apply.**esc_count:** no restrictions apply.

Fill elements containing an extension_payload with an extension_type of EXT_SBR_DATA or EXT_SBR_DATA_CRC shall not contain any other extension_payload of any other extension_type. For fill elements containing an extension_payload with an extension_type of EXT_SBR_DATA or EXT_SBR_DATA_CRC, the fill_element count field shall be set equal to the total length in bytes, including the SBR enhancement data plus the extension_type field.

7.4.1.2.2.10 gain_control_data()

For the audio object type AAC SSR the following restrictions apply:

alocode: shall satisfy the following conditions:

$$\text{alocode}[B][w][m_1] < \text{alocode}[B][w][m_2], 1 \leq m_1 < m_2 \leq \text{adjust_num}[B][w] + 1$$

where B is the Band ID, an integer between 1 and 3, and w is the Window ID, an integer from 0 to 7.

No restrictions apply for the remaining data elements inside of gain_control_data().

7.4.1.2.2.11 aac_scalable_main_header()**ics_reserved_bit:** see 7.4.1.2.2.4.**window_sequence:** see 7.4.1.2.2.4.**window_shape:** see 7.4.1.2.2.4.**max_sfb:** see 7.4.1.2.2.4.**scale_factor_grouping:** see 7.4.1.2.2.4.**ms_mask_present:** see 7.4.1.2.2.4.**tns_channel_mono_layer:** no restrictions apply.**tns_data_present:** see 7.4.1.2.2.13.**ltp_data_present:** Shall be zero if audioObjectType == 20 (ER AAC scalable). No restrictions apply otherwise.**7.4.1.2.2.12 aac_scalable_extension_header()****max_sfb:** see 7.4.1.2.2.4.**ms_mask_present:** see 7.4.1.2.2.4.

tns_data_present: see 7.4.1.2.2.13.

7.4.1.2.2.13 **diff_control_data()**

diff_control: no restrictions apply.

7.4.1.2.2.14 **diff_control_lr()**

diff_control_lr: no restrictions apply.

7.4.1.2.2.15 **individual_channel_stream()**

global_gain: no restrictions apply.

pulse_data_present: shall be encoded with a value of 0 for the audioObjectTypes 6 (AAC scalable) and 20 (ER AAC scalable); shall be encoded with a value of 0 when used in the Low Delay AAC profile; shall be encoded with a value of 0 if window_sequence is EIGHT_SHORT_SEQUENCE; no restrictions apply otherwise.

tns_data_present: no restrictions apply.

gain_control_data_present: no restrictions apply for AAC SSR; otherwise it shall be encoded with the value 0.

length_of_reordered_spectral_data: Shall be equal to the length of the reordered spectral data. In case of a SCE or LFE it shall be ≤ 6144 . In case of CPE the sum of both values shall be ≤ 12288 .

length_of_longest_codeword: Shall reflect the length of the longest codeword transmitted within the current frame. It shall be ≤ 48 .

7.4.1.2.2.16 **section_data()**

sect_cb[g][i]: Shall not be encoded with the decimal value 12 (bit sequence either "1100" (aacSectionDataResilienceFlag == 0) or "01100" (aacSectionDataResilienceFlag == 1)).

Intensity codebooks INTENSITY_HCB and INTENSITY_HCB2 shall not occur in a single_channel_element, the left channel of a channel pair element, a coupling channel element, or an LFE. Intensity codebooks can only occur in a channel_pair_element if the common_window field is set to 1.

Given that ms_used[g][sfb] is set to 1 or ms_mask_present equals the binary value 10, sfb_cb[g][sfb] shall not equal NOISE_HCB in only one channel of a channel pair element.

sect_len_incr: The sum of all sect_len_incr elements for a given window group shall equal max_sfb.

7.4.1.2.2.17 **scale_factor_data()**

hcod_sf[]: Shall only be encoded with the values listed in the scalefactor Huffman table. Shall be encoded such that the decoded scalefactors sf[g][sfb] are within the range of zero to 255, both inclusive.

dpcm_noise_nrg: No restrictions apply.

sf_concealment: No restrictions apply.

rev_global_gain: Shall be encoded with the PCM value of the last scale factor.

length_of_rvlc_sf: Shall be equal to the length of the RVLC data part in bits.

rvlc_cod_sf: Shall only be encoded with the values listed in the RVLC codebook table.

sf_escapes_present: No restrictions apply.

length_of_rvlc_escapes: Shall be equal to the length of the RVLC escape data part in bits.

rvlc_esc_sf: Shall only be encoded with the values listed in the Huffman codebook table for RVLC escape values.

dpcm_is_last_position: Shall be encoded with the last intensity stereo position.

dpcm_noise_last_position: Shall be encoded with the last noise energy value.

7.4.1.2.2.18 tns_data()

n_filt: no restrictions apply.

coef_res: no restrictions apply.

length[w][filt]: shall be small enough such that the lower bound of the filtered region, indicated by 'bottom', does not exceed the start of the array containing the spectral coefficients (spec[w])

order[w][filt]: shall not exceed the maximum permitted order depending on the specified object type and sampling frequency

direction: no restrictions apply.

coef_compress: no restrictions apply.

coef: no restrictions apply.

7.4.1.2.2.19 ltp_data()

No restrictions apply to any of the data elements inside ltp_data().

7.4.1.2.2.20 spectral_data()

hcod[sect_cb[g][i]][w][x][y][z]: shall only be encoded with the values listed in Huffman codebooks 1, 2, 3, or 4.

quad_sign_bits: no restrictions apply.

hcod[sect_cb[g][i]][y][z]: shall only be encoded with the values listed in Huffman codebooks 5 through 11.

pair_sign_bits: no restrictions apply.

hcod_esc_y: shall be encoded with a value smaller or equal to 8191, i.e., it shall be encoded with an initial escape sequence consisting of not more than nine '1' bits followed by an escape separator of '0'.

hcod_esc_z: shall be encoded with a value smaller or equal to 8191, i.e., it shall be encoded with an initial escape sequence consisting of not more than nine '1' bits followed by an escape separator of '0'.

7.4.1.2.2.21 extension_payload()

extension_type: no restrictions apply.

fill_nibble: shall be '0000'.

fill_byte: shall be '10100101'.

data_element_version: shall be '0000'.

dataElementLengthPart: no restrictions apply.

data_element_byte: no restrictions apply.

other_bits: no restrictions apply.

7.4.1.2.2.22 **dynamic_range_info()**

No restrictions apply to any of the data elements inside `dynamic_range_info()`.

7.4.1.2.2.23 **excluded_channels()**

No restrictions apply to any of the data elements inside `excluded_channels()`.

7.4.1.2.2.24 **ms_data()**

ms_used: see 7.4.1.2.2.3.

7.4.1.2.2.25 **channel_pair_element_eld()**

max_sfb: shall be $\leq \text{num_swb_long}$ as appropriate for sampling frequency.

ms_mask_present: shall not be encoded with the binary value 11

ms_used: no restrictions apply.

7.4.1.2.2.26 **individual_channel_stream_eld()**

global_gain: no restrictions apply

max_sfb: shall be $\leq \text{num_swb_long}$ as appropriate for sampling frequency.

tns_data_present: no restrictions apply

length_of_reordered_spectral_data: Shall be equal to the length of the reordered spectral data. In case of a SCE or LFE it shall be ≤ 6144 . In case of CPE the sum of both values shall be ≤ 12288 .

length_of_longest_codeword: Shall reflect the length of the longest codeword transmitted within the current frame. It shall be ≤ 48 .

7.4.2 Decoders

7.4.2.1 Characteristics

The object types AAC LC (Low Complexity), AAC main, AAC SSR (Scalable Sampling Rate) and AAC LTP (Long Term Prediction) build the basic object types supporting AAC-based audio coding within MPEG-4 using the ISO/IEC 13818-7 style syntax. The AAC Scalable Object type is built on top of the AAC LTP object type, but uses a different decoder structure, syntax and additional tools to provide large step scalability. The details are given in Table 21.

The AAC LC, AAC main and AAC SSR object types correspond to the LC, Main, SSR profiles of ISO/IEC 13818-7, with the inclusion of PNS as a mandatory tool in MPEG-4 AAC decoders. The AAC main and AAC LTP object types are built on top of the AAC LC object type. The AAC SSR is identical to the AAC LC object type with the exception of the filterbank, the additional gain control tool and some aspects of the TNS tool configuration. All these object types have an ISO/IEC 13818-7 syntax style.

Table 21 — AAC Object Types

Audio Object Type	GA Bitstream Syntax Type	Hierarchy	Object Type ID
AAC main	ISO/IEC 13818-7 Style	contains AAC LC	1
AAC LC	ISO/IEC 13818-7 Style		2
AAC SSR	ISO/IEC 13818-7 Style		3
AAC LTP	ISO/IEC 13818-7 Style	contains AAC LC	4
AAC scalable	Scalable		6

A number of ER AAC object types have been defined in ISO/IEC 14496-3. Table 22 gives an overview.

Table 22 — Overview about the ER AAC object types

<u>Audio Object Type</u>	Object Type ID	13818-7 LC	PNS	LTP	TLSS	Low Delay AAC	Error Robust	Low Delay SBR
ER AAC LC	17	X	X				X	
ER AAC LTP	19	X	X	X			X	
ER AAC scalable	20	X	X		X		X	
ER AAC LD	23		X	X		X	X	
ER AAC ELD	39		X			X	X	X

The object types ER AAC LC, ER AAC LTP, and ER AAC scalable are based on the object types AAC LC, AAC LTP, and AAC scalable respectively as defined in ISO/IEC 14496-3. The object type ER AAC LD is based on the object type ER AAC LTP, but introduces some changes in order to reduce the overall algorithmic delay. Table 23 shows these dependencies.

Table 23 — AAC object type dependencies

ER Audio object type	underlying Audio object type
ER AAC LC	AAC LC
ER AAC LTP	AAC LTP
ER AAC scalable	AAC scalable
ER AAC LD	ER AAC LTP

All ER AAC object types use the error resilient bitstream payload syntax. This syntax is based on the syntax of the underlying non-ER AAC object types. The error resilient bitstream payload can be derived by subdivision of the bitstream payload data elements into instances of error sensitivity classes.

The error resilient bitstream payload is mandatory.

Beside the error resilient bitstream syntax, modified noiseless coding tools are introduced for section data, scale factor data, and spectral data. These tools are optional.

In general, conformance criteria defined for the underlying object types are also valid for the Version 2 object types and will not be repeated here. Thus, characteristics defined for a new object type have to be treated as extensions or modifications with respect to the already defined characteristics of the underlying object type.

A compliant decoder may also support any of the modifications to the parameters in an audio bitstream described in Table 24.

Table 24 — AAC Parameter

Bitstream Characteristic	Variation
program configuration	any configuration of compressed data containing more than one program (in the sense of what is specified in a <code>program_config_element()</code>) is not allowed if a <code>program_config_element()</code> is used, syntactic elements (other than <code>ID_FILL</code> or <code>ID_END</code>) not referenced by any <code>program_config_element()</code> are not allowed
data_stream_element	a decoder is not required to store or present data recovered from <code>data_stream_element()</code> 's
mono-mixdown element	a decoder conforming with one of the currently defined profiles is not required to support compressed data containing any mono-mixdown element
stereo-mixdown element	a decoder conforming with one of the currently defined profiles is not required to support compressed data containing any stereo-mixdown element
matrix-mixdown	a decoder is not required to calculate a matrix-mixdown signal

The channel configuration information given in a `program_config_element()` inside an AAC payload conveyed as an MPEG-4 access unit shall be ignored. In this case, the channel configuration information is already available from the MPEG-4 decoder configuration, specifically the `GASpecificConfig()`, which can include a `program_config_element()`. The channel configuration information given in a `program_config_element()` inside an AAC payload is only evaluated in case of an MPEG-4 ADTS bitstream with `channel_configuration == 0`, since such a bitstream does not include a `GASpecificConfig()`.

Next to channel configuration information, a `program_config_element()` can carry additional information, e.g. coefficients for a matrix mixdown or a comment field. This additional information can be conveyed dynamically by a `program_config_element()` inside an MPEG-4 access unit as well as by a `program_config_element()` included in the `GASpecificConfig()`. Such information does not affect the normative behavior of a decoder and may hence be ignored. However it may be utilized for non-normative decoder operation modes like matrix mixdown.

7.4.2.1.1 AAC main

The MPEG-4 AAC main object type is the counterpart to the MPEG-2 AAC Main Profile, though also offering the PNS tool. The AAC main object type bitstream syntax is compatible with the syntax defined in ISO/IEC 13818-7. All the MPEG-2 AAC multi-channel capabilities are available. A decoder capable of decoding a MPEG-4 Main Access Unit can also parse and decode an MPEG-2 AAC Main Profile raw_data_stream(). On the other hand, an MPEG-2 Main profile decoder will not be able to parse an MPEG-4 AAC Main stream if PNS has been used. The AAC main Object Type is an extension of the AAC LC Object Type.

7.4.2.1.2 AAC LC

The MPEG-4 AAC Low Complexity (LC) object type is the counterpart to the MPEG-2 AAC Low Complexity Profile, though also offering the PNS tool. The AAC LC object type bitstream syntax is compatible with the syntax defined in ISO/IEC 13818-7. All the MPEG-2 AAC multi-channel capabilities are available. A decoder capable of decoding an MPEG-4 LC Access Unit can also parse and decode an MPEG-2 AAC LC Profile raw_data_stream(). On the other hand, an MPEG-2 AAC LC profile decoder will not be able to parse an MPEG-4 AAC-LC stream if PNS has been used.

7.4.2.1.3 AAC SSR

The MPEG-4 AAC Scalable Sampling Rate (SSR) object type is the counterpart to the MPEG-2 AAC SSR Profile, though also offering the PNS tool. The AAC SSR object type bitstream syntax is compatible with the syntax defined in ISO/IEC 13818-7. All the MPEG-2 SSR multi-channel capabilities are available. A decoder capable of decoding a MPEG-4 SSR Access Unit can also parse and decode a MPEG-2 SSR Main profile raw data stream. On the other hand, an MPEG-2 SSR profile decoder will not be able to parse an MPEG-4 AAC- SSR stream if PNS has been used.

7.4.2.1.4 AAC LTP

The AAC LTP Object Type is an extension of the AAC LC Object Type with a long term predictor. At the same time, the MPEG-4 AAC LTP object type is similar to the AAC main object type. However, an LTP replaces the MPEG-2 AAC predictor and the PNS tool can be used in addition. The LTP achieves a similar coding gain, but requires significantly lower implementation complexity. The bitstream syntax for this object type is very similar to the syntax defined in ISO/IEC 13818-7. An MPEG-2 AAC LC profile bitstream can be decoded without restrictions by an LTP decoder.

The decoder shall use the MPEG-4 long term predictor.

7.4.2.1.5 AAC scalable

The scalable AAC object type is built on top of the AAC LTP object type, but uses a different bitstream syntax, decoder structure and additional tools to support bitrate- and bandwidth- scalability. A large number of scalable combinations are available, including combinations with TwinVQ and CELP coder tools. However, only mono or 2-channel stereo objects are supported.

AAC-based scalable configurations shall support all object type combinations specified in ISO/IEC 14496-3. All AAC and TwinVQ layers and their enhancement layers shall operate at the same sampling rate. In case of using a CELP core coder the ratio between CELP core and AAC enhancement layer sampling rates is restricted according to the specification in the General Audio part of ISO/IEC 14496-3.

7.4.2.2 Test procedure

The test procedures specified in Table 28 has to be applied. The RMS test procedure always includes the LSB test. Table 25 provides the references to the according test specifications.

Table 25 — References to the test procedure descriptions

Name of the test procedure as used in Table 28	Reference to the test specification
RMS	7.1.2.2.1
PNS	7.1.2.2.4

If no test is specified, a check of conformance using appropriate measurements, e.g. the LSB criterion (for those sequences that do not utilize PNS) or objective perceptual measurement systems, is not mandatory but highly recommended. This also applies to bitstreams with non-meaningful window sequences.

7.4.2.3 Test sequences

To test AAC decoders, ISO/IEC JTC 1/SC 29/WG 11 supplies a number of test sequences. The test sequences are defined in Table 26, Table 27 and Table 28. In the case that the ChannelConfiguration equals zero the program_config_element() is defined in Table 29. Sequences are provided at sampling rates of 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 64, 88.2, and 96 kHz for the audio object types AAC main, AAC LC, AAC SSR, AAC scalable, ER AAC LC, ER AAC LTP and ER AAC scalable and at sampling rates of 22.05, 24, 32, 44.1 and 48 kHz for the audio object type ER AAC LD and ER AAC ELD. The extension _fs is appended to the sequence name to indicate the sampling rate of the test sequence. Possible values of fs are 08, 11, 12, 16, 22, 24, 32, 44, 48, 64, 88 and 96, corresponding to the possibly non-integer sampling rates listed above. If two bitrates are listed in the table for a certain sequence, the lower bitrate is to be used for sampling rates of 16 kHz and below, and the higher bitrate is to be used at sampling rates above 16 kHz.

All sequences for the object types AAC main, AAC LC, AAC SSR, AAC scalable, ER AAC LC, ER AAC LTP, ER AAC LD and ER AAC ELD are supplied in all sampling rates (as indicated by extension _fs). For a specific profile and level only the sequences with the appropriate channel configuration and sampling rate are applicable.

Some conformance test sequences have special properties as follows:

Dynamic Range Control: This field indicates that dynamic range control information is available in the bitstream payload of the compressed data. Conformant decoders shall be able to parse these test sequences. However, DRC semantics is optional, making the result of decoding the DRC test sequences informative only.

Arithmetic torture: This field indicates that as many different Huffman codewords from the spectrum Huffman codebooks as possible are used within the bitstream payload of the compressed data. At least 95 % of the total number of the individual codewords is processed.

Buffer test: This field indicates that the bitstream payload of the compressed data is intended to check the decoder's input buffer size. The number of remaining bits in the bit reservoir are first kept at a high level and then at one or several blocks within the stream all accumulated free bits are used by a single raw_data_block(). Thus, the resulting block lengths for these raw_data_block()'s are similar to the total decoder input buffer size, or, as this can be difficult to achieve, not more than 16 bit below that value.

Non-meaningful window_sequence transitions: This field indicates that the bitstream payload of the compressed data is intended to check the decoder's behavior in case of a non-meaningful window sequence transitions (see 7.4.1.2.2.1 for details).

Table 26 — AAC test sequences

file base name	content	bitrate (kbit/s)	AudioObjectType	SamplingFrequencyIndex	ChannelConfiguration	frameLengthFlag	coreCoderDelay	intensity	MS	window sequence switching	non-meaningful window _sequence transitions	window shape switching	tns_data_present	pulse data	prediction	LTP	PNS	data stream elements	gain compensation enabled	dynamic range control	bandwidth	buffer test	arithmetic torture	test procedure
am00	sine sweep	40/64	1	*	0	0	-	-	-	y	n	y	n	y	n	n	n	n	-	n	-	n	n	RMS
am01	music	40/64	1	*	0	0	-	-	-	y	n	y	n	n	y	n	n	n	-	n	-	n	n	none
am02	music	80/128	1	0.3,6..11	0	0	-	y	y	y	n	y	y	n	y	n	n	n	-	n	-	n	n	none
am02	music	128	1	4,5	0	0	-	y	y	y	n	y	y	n	y	n	n	n	-	n	-	n	y	none
am04	music	64/128	1	*	0	0	-	y	y	y	n	n	y	n	y	n	n	n	-	y	-	n	n	none
am05	music	192/384	1	*	0	0	-	y	y	y	n	n	y	n	y	n	n	n	-	y	-	n	n	none
am06	music	128/256	1	*	0	0	-	y	y	y	n	n	y	n	y	n	n	n	-	y	-	n	n	none
am07	music	200/320	1	*	0	0	-	y	y	y	n	y	y	n	y	n	n	n	-	n	-	n	y	none
al00	sine sweep	40/64	2	*	0	0	-	-	-	y	n	n	n	n	-	n	n	n	-	n	-	n	n	RMS
al01	music	40/64	2	*	0	0	-	-	-	y	n	n	n	n	-	n	n	y	-	n	-	n	n	none
al02	music	40/64	2	*	0	0	-	-	-	y	n	n	n	n	-	n	n	y	-	n	-	y	n	none
al03	music	40/64	2	*	0	0	-	-	-	y	y	n	n	n	-	n	n	y	-	n	-	n	n	none
al04	music	40/64	2	*	0	0	-	-	-	y	n	y	y	y	-	n	n	y	-	n	-	n	n	none
al05	music	80/128	2	*	0	0	-	y	y	y	n	n	n	n	-	n	n	y	-	n	-	n	n	none
al06	test mix	120/192	2	0.2,7..11	0	0	-	y	y	y	n	n	y	n	-	n	n	y	-	n	-	n	n	none
al06	test mix	192	2	3.6	0	0	-	y	y	y	n	n	y	n	-	n	n	y	-	n	-	n	y	none
al07	music, other	240/384	2	*	0	0	-	y	y	y	n	n	y	n	-	n	n	n	-	n	-	n	y	none
al08	music	1920/3072	2	0.2,7..11	0	0	-	y	y	y	n	n	y	n	-	n	n	n	-	n	-	n	n	none
al08	music	3072	2	3..6	0	0	-	y	y	y	n	n	y	n	-	n	n	n	-	n	-	n	y	none
al14	music	64/128	2	*	0	0	-	y	y	y	n	n	y	n	-	n	n	n	-	y	-	n	n	none
al16	music	96/384	2	*	0	0	-	y	y	y	n	n	y	n	-	n	n	n	-	y	-	n	n	none
al17	music	80/128	2	*	0	0	-	n	n	y	n	n	n	n	-	n	n	n	-	n	-	n	n	none
al18	noise	40/64	2	*	1	0	-	-	-	l	n	n	n	n	-	n	y	n	-	n	-	n	n	PNS-1
al19	noise	40/64	2	*	1	0	-	-	-	s	n	n	n	n	-	n	y	n	-	n	-	n	n	PNS-2/3
al20	sine	384	2	3..5	11	0	-	y	y	y	n	y	n	n	-	n	n	y	-	y	-	n	n	none
al21	sine	448	2	3..5	12	0	-	y	y	y	n	y	n	n	-	n	n	y	-	y	-	n	n	none
al22	sine	448	2	3..5	14	0	-	y	y	y	n	y	n	n	-	n	n	y	-	y	-	n	n	none
as00	sine sweep	40/64	3	*	0	0	-	-	-	y	n	n	n	n	-	n	n	n	n	n	-	y	n	RMS
as01	music	40/64	3	*	0	0	-	-	-	y	n	y	y	n	-	n	n	n	y	n	4	n	n	none
as02	music	40/64	3	0.5, 8	0	0	-	-	-	y	n	y	y	n	-	n	n	n	y	n	3	n	n	none
as02	music	40/64	3	6, 7, 9..11	0	0	-	-	-	y	n	y	y	y	-	n	n	n	y	n	3	y	n	none
as03	music	40/64	3	0.9	0	0	-	-	-	y	n	y	y	n	-	n	n	n	y	n	2	n	n	none
as03	music	40	3	10, 11	0	0	-	-	-	y	n	y	y	y	-	n	n	n	y	n	2	y	n	none
as04	music	40/64	3	*	0	0	-	-	-	y	n	y	y	y	-	n	n	n	n	n	1	y	n	none
as05	music	40/64	3	0.4,6..11	0	0	-	-	-	y	n	n	n	n	-	n	n	y	n	n	4	n	n	none
as05	music	64	3	5	0	0	-	-	-	y	n	n	n	n	-	n	n	y	n	n	4	n	y	none
as06	music	40/64	3	0.4,6..11	0	0	-	-	-	y	n	n	n	n	-	n	n	y	n	n	3	n	n	none
as06	music	64	3	5	0	0	-	-	-	y	n	n	n	n	-	n	n	y	n	n	3	n	y	none
as07	music	40/64	3	0.3,5..11	0	0	-	-	-	y	n	n	n	n	-	n	n	y	n	n	2	n	n	none
as07	music	64	3	4	0	0	-	-	-	y	n	n	n	n	-	n	n	y	n	n	2	n	y	none
as08	music	40/64	3	*	0	0	-	-	-	y	n	n	n	n	-	n	n	y	n	n	1	n	n	none
as09	music	80/128	3	0.2,5..11	0	0	-	y	y	y	n	n	n	n	-	n	n	n	y	n	4	n	n	none
as09	music	128	3	3,4	0	0	-	y	y	y	n	n	n	n	-	n	n	n	y	n	4	n	y	none
as10	music	80/128	3	*	0	0	-	y	y	y	n	n	n	n	-	n	n	n	y	n	3	n	n	none
as11	music	80/128	3	*	0	0	-	y	y	y	n	n	n	n	-	n	n	n	y	n	2	n	n	none
as12	music	80/128	3	*	0	0	-	y	y	y	n	n	n	n	-	n	n	n	n	n	1	n	n	none
as13	music	320	3	0,1	0	0	-	y	y	y	n	y	y	y	-	n	n	n	y	n	4	n	n	none
as13	music	200/320	3	2..11	0	0	-	y	y	y	n	y	y	y	-	n	n	n	y	n	4	n	y	none
as14	music	200/320	3	0,9..11	0	0	-	y	y	y	n	y	y	y	-	n	n	n	y	n	3	n	n	none

file base name	content	bitrate (kbit/s)	AudioObjectType	SamplingFrequency/Index	ChannelConfiguration	frameLengthFlag	coreCoderDelay	intensity	MS	window sequence switching	non-meaningful window _sequence transitions	window shape switching	tns_data_present	pulse data	prediction	LTP	PNS	data stream elements	gain compensation enabled	dynamic range control	bandwidth	buffer test	arithmetic torture	test procedure
as14	music	200/320	3	1.8	0	0	-	y	y	y	n	y	y	y	-	n	n	n	y	n	3	n	y	none
as15	music	280/448	3	0	0	0	-	y	y	y	n	y	y	y	-	n	n	n	y	n	4	n	n	none
as15	music	280/448	3	1..11	0	0	-	y	y	y	n	y	y	y	-	n	n	n	y	n	4	n	y	none
as16	music	280/448	3	0.8	0	0	-	y	y	y	n	y	y	y	-	n	n	n	y	n	3	n	y	none
as16	music	280	3	9..11	0	0	-	y	y	y	n	y	y	y	-	n	n	n	y	n	3	n	n	none
as17	music	40/64	3	*	0	0	-	-	-	y	y	n	n	n	-	n	n	n	y	n	4	n	n	none
ap01	sine sweep	64	4	3	0	0	-	-	-	y	n	y	n	n	-	y	n	n	-	n	-	n	n	RMS
ap02	music	128	4	3	0	0	-	n	y	y	n	y	y	n	-	y	n	n	-	n	-	n	n	none
ap03	music	64	4	3	0	0	-	-	-	y	n	y	y	n	-	y	n	n	-	n	-	n	n	none
ap04	music	64	4	3	0	0	-	-	-	y	n	y	n	n	-	y	n	n	-	n	-	n	n	none
ap05	music	128	4	3	0	0	-	y	y	y	n	y	y	n	-	y	n	n	-	n	-	n	n	none
ac01	test mix	32	6	3	1	0	-	-	-	y	n	y	?	n	-	y	n	-	-	n	-	n	n	none
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
ac02	test mix	32	6	3	1	0	-	-	-	y	n	y	?	n	-	y	n	-	-	n	-	n	n	none
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		32	6	3	2	0	-	n	?	-	-	-	n	n	-	-	n	-	-	n	-	n	n	
		32	6	3	2	0	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		32	6	3	2	0	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
ac03	test mix	64	6	3	2	0	-	-	-	y	n	y	?	n	-	y	n	-	-	n	-	n	n	none
		32	6	3	2	0	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		32	6	3	2	0	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		32	6	3	2	0	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		32	6	3	2	0	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		32	6	3	2	0	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		32	6	3	2	0	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
ac04	test mix	6	8	12	1	-	-	-	-	-	-	-	-	n	-	-	n	-	-	-	-	-	-	none
		32	6	4	1	1	8000	-	-	y	n	y	?	n	-	-	n	-	-	n	-	n	n	
		32	6	4	1	1	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		32	6	4	1	1	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		32	6	4	2	1	-	n	?	-	-	-	n	n	-	-	n	-	-	n	-	n	n	
		32	6	4	2	1	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		32	6	4	2	1	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
ac05	test mix	12.2	8	11	1	-	-	-	-	-	-	-	-	n	-	-	n	-	-	-	-	-	-	none
		2	8	11	1	1	-	-	-	-	-	-	-	n	-	-	n	-	-	-	-	-	-	
		16	6	3	1	1	0	-	-	y	n	y	?	n	-	-	n	-	-	n	-	n	n	
		16	6	3	1	1	-	-	-	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		16	6	3	1	1	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		32	6	3	2	1	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		32	6	3	2	1	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n	n	
		32	6	3	2	1	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n	n	

ISO/IEC 14496-26:2024(en)

file base name	content	bitrate (kbit/s)	AudioObjectType	SamplingFrequencyIndex	ChannelConfiguration	frameLengthFlag	coreCoderDelay	intensity	MS	window sequence switching	non-meaningful window _sequence transitions	window shape switching	tns_data_present	pulse data	prediction	LTP	PNS	data stream elements	gain compensation enabled	dynamic range control	bandwidth	buffer test	arithmetic torture	test procedure	
ac06	test mix	12.2	8	12	1	-	-	-	-	-	-	-	-	n	-	-	n	-	-	-	-	-	-	none	
		2	8	12	1	1	-	-	-	-	-	-	-	n	-	-	n	-	-	-	-	-	-		
		32	6	7	2	1	798	n	?	y	n	y	?	n	-	-	n	-	-	n	-	n	n		
		32	6	7	2	1	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n	n		
		32	6	7	2	1	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n	n		
		32	6	7	2	1	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n	n		
ac08	test mix	16	7	3	1	0	-	-	-	y	n	y	n	n	-	y	n	-	-	-	-	-	-	none	
		16	6	3	1	0	-	-	-	-	-	-	y	n	-	-	n	-	-	-	n	-	n		n
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	-	n	-	n		n
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	-	n	-	n		n
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	-	n	-	n		n
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	-	n	-	n		n
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	-	n	-	n		n
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	-	n	-	n		n
ac09	test mix	16	7	3	1	0	-	-	-	y	n	y	n	n	-	n	n	-	-	-	-	-	-	none	
		16	6	3	1	0	-	-	-	-	-	-	y	n	-	-	n	-	-	-	n	-	n		n
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	-	n	-	n		n
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	-	n	-	n		n
		16	6	3	1	0	-	-	-	-	-	-	-	n	-	-	n	-	-	-	n	-	n		n
		32	6	3	2	0	-	n	?	-	-	-	y	n	-	-	n	-	-	-	n	-	n		n
		32	6	3	2	0	-	n	?	-	-	-	-	n	-	-	n	-	-	-	n	-	n		n
		32	6	3	2	0	-	n	?	-	-	-	-	n	-	-	n	-	-	-	n	-	n		n
ac10	test mix	16	7	7	1	0	-	-	-	y	n	y	n	n	-	n	n	-	-	-	-	-	-	none	
		32	6	7	2	0	-	n	?	-	-	-	y	n	-	-	n	-	-	-	n	-	n		n
		32	6	7	2	0	-	n	?	-	-	-	-	n	-	-	n	-	-	-	n	-	n		n
		32	6	7	2	0	-	n	?	-	-	-	-	n	-	-	n	-	-	-	n	-	n		n
		32	6	7	2	0	-	n	?	-	-	-	-	n	-	-	n	-	-	-	n	-	n		n
ac12	test mix	6	8	11	1	1	-	-	-	-	-	-	n	-	-	n	-	-	-	-	-	-	none		
		16	6	11	1	1	946	n	?	y	n	y	y	n	-	-	n	-	-	-	n	-		n	n
ac13	test mix	6 32	8 6	11 6	1 1	1 1	- 798	- -	- -	- y	- n	- y	- y	n n	- -	- -	n n	- -	- -	- n	- -	- n	- n	none	
ac14	test mix	6	8	11	1	1	-	-	-	-	-	-	-	n	-	-	n	-	-	-	-	-	none		
		64	6	5	2	1	404	n	?	y	n	y	?	n	-	-	n	-	-	n	-	n		n	
		128	6	5	2	1	-	n	?	-	-	-	-	n	-	-	n	-	-	n	-	n		n	
ac15	test mix	6	8	11	1	1	-	-	-	-	-	-	-	n	-	-	n	-	-	-	-	-	none		
		64	6	0	1	1	0	-	-	y	n	y	?	n	-	-	n	-	-	n	-	n		n	
		128	6	0	2	1	-	n	?	-	-	-	n	n	-	-	n	-	-	n	-	n		n	
ac16	test mix	8	7	9	1	0	-	-	-	y	n	y	n	n	-	y	n	-	-	-	-	-	none		
		16	6	9	1	0	-	-	-	-	-	-	?	n	-	-	n	-	-	n	-	n		n	
		32	6	9	2	0	-	n	?	-	-	-	n	n	-	-	n	-	-	n	-	n		n	
ac17	test mix	16	7	6	1	0	-	-	-	y	n	y	n	n	-	y	n	-	-	-	-	-	none		
		32	6	6	1	0	-	-	-	-	-	-	?	n	-	-	n	-	-	n	-	n		n	
		64	6	6	2	0	-	n	?	-	-	-	n	n	-	-	n	-	-	n	-	n		n	
ac18	test mix	16	7	5	1	0	-	-	-	y	n	y	n	n	-	y	n	-	-	-	-	-	none		
		64	6	5	1	0	-	-	-	-	-	-	?	n	-	-	n	-	-	n	-	n		n	
		128	6	5	2	0	-	n	?	-	-	-	n	n	-	-	n	-	-	n	-	n		n	
ac19	test mix	32	7	2	1	0	-	-	-	y	n	y	n	n	-	y	n	-	-	-	-	-	none		
		96	6	2	1	0	-	-	-	-	-	-	?	n	-	-	n	-	-	n	-	n		n	
		192	6	2	2	0	-	n	?	-	-	-	n	n	-	-	n	-	-	n	-	n		n	
ac20	test mix	16	7	1	1	1	-	-	-	y	n	y	n	n	-	n	n	-	-	-	-	-	none		
		64	6	1	2	1	-	n	y	-	-	-	y	n	-	-	n	-	-	n	-	n		n	
		64	6	1	2	1	-	n	y	-	-	-	-	n	-	-	n	-	-	n	-	n		n	
		64	6	1	2	1	-	n	y	-	-	-	-	n	-	-	n	-	-	n	-	n		n	
		64	6	1	2	1	-	n	v	-	-	-	-	-	n	-	-	n	-	-	n	-		n	n

file base name	content	bitrate (kbit/s)	AudioObjectType	SamplingFrequency/Index	ChannelConfiguration	frameLengthFlag	coreCoderDelay	intensity	MS	window sequence switching	non-meaningful window _sequence transitions	window shape switching	tns_data_present	pulse data	prediction	LTP	PNS	data stream elements	gain compensation enabled	dynamic range control	bandwidth	buffer test	arithmetic torture	test procedure
ac22	test mix	32 32 32 32	6 6 6 6	3 3 3 3	2 2 2 2	0 0 0 0	- - - -	y y n n	y y y y	y - - -	n - - -	y - - -	y n n n	n n n n	- - - -	y n n n	n n n n	- - - -	n n n n	- - - -	n n n n	n n n n	none	
NOTE With al22, two different signalling methodologies of channel configurations are available, denoted by _chCfg14 or _chCfg0Pce resp.																								
Key: ‘?’: may be used ‘-’: not applicable ‘n’: no ‘y’: yes																								

Table 27 — AAC test sequences with a framelength 960 samples

file base name	content	lower bitrate (kbit/s)	higher bitrate (kbit/s)	AudioObjectType	SamplingFrequency/Index	ChannelConfiguration	corresponding sequence in terms of properties	frameLengthFlag	test procedure
al00sf	sine sweep	43	69	2	*	0	al00	1	RMS
al01sf	music	43	69	2	*	0	al01	1	none
al02sf	music	43	69	2	*	0	al02	1	none
al03sf	music	43	69	2	*	0	al03	1	none
al04sf	music	43	69	2	*	0	al04	1	none
al05sf	music	86	137	2	*	0	al05	1	none
al06sf	test mix	128	205	2	0..2,7..11	0	al06	1	none
al06sf	test mix	205	205	2	3..6	0	al06	1	none
al07sf	music, other	256	410	2	*	0	al07	1	none
al08sf	music	2048	3277	2	0..2,7..11	0	al08	1	none
al08sf	music	3277	3277	2	3..6	0	al08	1	none
al14sf	music	69	137	2	*	0	al14	1	none
al15sf	music	205	410	2	*	0	al15	1	none
al16sf	music	103	410	2	*	0	al16	1	none
al17sf	music	86	137	2	*	0	al17	1	none
al18sf	noise	43	69	2	*	1	al18	1	PNS-1
al19sf	noise	43	69	2	*	1	al19	1	PNS-2/3
Key: '*': variable									

Table 28 — ER AAC test sequences

file base name	content	bitrate (kbit/s)	AudioObjectType	SamplingFrequencyIndex	ChannelConfiguration	epConfig	frameLengthFlag	coreCoderDelay	aacSectionDataResilienceFlag	aacScalefactorDataResilienceFlag	aacSpectralDataResilienceFlag	intensity	MS	window sequence switching	window shape switching	TNS	pulse data	LTP	PNS	test procedure
er_al10	sine sweep	40/64	17	*	1	0,1	0	-	0	0	0	-	-	y	y	n	n	n	n	RMS
er_al12	test mix	40/64	17	*	1	0,1	0	-	1	0	0	-	-	?	?	?	n	n	fs1	none
er_al15	test mix	40/64	17	*	1	0,1	0	-	0	0	1	-	-	?	?	?	n	n	fs2	none
er_al18	test mix	40/64	17	*	1	0,1	0	-	1	1	1	-	-	?	?	?	n	n	fs1	none
er_al21	test mix	80/128	17	*	2	0,1	0	-	0	0	0	y	y	?	?	?	n	n	fs2	none
er_al23	test mix	80/128	17	*	2	0,1	0	-	0	1	0	y	y	?	?	?	n	n	fs1	none
er_al26	test mix	80/128	17	*	2	0,1	0	-	1	0	1	y	y	?	?	?	n	n	fs2	none
er_ap10	sine sweep	40/64	19	*	1	0,1	0	-	0	0	0	-	-	y	y	n	n	?	n	RMS
er_ap14	test mix	40/64	19	*	1	0,1	0	-	1	1	0	-	-	?	?	?	n	?	fs1	none
er_ap27	test mix	80/128	19	*	2	0,1	0	-	0	1	1	y	y	?	?	?	n	?	fs2	none
er_ad1000	sine sweep	64	23	*	1	0,1	0	-	0	0	0	-	-	-	y	n	n	?	n	RMS
er_ad1109	noise	64	23	*	1	0,1	1	-	0	0	0	-	-	-	n	n	n	n	y	PNS-1
er_ad1102	sine sweep	64	23	*	1	0,1	1	-	0	0	0	-	-	-	y	n	n	?	n	RMS
er_ad1103	test mix	64	23	*	1	0,1	1	-	0	0	0	-	-	-	?	?	n	?	fs2	none
er_ad1111	test mix	64	23	*	1	0,1	1	-	0	0	1	-	-	-	?	?	n	?	fs1	none
er_ad2040	test mix	128	23	*	2	0,1	0	-	1	0	0	y	y	-	?	?	n	?	fs1	none
er_ad2050	test mix	128	23	*	2	0,1	0	-	1	0	1	y	y	-	?	?	n	?	fs2	none
er_ad2160	test mix	128	23	*	2	0,1	1	-	1	1	0	y	y	-	?	?	n	?	fs1	none
er_ad2170	test mix	128	23	*	2	0,1	1	-	1	1	1	y	y	-	?	?	n	?	fs2	none
er_ad1000np	sine sweep	64	23	*	1	0,1	0	-	0	0	0	-	-	-	y	n	n	n	n	RMS
er_ad1020np	test mix	64	23	*	1	0,1	0	-	0	1	0	-	-	-	?	?	n	n	fs1	none
er_ad1030np	test mix	64	23	*	1	0,1	0	-	0	1	1	-	-	-	?	?	n	n	fs2	none
er_ad1102np	sine sweep	64	23	*	1	0,1	1	-	0	0	0	-	-	-	y	n	n	n	n	RMS
er_ad1103np	test mix	64	23	*	1	0,1	1	-	0	0	0	-	-	-	?	?	n	n	fs2	none
er_ad1111np	test mix	64	23	*	1	0,1	1	-	0	0	1	-	-	-	?	?	n	n	fs1	none
er_ad2040np	test mix	128	23	*	2	0,1	0	-	1	0	0	-	-	-	?	?	n	n	fs1	none
er_ad2050np	test mix	128	23	*	2	0,1	0	-	1	0	1	-	-	-	?	?	n	n	fs2	none
er_ad2160np	test mix	128	23	*	2	0,1	1	-	1	1	0	-	-	-	?	?	n	n	fs1	none
er_ad2170np	test mix	128	23	*	2	0,1	1	-	1	1	1	-	-	-	?	?	n	n	fs2	none
er_ad1100np_ep0	sine sweep	64	23	3,4,5,6,7	1	0	1	-	0	0	0	-	-	-	?	?	n	n	n	RMS
er_ad1009np_ep0	noise	64	23	3,4,5,6,7	1	0	0	-	0	0	0	-	-	-	?	?	n	n	y	PNS-1
er_ad1109np_ep0	noise	64	23	3,4,5,6,7	1	0	1	-	0	0	0	-	-	-	?	?	n	n	y	PNS-1
er_ad1110np_ep0	test mix	64	23	3,4,5,6,7	1	0	1	-	0	0	1	-	-	-	?	?	n	n	n	RMS
er_ad1120np_ep0	test mix	64	23	3,4,5,6,7	1	0	1	-	0	1	0	-	-	-	?	?	n	n	n	RMS
er_ad1140np_ep0	test mix	64	23	3,4,5,6,7	1	0	1	-	1	0	0	-	-	-	?	?	n	n	n	RMS
er_ad1130np_ep0	test mix	64	23	3,4,5,6,7	1	0	1	-	0	1	1	-	-	-	?	?	n	n	n	RMS
er_ad1150np_ep0	test mix	64	23	3,4,5,6,7	1	0	1	-	1	0	1	-	-	-	?	?	n	n	n	RMS
er_ad1160np_ep0	test mix	64	23	3,4,5,6,7	1	0	1	-	1	1	0	-	-	-	?	?	n	n	n	RMS
er_ad1170np_ep0	test mix	64	23	3,4,5,6,7	1	0	1	-	1	1	1	-	-	-	?	?	n	n	n	RMS

file base name	content	bitrate (kbit/s)	AudioObjectType	SamplingFrequencyIndex	ChannelConfiguration	epConfig	frameLengthFlag	coreCoderDelay	aacSectionDataResilienceFlag	aacScalefactorDataResilienceFlag	aacSpectralDataResilienceFlag	intensity	MS	window sequence switching	window shape switching	TNS	pulse data	LTP	PNS	test procedure
er_ad3100np_ep0	test mix	192	23	7	3	0	1	-	0	0	0	?	?	-	?	?	-	n	n	RMS
er_ad4000np_ep0	test mix	256	23	6	4	0	0	-	0	0	0	?	?	-	?	?	-	n	n	RMS
er_ad5100np_ep0	test mix	320	23	5	5	0	1	-	0	0	0	?	?	-	?	?	-	n	n	RMS
er_ad6000np_ep0	test mix	384	23	4	6	0	0	-	0	0	0	?	?	-	?	?	-	n	n	RMS
er_ad7100np_ep0	test mix	448	23	3	7	0	1	-	0	0	0	?	?	-	?	?	-	n	n	RMS
er_ad2101np_ep0	test mix	128	23	0,1,2,3, 4,5,6,7, 8,9,10, 11,12	2	0	1	-	0	0	0	y	n	-	n	?	-	n	n	RMS
er_eld1000np_ep0	sine sweep	64	39	3,4,5,6,7	1	0	0	-	0	0	0	-	-	-	-	n	-	-	n	RMS
er_eld1100np_ep0	sine sweep	64	39	3,4,5,6,7	1	0	1	-	0	0	0	-	-	-	-	n	-	-	n	RMS
er_eld1101np_ep0	test mix	64	39	3,4,5,6,7	1	0	1	-	0	0	0	-	-	-	-	y	-	-	n	none
er_eld1001np_ep0	test mix	64	39	3,4,5,6,7	1	0	0	-	0	0	0	-	-	-	-	y	-	-	n	none
er_eld1102np_ep0	noise	64	39	3,4,5,6,7	1	0	1	-	0	0	0	-	-	-	-	n	-	-	y	PNS-1
er_eld1002np_ep0	noise	64	39	3,4,5,6,7	1	0	0	-	0	0	0	-	-	-	-	n	-	-	y	PNS-1
er_eld2100np_ep0	test mix	128	39	3,4,5,6,7	2	0	1	-	0	0	0	n	y	-	-	n	-	-	n	RMS k=TBD
er_eld2000np_ep0	test mix	128	39	3,4,5,6,7	2	0	0	-	0	0	0	n	y	-	-	n	-	-	n	RMS k=TBD
er_eld1110np_ep0	test mix	64	39	3,4,5,6,7	1	0	1	-	0	0	1	-	-	-	-	n	-	-	n	RMS k=TBD
er_eld1120np_ep0	test mix	64	39	3,4,5,6,7	1	0	1	-	0	1	0	-	-	-	-	n	-	-	n	RMS k=TBD
er_eld1140np_ep0	test mix	64	39	3,4,5,6,7	1	0	1	-	1	0	0	-	-	-	-	n	-	-	n	RMS k=TBD
er_eld1130np_ep0	test mix	64	39	3,4,5,6,7	1	0	1	-	0	1	1	-	-	-	-	n	-	-	n	RMS k=TBD
er_eld2150np_ep0	test mix	128	39	3,4,5,6,7	2	0	1	-	1	0	1	n	y	-	-	n	-	-	n	RMS k=TBD
er_eld1160np_ep0	test mix	64	39	3,4,5,6,7	1	0	1	-	1	1	0	-	-	-	-	n	-	-	n	RMS k=TBD
er_eld1170np_ep0	test mix	64	39	3,4,5,6,7	1	0	1	-	1	1	1	-	-	-	-	n	-	-	n	RMS k=TBD
er_eld1010np_ep0	test mix	64	39	3,4,5,6,7	1	0	0	-	0	0	1	-	-	-	-	n	-	-	n	RMS
er_eld1020np_ep0	test mix	64	39	3,4,5,6,7	1	0	0	-	0	1	0	-	-	-	-	n	-	-	n	RMS
er_eld1040np_ep0	test mix	64	39	3,4,5,6,7	1	0	0	-	1	0	0	-	-	-	-	n	-	-	n	RMS
er_eld1030np_ep0	test mix	64	39	3,4,5,6,7	1	0	0	-	0	1	1	-	-	-	-	n	-	-	n	RMS
er_eld1050np_ep0	test mix	64	39	3,4,5,6,7	1	0	0	-	1	0	1	-	-	-	-	n	-	-	n	RMS
er_eld1060np_ep0	Test mix	64	39	3,4,5,6,7	1	0	0	-	1	1	0	-	-	-	-	n	-	-	n	RMS
er_eld2070np_ep0	Test mix	128	39	3,4,5,6,7	2	0	0	-	1	1	1	n	y	-	-	n	-	-	n	RMS

ISO/IEC 14496-26:2024(en)

file base name	content	bitrate (kbit/s)	AudioObjectType	SamplingFrequencyIndex	ChannelConfiguration	epConfig	frameLengthFlag	coreCoderDelay	aacSectionDataResilienceFlag	aacScalefactorDataResilienceFlag	aacSpectralDataResilienceFlag	intensity	MS	window sequence switching	window shape switching	TNS	pulse data	LTP	PNS	test procedure
er_eld2101np_ep0	test mix	128	39	0,1,2,3, 4,5,6,7, 8,9,10, 11,12	2	0	1	-	0	0	0	y	n	-	-	?	-	-	n	RMS
er_eld3000np_ep0	test mix	192	39	7	3	0	0	-	0	0	0	?	?	-	-	?	-	-	n	RMS
er_eld4100np_ep0	test mix	256	39	6	4	0	1	-	0	0	0	?	?	-	-	?	-	-	n	RMS
er_eld5000np_ep0	test mix	320	39	5	5	0	0	-	0	0	0	?	?	-	-	?	-	-	n	RMS
er_eld6100np_ep0	test mix	384	39	4	6	0	1	-	0	0	0	?	?	-	-	?	-	-	n	RMS
er_eld7000np_ep0	test mix	448	39	3	7	0	0	-	0	0	0	?	?	-	-	?	-	-	n	RMS
er_ac111	test mix	64	20	3	1	0,1	0	-	0	0	0	-	-	?	?	?	n	n	n	none
		64	20	3	2		0	-	0	0	1	n	y	-	-	?	n	-	n	
er_ac118	sine sweep	40	20	2	1	0,1	0	-	0	1	0	-	-	y	y	n	n	n	n	RMS
		24	20	2	1		0	-	0	1	1	-	-	-	-	-	n	-	n	
		40	20	2	2		0	-	1	0	0	n	y	-	-	n	n	-	n	
		40	20	2	2		0	-	1	0	1	n	y	-	-	-	n	-	n	
er_ac119	sine sweep	30	20	7	1	0,1	0	-	0	1	0	-	-	y	y	n	n	n	n	RMS
		16	20	7	1		0	-	0	1	1	-	-	-	-	-	n	-	n	
		96	20	7	2		0	-	1	0	0	n	y	-	-	n	n	-	n	
		20	20	7	2		0	-	1	0	1	n	y	-	-	-	n	-	n	
er_ac121	music	32	20	3	2	0	0	-	0	0	0	y	y	n	n	n	n	-	n	none
		32	20	3	2		0	-	0	0	0	y	y	-	-	-	n	-	n	
		32	20	3	2		0	-	0	0	0	n	y	-	-	-	n	-	n	
		32	20	3	2		0	-	0	0	0	n	y	-	-	-	n	-	n	
er_ac123	test mix	40	20	4	2	0,1	0	-	1	1	0	n	y	?	?	?	n	n	y	none
		64	20	4	2		0	-	1	1	1	n	y	-	-	-	n	-	n	
er_ac211	test mix	6	24	11	1	0,1	-	-	-	-	-	-	-	-	-	-	-	-	-	none
		40	20	6	1		1	0	0	1	0	-	-	?	?	?	n	-	y	
		64	20	6	2		1	-	0	0	1	n	y	-	-	?	n	-	n	
er_ac221	test mix	6.2	24	12	1	0,1	-	-	-	-	-	-	-	-	-	-	-	-	-	none
		64	20	7	2		1	0	0	0	0	n	y	?	?	?	n	-	n	
		128	20	7	2		1	-	0	1	1	n	y	-	-	-	n	-	n	
er_ac311	test mix	8	21	5	1	0,1	0	-	-	-	-	-	-	?	?	-	-	n	-	none
		40	20	5	1		0	-	1	0	0	-	-	-	-	?	n	-	n	
		64	20	5	2		0	-	1	1	1	n	y	-	-	?	n	-	n	
er_ac321	test mix	12	21	6	1	0,1	1	-	-	-	-	-	-	?	?	-	-	n	-	none
		64	20	6	2		1	-	1	1	0	n	y	-	-	?	n	-	y	
		96	20	6	2		1	-	1	0	1	n	y	-	-	-	n	-	n	

Key:
'?: may be used
': not applicable
'fs1': yes if fs is one of the following: 08, 12, 22, 32, 48, 88; no otherwise
'fs2': yes if fs is one of the following: 11, 16, 24, 44, 64, 96; no otherwise
'n' - no
'y' - yes

Table 29 — Specification of the program_config_element()

file name	num_front_channel_elements	front_element_is_cpe	num_side_channel_elements	side_element_is_cpe	num_back_channel_elements	back_element_is_cpe	num_lfe_channel_elements	num_valid_cc_elements	cc_element_is_ind_sw
am00	1	n	0	-	0	-	0	0	-
am01	1	n	0	-	0	-	0	0	-
am02	1	y	0	-	0	-	0	0	-
am04	1	y	0	-	0	-	0	0	-
am05	2	n y	0	-	1	y	1	1	y
am06	2	n y	0	-	0	-	0	0	-
am07	2	n y	0	-	1	y	1	0	-
al00	1	n	0	-	0	-	0	0	-
al01	1	n	0	-	0	-	0	0	-
al02	1	n	0	-	0	-	0	0	-
al03	1	n	0	-	0	-	0	0	-
al04	1	n	0	-	0	-	0	0	-
al05	1	y	0	-	0	-	0	0	-
al06	2	n y	0	-	0	-	0	0	-
al07	2	n y	0	-	1	y	1	1	n
al08	11	n n n n n n y y y y	9	y y y y y y y y	12	y y y n n n n n n n	0	0	-
al14	1	y	0	-	0	-	0	0	-
al16	2	n y	0	-	0	-	0	0	-
al17	2	n n	0	-	0	-	0	0	-
as00	1	n	0	-	0	-	0	0	-
as01	1	n	0	-	0	-	0	0	-
as02	1	n	0	-	0	-	0	0	-
as03	1	n	0	-	0	-	0	0	-
as04	1	n	0	-	0	-	0	0	-
as05	1	n	0	-	0	-	0	0	-
as06	1	n	0	-	0	-	0	0	-
as07	1	n	0	-	0	-	0	0	-
as08	1	n	0	-	0	-	0	0	-
as09	1	y	0	-	0	-	0	0	-
as10	1	y	0	-	0	-	0	0	-
as11	1	y	0	-	0	-	0	0	-
as12	1	y	0	-	0	-	0	0	-
as13	2	n y	1	n	0	-	1	0	-
as14	2	n y	1	n	0	-	1	0	-
as15	3	n y y	0	-	1	n	1	0	-
as16	3	n y y	0	-	1	n	1	0	-
as17	1	n	0	-	0	-	0	0	-
ap01	1	n	0	-	0	-	0	0	-
ap02	2	n	0	-	0	-	0	0	-

file name	num_front_channel_elements	front_element_is_cpe	num_side_channel_elements	side_element_is_cpe	num_back_channel_elements	back_element_is_cpe	num_lfe_channel_elements	num_valid_cc_elements	cc_element_is_ind_sw
ap03	1	n	0	-	0	-	0	0	-
ap04	1	n	0	-	0	-	0	0	-
ap05	2	n	0	-	0	-	0	0	-
Key: '*': variable '?': may be used '-': not applicable 'n': no 'y': yes									

7.5 TwinVQ and ER_TwinVQ

7.5.1 DecoderSpecificInfo characteristics

Encoders may apply restrictions to the following parameters of the Object Descriptor Stream:

- samplingFrequencyIndex (descriptor element which indicates sampling rate).
- bitrate (indicates bitrate).
- number of layers (indicates the number of scalable layers).
- number of channels (indicates the number of channels of input signal).
- frameLength (indicate frame length is 1024 or 960).

7.5.2 Audio access unit characteristics

Encoders may apply restrictions to the following parameters of the bitstream:

- window_sequence
- window_shape
- LTP (ltp_present)
- M/S stereo (msmask_present)
- TNS (tns_present)
- quantizer option (bandlimit, ppc, postprocess)

7.5.3 Procedure to test bitstream conformance

7.5.3.1 Parsing system layer parameters

The decoder shall get the information of the sampling frequency, number of layers, bitrate and number of channels from the system layer.

7.5.3.2 Decoding of the payload

7.5.3.2.1 parsing tvq_scalable_main_header()

The syntax has the **window_sequence**, **window_shape**, **ms_mask_present**, **scale_factor_grouping**, **ltp_data_present**, and **tns_data_resent**. These syntax elements are common to those for AAC.

7.5.3.2.2 parsing tvq_scalable_extension_header()

The syntax has **ms_mask_present** common to AAC.

7.5.3.2.3 parsing vq_single_element()

The syntax has the flags for the quantizer option of **band_limit**, **ppc**, **postprocess**, as well as the main quantization information. Detailed specification is described in ISO/IEC 14496-3 subpart 4.

7.5.4 Decoder characteristics

A conformant decoder shall support all characteristics given by the definition of level in the scaleable profile.

7.5.5 Procedure to test decoder conformance

For the purpose of testing the processing at the decoder, number of bitstreams and the associated reference output PCM signals are supplied as listed in Table 30, Table 31, Table 32 and Table 33. They cover the wide range of sampling rate, bit rate, number of channels, number of scalable layers, AAC related tools (window_shape, LTP, M/S stereo, TNS), and TwinVQ specific quantizer options (bandlimit, ppc, postprocess).

TV20 and TV24 contain the code to scan all codebook tables of the vector quantizers.

The ms_mode 1 means that the **ms_mask_present** == 1 or 0, ms_mode_present 2 means that **ms_mask_present** == 2 or 0.

The actual bit rate of the bitstreams may be slightly less than the values listed due to the byte alignment process.

Two-step accuracy criteria for conformance

A two-step approach is used to distinguish between two levels of accuracy, namely Fixed-Point accuracy and full accuracy of accuracy for decoder conformance.

Full Accuracy: A decoder meeting the stronger Full Accuracy conformance requirements may be called a Full Accuracy conformant decoder. This level of accuracy is intended for decoders running on floating-point platforms, enabling higher-precision mathematical operations.

Fixed-Point Accuracy: A decoder may be called conformant with Fixed-Point Accuracy in case the Fixed-Point Accuracy conformance criteria are met. Decoders with a limited accuracy due to fixed-point internal calculations may use these conformance criteria to verify the validity of the decoder.

Conformance criterion for Full Accuracy TwinVQ and ER_TwinVQ decoders

The RMS/LSB Measurement test procedure applies (see 7.1.2.2.1)

Conformance criteria for Fixed-Point Accuracy TwinVQ and ER_TwinVQ decoders

The conformance criteria for Fixed-Point Accuracy decoders are based on measuring the segmental SNR and the LPC cepstral distortion (CD) between the Reference decoder output and the output of the decoder to be tested. The segment length to be used in the calculation of the SNR is equal to the general audio frame length, namely 1024 or 960. The SNR and the CD have to be calculated only for the segments of which the power of the Reference signal is in the range [-50...-15] dB. CD is defined as

$$CD = \frac{10}{\ln(10)} \cdot \sqrt{2D}$$

D is the accumulated distortion of the LPC cepstrum C_{ref} of the reference signal and C_{test} of the output of the decoder under test. D is defined as

$$D = \sum_{i=1}^N (C_{ref}[i] - C_{test}[i])^2$$

N is the LPC cepstrum order which equals 32. The LPC cepstrum $C[i]$ is defined by means of the algorithm `lpc2cepstrum` based on the LPC coefficients of a 16th order linear prediction filter. The computation of the LPC filter coefficients `lpc_coef[j]` is defined by the algorithm `calculate_lpc`.

To be called an ISO/IEC 14496-3 TwinVQ object type decoder with Fixed-Point Accuracy, the average value of the segmental SNR shall exceed 30 dB and at the same time the average value of the CD shall not exceed 1 dB.

7.5.6 Descriptions of the audio test bitstreams

Table 30 — TwinVQ object type test bitstreams

File Name	TV00	TV01	TV02	TV03	TV04	TV05	TV06	TV07
content	music	music	music	music	music	music	music	music
level in scalable profile	1	1	1	1	1	1	1	1
bitrate [kbit/s]	8	16	16	16	16	16	16	16
sampling rate [kHz]	8	16	16	16	16	16	16	16
frame length	1024	1024	1024	1024	960	1024	1024	1024
number of scaleable layers	1	1	1	1	1	1	1	1
number of channels	1	1	1	1	1	1	1	1
long-term prediction (LTP)	no	no	yes	no	no	no	no	no
adaptive window shape	no	yes	no	no	no	no	no	no
TNS	no	no	no	yes	no	no	no	no
M/S stereo mode 2	-	-	-	-	-	-	-	-
M/S stereo mode 1	-	-	-	-	-	-	-	-
bandlimit_present	no	no	no	no	no	yes	no	no
ppc_present	no	no	no	no	no	no	yes	no
postprocess_present	no	no	no	no	no	no	no	yes

Table 31 — TwinVQ object type test bitstreams (continued)

File Name	TV11	TV14	TV15	TV16	TV20	TV21
content	music	music	music	music	music	music
level in scalable profile	2	1	3	2	1	2
bitrate [kbit/s]	16 + 16	8+8+8	32+32+32	12*8	8	16
sampling rate [kHz]	16	24	48	44.1	16	44.1
frame length	1024	1024	1024	1024	1024	1024
number of scaleable layers	1	3	3	8	1	1
number of channels	2	1	2	1	1	1
long-term prediction (LTP)	no	no	no	no	no	no
adaptive window shape	no	no	no	no	no	yes
TNS	no	no	no	no	no	no
M/S stereo mode 2	yes	-	yes	-	-	-
M/S stereo mode 1	yes	-	no	-	-	-
bandlimit_present	no	no	no	no	no	yes
ppc_present	no	no	no	no	no	yes
postprocess_present	no	no	no	no	no	yes
					yes	no