INTERNATIONAL STANDARD

ISO/IEC 23008-9

Second edition
2022-03

Information technology — High efficiency coding and media delivery in heterogeneous environments —

Part 9: 3D Audio conformance testing

Technologies de l'information — Codage à haut rendement et fourniture de supports dans les environnements hétérogènes —

Partie 9: Essais de conformité 3D Audio

Reference number
ISO/IEC 23008-9:2022(E)

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

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).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

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 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 23008-9:2019), which has been technically revised.

The main changes are as follows:

— conformance testing of Baseline Profile support.

A list of all parts in the ISO 23008 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

This document specifies how tests can be designed to verify whether bitstreams and decoders meet the requirements as specified in ISO/IEC 23008-3 and allow interoperability with remote terminals in interactive, broadcast, streaming and local (with stored contents) sessions. These tests 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 23008-3,

— manufacturers of decoders and their customers can use the tests to verify whether the decoder meets the requirements specified in ISO/IEC 23008-3 for the claimed decoder capabilities,

— manufacturers and customers of terminals supporting interactive, broadcast, streaming, 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 23008-3,

— manufacturers of test equipment, and their customers can use the tests to verify compliance with ISO/IEC 23008-3.

Information technology — High efficiency coding and media delivery in heterogeneous environments —

Part 9:
3D Audio conformance testing

1 Scope

This document specifies conformance criteria for both bitstreams and decoders compliant with the MPEG-H 3D audio standard as defined in ISO/IEC 23008-3. This is done to assist implementers and to ensure interoperability.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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

ISO/IEC 23003-2, *Information technology — MPEG audio technologies — Part 2: Spatial Audio Object Coding (SAOC) — Amendment 4: SAOC Conformance*

ISO/IEC 23003-3:2020, *Information technology — MPEG audio technologies — Part 3: Unified speech and audio coding*

ISO/IEC 23003-4:2020, *Information technology — MPEG audio technologies — Part 4: Dynamic range control*

ISO/IEC 23008-3:2019/Amd 2:2020, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 3: 3D audio/ — Amendment 2: 3D Audio baseline profile, corrections and improvements*

ISO/IEC 23091-3, *Information technology — Coding-independent code points — Part 3: Audio*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions 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/

3.1.1
bitstream
encoded audio data

3.1.2
conformance test bitstream
MPEG-H 3DA encoded bitstream used for testing the conformance of a MPEG-H 3DA decoder

3.1.3
conformance test case
combination of one or more conformance test conditions for which one conformance test bitstream is provided

3.1.4
conformance test condition
condition which applies to properties of a conformance test bitstream in order to test a certain functionality of the MPEG-H 3DA decoder

3.1.5
conformance test criteria
one or more conformance test tools with requirements that define whether a given output from a decoder under test fulfils the conformance

3.1.6
conformance test data
conformance test sequences and conformance criteria

3.1.7
conformance test sequences
generic term for conformance test bitstream and decoder settings with a corresponding reference

3.1.8
conformance test tool
tool to compare the reference waveform with the output from a decoder under test

3.1.9
decoder setting case
combination of one or more decoder setting conditions to trigger specific settings of the decoder

3.1.10
decoder setting condition
condition applied to the decoder behaviour in order to test functionality of the MPEG-H 3DA decoder

3.1.11
reference waveform
decoded counterpart of a conformance test bitstream with specific decoder settings

3.2 Abbreviated terms

3DA	3D audio
MPEG-H 3DA bitstream	data encoded according to ISO/IEC 23008-3
MPEG-H 3DA CPE	mpegh3daChannelPairElement
MPEG-H 3DA EXT	mpegh3daExtElement
MPEG-H 3DA LFE	mpegh3daLfeElement
MPEG-H 3DA SCE	mpegh3daSingleChannelElement

4 MPEG-H 3D audio conformance testing

4.1 General

This clause specifies conformance criteria for both bitstreams and decoders compliant with ISO/IEC 23008-3 as defined in this document. This is done to assist implementers and to ensure interoperability.

4.2 Profiles

Profiles are defined in ISO/IEC 23008-3:2019, 4.8. Some conformance criteria apply to MPEG-H 3D audio in general, while others are specific to certain profiles and their respective levels. Conformance shall be tested for the level of the profile with which a given bitstream or decoder claims to comply.

In addition to the conformance requirements described in this clause, a decoder which claims to comply with the MPEG-H 3D audio shall fulfil conformance defined in ISO/IEC 23003-4:2020, Clause 9.

4.3 Test procedure

4.3.1 General

To test a decoder for compliance to MPEG-H 3D audio decoding, conformance test data is provided. The package of the conformance test data is described in Figure 1. It is accessible at https://standards.iso.org/iso-iec/23008/-9/ed-2/en and contains all conformance test tools, conformance bitstreams, reference waveforms, and conformance tables in a spreadsheet. The latter defines all the conformance test sequences. To fulfil a conformance test sequence, the decoder under test shall decode the corresponding conformance test bitstream with the given decoder setting case. The output of the decoder under test shall meet the conformance test criteria in comparison with the respective reference waveform. Some conformance test sequences test only the stability of the decoder under test. Such conformance test sequences consist only of the conformance test case with the corresponding conformance test bitstream. The decoder under test shall pass all conformance test sequences.

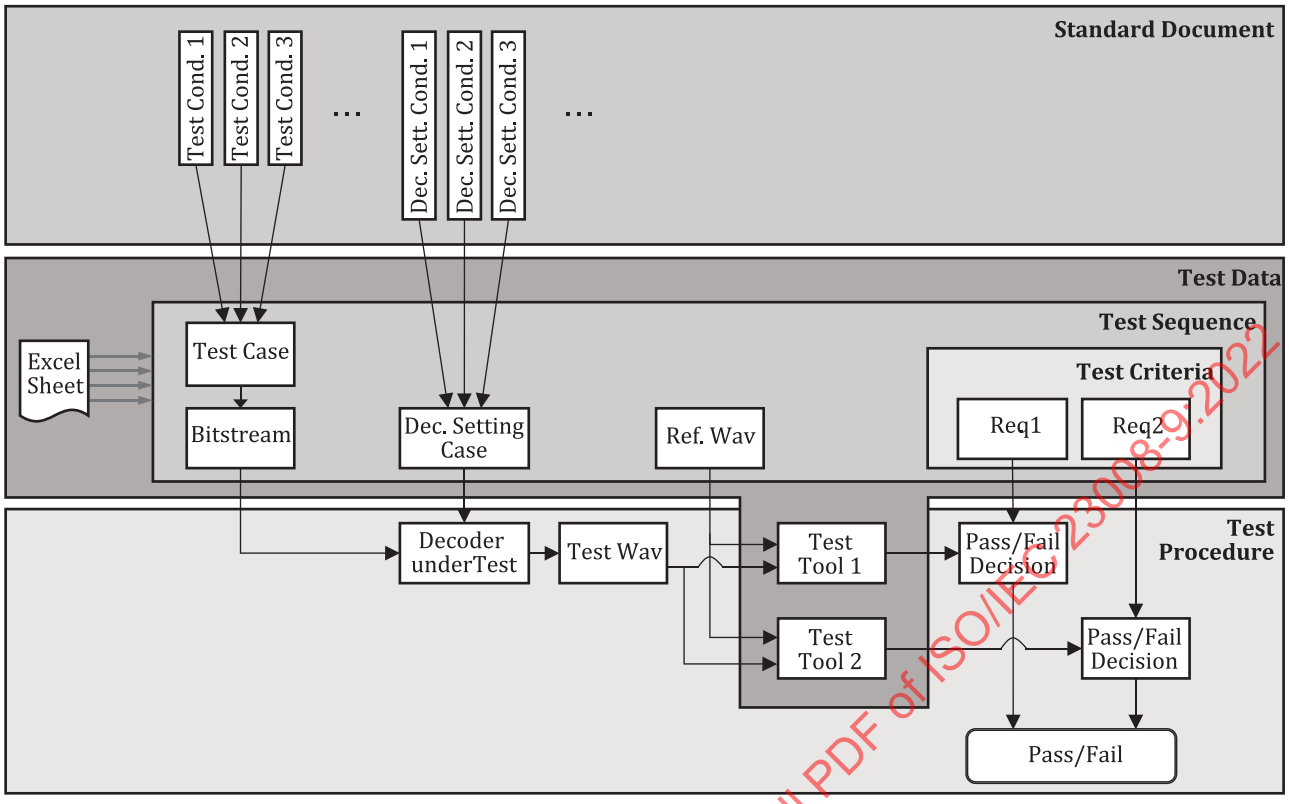

Figure 1 — Conformance testing components

In case 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 decoded reference waveforms are supplied as ".wav" files. These are always supplied as 24 bit resolution (RIFF (little-endian) data, WAVE audio, pulse code modulation (PCM), 24 bit).

The conformance test criteria define what conformance test tools and parameter are used to compare the reference waveforms with the output waveforms from the decoder under test. The conformance test tools are defined in 4.3.3.

The characteristics of the conformance test bitstream are defined by the corresponding conformance test case. A conformance test case is a combination of all basic conformance test conditions, specified in 7.2, and possibly one or several additional conformance test conditions, specified in 7.3. Note that the same conformance test bitstream may be a part of several conformance test sequences.

The decoder setting case is a combination of the decoder setting conditions. The decoder setting conditions are defined in 7.3.9. If no specific decoder setting case is applicable, then all the decoder settings for the decoder default behaviour shall be used.

4.3.2 Naming convention

The name of the conformance test case is composed of several parts. The first part conveys the information of all basic conformance test conditions. These abbreviations are appended in the order as defined in 7.2. The second part of the name conveys the information of all additional test conditions. The abbreviations of the additional test conditions are defined in Table 1. The abbreviations of the conformance test conditions are in an alphabetical order. All conformance test conditions are connected by an underscore.

The name of the conformance test case is illustrated as follows:

[3daC_ElemConfIdx]_[3daC_SampFreqIdx]_[FD|LPD|Cct]_<testCase1>_<testCase2>

The name for the corresponding conformance test bitstream is the same appended with the transport format extension.

The name of the decoder setting case is composed of several decoder setting conditions. The abbreviations of the decoder setting conditions are defined in Table 2. The abbreviations of the decoder setting conditions are in the order as defined in 7.3.9.

The name of the conformance test sequence is composed of the name of the conformance test case and the name of the decoder setting case connected by two underscores.

The name of the conformance test sequence is illustrated as follows:

[3daC_ElemConfIdx]_[3daC_SampFreqIdx]_[FD|LPD|Cct]_<testCase1>_<testCase2>__<decoderSettingCase1>_<decoderSettingCase2>

The name of the corresponding reference waveform is the same as the name of the conformance test sequence appended with the audio file extension.

Table 1 — Conformance test abbreviations for additional test conditions

Test condition	Abbreviation
Basic FD window	Win
Non-meaningful FD window switching	Nmf
Aliasing symmetries	Asy
Noise filling	Nf
Varying max_sfb	Sfb
TNS test condition	Tns
M/S stereo	Ms
Complex prediction stereo	Cp
LPC coding	Lpc
ACELP core mode	Ace
TCX and noise filling	Tcx
fullband LPD	fbL-<x1>-<x2>...
LPD mode coverage and FAC	Lpd
AVQ test condition	avq
stereo LPD	sLP-<x1>-<x2>...
Time domain bandwidth extension	Tbe
Frequency domain prediction	Fdp
Long-term postfilter	Lpf
Bass-post filter	Bpf
Channel pair element configuration	cpc-<x1>-<x2>-...
IGF range signalling	E-ran-<x>-<y>
IGF tiling	E-Cti
IGF whitening	E-Wht
IGF envelope noise flattening	E-Enf
IGF after TNS synth	E-Ats
IGF no high resolution	E-Nhr
IGF no independent tiling	E-Nit

Table 1 *(continued)*

Test condition	Abbreviation
Stereo filling	E-SFi
MCT channel signalling	M-chM-<x>
MCT signalling type	M-Typ-<x>
mct stereo filling	M-SFi-<x>
MCT mechanics	M-Mec
MCT rotation content	M-Rot
MCT prediction content	M-Pre
OAM position and gain	O-Pos
OAM transmission rate	O-rat-<x>
OAM spread modes	O-spr-<x>
Loudness normalization	D-Ln-Lay-<x0-x1-...>-Gr-<y0-y1-...>-Pr-<z0-z1-...>
Dynamic range control	D-Drc-<w0-w1-...>-Lay-<x0-x1-...>-Gr-<y0-y1-...>-Pr-<z0-z1-...>
Ducking	D-Duck-Gr-<x0-x1-...>-Pr-<y0-y1-...>
AudioPreRoll, IPF freq. of occurrence	I-foo-<x>

Table 2 — Conformance test abbreviations for decoder setting condition

Test condition	Abbreviation
Target layout	Lay-<x>
Target loudness	Lou-<x>
DRC effect type request	Eff-<x>
Group preset selection	Pr-<x>
Conformance point	Cpo-<x>

4.3.3 Conformance test tools

4.3.3.1 RMS/LSB measurement

The RMS/LSB measurement is defined in ISO/IEC 14496-26:2010, 7.1.2.2.1.

4.3.3.2 Segmental SNR

The Segmental SNR is defined in ISO/IEC 14496-26:2010, 7.1.2.2.2.

5 MPEG-H 3D audio bitstreams

5.1 Characteristics, test procedure

Characteristics of bitstreams specify the constraints that are applied by the encoder in generating the bitstream. These syntactic and semantic constraints may, for example, restrict the range or the values of parameters that are encoded directly or indirectly in the bitstreams.

Each MPEG-H 3DA bitstream shall meet the syntactic and semantic requirements specified in this document. The present clause defines the conformance criteria that shall be fulfilled by a compliant bitstream. These criteria are specified for the syntactic elements of the bitstream and for some parameters decoded from the MPEG-H 3DA bitstream payload.

For each tool a set of semantic tests to be performed on the bitstreams 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 bitstreams contain 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.

5.2 MPEG-H 3D audio general configuration

5.2.1 mpegh3daConfig()

mpegh3daProfileLevelIndication	shall be one of the non-reserved values as defined in ISO/IEC 23008-3:2019, Table 64
usacSamplingFrequencyIndex	shall be one of the non-reserved values as defined in ISO/IEC 23003-3:2020, Table 67
usacSamplingFrequency	no restrictions apply
coreSbrFrameLengthIndex	shall be one of the non-reserved values as defined in ISO/IEC 23003-3:2020, Table 70
reserved	shall be 0
receiverDelayCompensation	no restrictions apply
usacConfigExtensionPresent	no restrictions apply

5.2.2 FrameworkConfig3d()

No restrictions are applicable to this bitstream element.

5.2.3 Signals3d()

bsNumSignalGroups	shall be 0 if speakerLayoutType of the referenceLayout is 3
signalGroupType	shall be 0 if speakerLayoutType of the referenceLayout is 3
differsFromReferenceLayout	shall be 0 if speakerLayoutType of the referenceLayout is 3
saocDmxLayoutPresent	no restrictions apply

5.2.4 SpeakerConfig3d()

speakerLayoutType	no restrictions apply
CICPspeakerLayoutIdx	shall be encoded in accordance with ChannelConfiguration as defined in ISO/IEC 23091-3
numSpeakers	no restrictions apply
CICPspeakerIdx	shall be encoded in accordance with LoudspeakerGeometry as defined in ISO/IEC 23091-3

5.2.5 mpegh3daFlexibleSpeakerConfig()

angularPrecision	no restrictions apply

alsoAddSymmetricPair	no restrictions apply

5.2.6 mpegh3daSpeakerDescription()

isCICPspeakerIdx	no restrictions apply
CICPspeakerIdx	shall be encoded in accordance with Loudspeaker-Geometry as defined in ISO/IEC 23091-3
ElevationClass	no restrictions apply
ElevationAngleIdx	if angularPrecision == 0, shall have a value between and including 0 and 18; if angularPrecision == 1, shall have a value between and including 0 and 90
ElevationDirection	no restrictions apply
AzimuthAngleIdx	if angularPrecision == 0, shall have a value between and including 0 and 36; if angularPrecision == 1, shall have a value between and including 0 and 180
AzimuthDirection	no restrictions apply
isLFE	no restrictions apply

5.3 MPEG-H 3D core audio configuration

5.3.1 mpegh3daDecoderConfig()

numElements	no restrictions apply
elementLengthPresent	no restrictions apply
usacElementType	no restrictions apply

5.3.2 mpegh3daSingleChannelElementConfig()

No restrictions are applicable to this bitstream element.

5.3.3 mpegh3daChannelPairElementConfig()

igfIndependentTiling	no restrictions apply
stereoConfigIndex	no restrictions apply
qceIndex	no restrictions apply
shiftIndex0	no restrictions apply
shiftChannel0	a shifted channel shall not exceed the maximum channel index of the signal group it is associated with
shiftIndex1	no restrictions apply

shiftChannel1 — a shifted channel shall not exceed the maximum channel index of the signal group it is associated with

lpdStereoIndex — no restrictions apply

5.3.4 mpegh3daCoreConfig()

tw_mdct — no restrictions apply

fullbandLpd — no restrictions apply

noiseFilling — no restrictions apply

enhancedNoiseFilling — no restrictions apply

igfUseEnf — no restrictions apply

igfUseHighRes — no restrictions apply

igfUseWhitening — no restrictions apply

igfAfterTnsSynth — no restrictions apply

igfStartIndex — no restrictions apply

igfStopIndex — no restrictions apply

5.3.5 mpegh3daLfeElementConfig()

No restrictions are applicable to this bitstream element.

5.3.6 mpegh3daExtElementConfig()

Depending on the **usacExtElementType** the following restrictions apply as defined in Table 3.

Table 3 — Type dependent restrictions for mpegh3daExtElementConfig()

usacExt ElementType (ID_EXT_ELE_...)	elemIdx of mpegh3da Decoder Config()	Number of occurrences	usacExt Element Config Length	usacExt Element Default Length Present	usacExt Element Default Length	usacExt Element Payload Frag
FILL	NR	NR	== 0	NR	NR	== 0
MPEGS	NR	NR	NR	NR	NR	NR
SAOC	NR	NR	NR	NR	NR	NR
AUDIOPREROLL	== 0	≤ 1	== 0	== 0	N/A	== 0
UNI_DRC	NR	≤ 1	NR	NR	NR	== 0
OBJ_METADATA	NR	NR	NR	NR	NR	NR
SAOC_3D	NR	NR	NR	NR	NR	NR
HOA	NR	NR	NR	NR	NR	NR
FMT_CNVRTR	NR	NR	== 0	NR	NR	NR
MCT	NR	NR	NR	NR	NR	== 0
Key NR: no restrictions apply N/A: not applicable						

Table 3 *(continued)*

usacExt ElementType (ID_EXT_ELE_...)	elemIdx of mpegh3da Decoder Config()	Number of occurrences	usacExt Element Config Length	usacExt Element Default Length Present	usacExt Element Default Length	usacExt Element Payload Frag
TCC	NR	NR	NR	NR	NR	NR
HOA_ENH_LAYER	NR	NR	NR	NR	NR	NR
HREP	NR	NR	NR	NR	NR	== 0
ENHANCED_OBJ_METADATA	NR	NR	NR	NR	NR	NR
Key NR: no restrictions apply N/A: not applicable						

tmp no restrictions apply

5.3.7 mpegh3daConfigExtension()

numConfigExtensions no restrictions apply

usacConfigExtType no restrictions apply

usacConfigExtLength no restrictions apply

fill_byte should be '10100101'

tmp no restrictions apply

5.3.8 SbrConfig()

The SbrConfig() bitstream structure and all recursively included bitstream structures shall be restricted as specified in ISO/IEC 23003-3:2020.

5.3.9 Mps212Config()

The Mps212Config() bitstream structure and all recursively included bitstream structures shall be restricted as specified in ISO/IEC 23003-3:2020.

5.4 MPEG-H 3D core audio frame

5.4.1 mpegh3daFrame()

usacIndependencyFlag no restrictions apply

elementLength no restrictions apply

5.4.2 mpegh3daSingleChannelElement()

No restrictions are applicable to this bitstream element.

5.4.3 mpegh3daChannelPairElement()

No restrictions are applicable to this bitstream element.

5.4.4 mpegh3daLfeElement()

No restrictions are applicable to this bitstream element.

5.4.5 mpegh3daExtElement()

Depending on the **usacExtElementType** the restrictions in Table 4 apply.

Table 4 — Type dependent restrictions for mpegh3daExtElement()

usacExtElementType (ID_EXT_ELE_...)	usacExtElement Present	usacExtElementUse Default-Length	usacExtElement Pay-loadLength
FILL	NR	NR	NR
MPEGS	NR	NR	NR
SAOC	NR	NR	NR
AUDIOPREROLL	NR	== 0	NR
UNI_DRC	NR	NR	NR
OBJ_METADATA	NR	NR	NR
SAOC_3D	NR	NR	NR
HOA	NR	NR	NR
FMT_CNVRTR	NR	NR	NR
MCT	== 1	NR	> 0
TCC	NR	NR	NR
HOA_ENH_LAYER	NR	NR	NR
HREP	NR	NR	NR
ENHANCED_OBJ_ METADATA	NR	NR	NR
Key NR: no restrictions apply N/A: not applicable			

usacExtElementStart no restrictions apply

usacExtElementStop no restrictions apply

5.4.6 ics_info()

window_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. The meaningful window_sequence transitions are shown in ISO/IEC 23003-3:2020, 7.9 (Table 138).

window_shape no restrictions apply

max_sfb	shall be <= num_swb_long_window or num_swb_short_window as appropriate for window_sequence and sampling frequency and core coder frame length
scale_factor_grouping	no restrictions apply

5.4.7 mpegh3daCoreCoderData()

core_mode	no restrictions apply
tns_data_present	no restrictions apply

5.4.8 StereoCoreToolInfo()

common_ltpf	no restrictions apply
ltpf_data_present	no restrictions apply
ltpf_pitch_lag_index	no restrictions apply
ltpf_gain_index	no restrictions apply
tns_active	no restrictions apply
common_window	no restrictions apply
common_max_sfb	no restrictions apply
max_sfb1	shall be <= num_swb_long_window or num_swb_short_window as appropriate for window_sequence and sampling frequency and core coder frame length
ms_mask_present	no restrictions apply
ms_used	no restrictions apply
igf_ms_mask_present	no restrictions apply
ms_used	no restrictions apply
common_tw	no restrictions apply
common_tns	no restrictions apply
tns_on_lr	no restrictions apply
tns_present_both	no restrictions apply
tns_data_present[1]	no restrictions apply

5.4.9 fd_channel_stream()

global_gain	no restrictions apply
noise_level	no restrictions apply
noise_offset	no restrictions apply
ltpf_data_present	no restrictions apply

ltpf_pitch_lag_index	no restrictions apply
ltpf_gain_index	no restrictions apply
fdp_data_present	no restrictions apply
fdp_spacing_index	no restrictions apply
prev_aliasing_symmetry	no restrictions apply
curr_aliasing_symmetry	no restrictions apply
igf_AllZero	shall be 1 if one of the following conditions is true: — m_igfStartSfb == m_igfStopSfb — all values in igf_curr[ ][ ][ ] == 0 — swb_offset[m_igfStartSfb] ≤ igfMin
igf_level	no restrictions apply
fac_data_present	shall be 0, if the core_mode of the preceding frame of the same channel was 0 or if mod[3] of the preceding frame of the same channel was > 0

5.4.10 lpd_channel_stream()

tns_data_present	no restrictions apply
window_shape	no restrictions apply
max_sfb	no restrictions apply
acelp_core_mode	no restrictions apply
lpd_mode	if fullbandLpd == 0, then lpd_mode shall have a value between and including 0 and 25. If fullbandLpd == 1, then lpd_mode shall have a value between and including 0 and 4.
bpf_control_info	no restrictions apply
core_mode_last	shall be encoded with the value of data element core_mode of the previous frame
fac_data_present	shall be 0, if mod[0] of the current frame is > 0 and the core_mode of the preceding frame of the same channel was 0 ("TCX follows FD"), or if mod[0] of the current frame is > 0 and mod[nbDiv-1] of the preceding frame of the same channel was > 0 ("TCX follows TCX"), or if mod[0] of the current frame is 0 and mod[nbDiv-1] of the preceding frame of the same channel was 0 ("ACELP follows ACELP")

short_fac_flag	shall be encoded with a value of 1 if the window_sequence of the previous frame was 2 (EIGHT_SHORT_SEQUENCE). Otherwise short_fac_flag shall be encoded with a value of 0.

5.4.11 acelp_coding()

mean_energy	no restrictions apply
acb_index	no restrictions apply
ltp_filtering_flag	no restrictions apply
icb_index	no restrictions apply
gains	no restrictions apply

5.4.12 tcx_coding ()

noise_factor	no restrictions apply
global_gain	no restrictions apply
ltpf_data_present	no restrictions apply
ltpf_pitch_lag_index	no restrictions apply
ltpf_gain_index	no restrictions apply
fdp_data_present	no restrictions apply
fdp_spacing_index	no restrictions apply
igf_AllZero	shall be 1 if one of the following conditions is true: — m_igfStartSfb == m_igfStopSfb — all values in igf_curr[ ][ ][ ] == 0 — swb_offset[m_igfStartSfb] ≤ igfMin
igf_level	no restrictions apply
arith_reset_flag	no restrictions apply

5.4.13 lpd_stereo_stream()

res_mode	no restrictions apply
q_mode	no restrictions apply
ipd_mode	no restrictions apply
pred_mode	no restrictions apply
cod_mode	no restrictions apply
ild_idx	no restrictions apply

pred_gain_idx	no restrictions apply
cod_gain_idx	no restrictions apply

5.4.14 igf_stereo_pred_data()

igf_stereo_pred_all	no restrictions apply
cplx_pred_used	no restrictions apply
igf_pred_dir	no restrictions apply
igf_delta_code_time	no restrictions apply
hcod_sf	no restrictions apply

5.4.15 igf_data()

igf_UsePrevTileIdx	no restrictions apply
igfCurrTileIdx	no restrictions apply
igf_UsePrevWhiteningLevel	no restrictions apply
igf_WhiteningLevel	no restrictions apply
remainingTilesDifferent	no restrictions apply
igf_WhiteningLevel	no restrictions apply
igfApplyTNF	no restrictions apply

5.4.16 tbe_data()

tbe_heMode	no restrictions apply
idxFrameGain	no restrictions apply
idxSubGains	no restrictions apply
lsf_idx[0]	no restrictions apply
lsf_idx[1]	no restrictions apply
tbe_hrConfig	no restrictions apply
tbe_nlConfig	no restrictions apply
idxMixConfig	no restrictions apply
idxShbFrGain	no restrictions apply
idxResSubGains	no restrictions apply
idxShbExcResp[0]	no restrictions apply
idxShbExcResp[1]	no restrictions apply

5.4.17 tw_data()

The tw_data() bitstream structure and all recursively included bitstream structures shall be restricted as specified in ISO/IEC 23003-3:2020.

5.4.18 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.

5.4.19 tns_data()

The tns_data() bitstream structure and all recursively included bitstream structures shall be restricted as specified in ISO/IEC 23003-3:2020, 7.8 (Table 135)

5.4.20 ac_spectral_data()

The ac_spectral_data() bitstream structure and all recursively included bitstream structures shall be restricted as specified in ISO/IEC 23003-3:2020.

5.4.21 arith_data()

The arith_data() bitstream structure and all recursively included bitstream structures shall be restricted as specified in ISO/IEC 23003-3:2020.

5.4.22 fac_data()

The fac_data() bitstream structure and all recursively included bitstream structures shall be restricted as specified in ISO/IEC 23003-3:2020.

5.4.23 code_book_indices()

The code_book_indices() bitstream structure and all recursively included bitstream structures shall be restricted as specified in ISO/IEC 23003-3:2020.

5.4.24 UsacSbrData()

The UsacSbrData() bitstream structure and all recursively included bitstream structures shall be restricted as specified in ISO/IEC 23003-3:2020.

5.4.25 Mps212Data()

The Mps212Data() bitstream structure and all recursively included bitstream structures shall be restricted as specified in ISO/IEC 23003-3:2020.

5.5 Fill element

fill_byte should be '10100101'

5.6 MPEG surround configuration, SpatialSpecificConfig()

The SpatialSpecificConfig() bitstream structure and all recursively included bitstream structures shall be restricted as specified in ISO/IEC 23003-1.

5.7 MPEG surround frame, SpatialFrame()

The SpatialFrame() bitstream structure and all recursively included bitstream structures shall be restricted as specified in ISO/IEC 23003-1.

5.8 SAOC configuration, SAOCSpecificConfig()

The SAOCSpecificConfig() bitstream structure and all recursively included bitstream structures shall be restricted as specified in ISO/IEC 23003-2.

5.9 SAOC frame, SAOCFrame()

The SAOCFrame() bitstream structure and all recursively included bitstream structures shall be restricted as specified in ISO/IEC 23003-2.

5.10 AudioPreRoll

5.10.1 Recursive presence of AudioPreRoll extension payload

An access unit which is part of an AudioPreRoll shall not have usacExtElementPresent equal to 1 for the extension payload type ID_EXT_ELE_AUDIOPREROLL. That means there shall be no recursively embedded AudioPreRoll extension payload.

5.10.2 AudioPreRoll()

configLen	no restrictions apply
applyCrossfade	no restrictions apply
reserved	should be 0
numPreRollFrames	no restrictions apply
auLen	no restrictions apply

5.11 Dynamic range control configuration

5.11.1 mpegh3daUniDrcConfig()

drcCoefficientsUniDrcCount	no restrictions apply
drcInstructionsUniDrcCount	no restrictions apply
drcInstructionsType	no restrictions apply
mae_groupID	shall match the mae_groupID value of a present mae_GroupDefinition() structure
mae_groupPresetID	shall match the mae_groupPresetID value of a present mae_GroupPresetDefinition() structure
uniDrcConfigExtPresent	no restrictions apply
loudnessInfoSetPresent	no restrictions apply

5.11.2 mpegh3daUniDrcChannelLayout()

baseChannelCount no restrictions apply

5.11.3 drcCoefficientsUniDrc()

The drcCoefficientsUniDrc() bitstream structure shall be restricted as specified in ISO/IEC 23003-4:2020, Clause 9.

5.11.4 drcInstructionsUniDrc()

The drcInstructionUniDrc() bitstream structure shall be restricted as specified in ISO/IEC 23003-4:2020, Clause 9.

5.11.5 uniDrcConfigExtension()

The uniDrcConfigExtension() bitstream structure shall be restricted as specified in ISO/IEC 23003-4:2020, Clause 9.

5.12 Dynamic range control frame, uniDrcGain()

The uniDrcGain() bitstream structure shall be restricted as specified in ISO/IEC 23003-4:2020, Clause 9.

5.13 Object metadata configuration, ObjectMetadataConfig()

lowDelayMetadataCoding no restrictions apply

hasCoreLength no restrictions apply

frameLength no restrictions apply

hasScreenRelativeObjects no restrictions apply

isScreenRelativeObject no restrictions apply

hasDynamicObjectPriority no restrictions apply

hasUniformSpread no restrictions apply

5.14 Object metadata frame

5.14.1 object_metadata_efficient()

object_metadata_present no restrictions apply

5.14.2 object_metadata()

No restrictions are applicable to this bitstream element.

5.14.3 object_metadata_efficient()

has_differential_metadata shall be 0 if **usacIndependencyFlag** == 1

5.14.4 intracoded_object_metadata_efficient()

ifperiod	The number of audio frames for which OAM data are processed as indicated by **ifperiod** shall not exceed subsequent frames for which **usacIndependencyFlag** == 1.
common_azimuth	no restrictions apply
default_azimuth	no restrictions apply
position_azimuth	no restrictions apply
common_elevation	no restrictions apply
default_elevation	no restrictions apply
position_elevation	no restrictions apply
common_radius	no restrictions apply
default_radius	no restrictions apply
position_radius	no restrictions apply
common_gain	no restrictions apply
default_gain	no restrictions apply
gain_factor	no restrictions apply
common_spread	no restrictions apply
default_spread	no restrictions apply
default_spread_width	no restrictions apply
default_spread_height	no restrictions apply
default_spread_depth	no restrictions apply
spread	no restrictions apply
spread_width	no restrictions apply
spread_height	no restrictions apply
spread_depth	no restrictions apply
common_dynamic_object_priority	no restrictions apply
default_dynamic_object_priority	no restrictions apply
dynamic_object_priority	no restrictions apply

5.14.5 differential_object_metadata()

bits_per_point	no restrictions apply
fixed_azimuth	no restrictions apply
flag_azimuth	no restrictions apply

nbits_azimuth	no restrictions apply
differential_azimuth	no restrictions apply
fixed_elevation	no restrictions apply
flag_elevation	no restrictions apply
nbits_elevation	no restrictions apply
differential_elevation	no restrictions apply
fixed_radius	no restrictions apply
flag_radius	no restrictions apply
nbits_radius	no restrictions apply
differential_radius	no restrictions apply
fixed_gain	no restrictions apply
flag_gain	no restrictions apply
nbits_gain	no restrictions apply
differential_gain	no restrictions apply
fixed_spread	no restrictions apply
flag_spread	no restrictions apply
nbits_spread	no restrictions apply
differential_spread	no restrictions apply
flag_spread_width	no restrictions apply
nbits_spread_width	no restrictions apply
differential_spread_width	no restrictions apply
flag_spread_height	no restrictions apply
nbits_spread_height	no restrictions apply
differential_spread_height	no restrictions apply
flag_spread_depth	no restrictions apply
nbits_spread_depth	no restrictions apply
differential_spread_depth	no restrictions apply
fixed_dynamic_object_priority	no restrictions apply
flag_dynamic_object_priority	no restrictions apply
nbits_dynamic_object_priority	no restrictions apply
differential_dynamic_object_priority	no restrictions apply

5.14.6 offset_data()

bitfield_syntax	no restrictions apply
offset_bitfield	no restrictions apply
npoints	no restrictions apply
foffset	no restrictions apply

5.14.7 object_metadata_low_delay()

has_intracoded_object_metadata	shall be 1 if **usacIndependencyFlag** == 1

5.14.8 intracoded_object_metadata_low_delay()

fixed_azimuth	no restrictions apply
default_azimuth	no restrictions apply
common_azimuth	no restrictions apply
default_azimuth	no restrictions apply
position_azimuth	no restrictions apply
fixed_elevation	no restrictions apply
default_elevation	no restrictions apply
common_ elevation	no restrictions apply
default_elevation	no restrictions apply
position_elevation	no restrictions apply
fixed_radius	no restrictions apply
default_radius	no restrictions apply
common_radius	no restrictions apply
default_radius	no restrictions apply
position_radius	no restrictions apply
fixed_gain	no restrictions apply
default_gain	no restrictions apply
common_gain	no restrictions apply
default_gain	no restrictions apply
gain_factor	no restrictions apply
fixed_spread	no restrictions apply
default_spread	no restrictions apply

default_spread_width	no restrictions apply
default_spread_height	no restrictions apply
default_spread_depth	no restrictions apply
common_spread	no restrictions apply
default_spread	no restrictions apply
spread	no restrictions apply
spread_width	no restrictions apply
spread_height	no restrictions apply
spread_depth	no restrictions apply
fixed_dynamic_object_priority	no restrictions apply
default_dynamic_object_priority	no restrictions apply
common_dynamic_object_priority	no restrictions apply
default_dynamic_object_priority	no restrictions apply
dynamic_object_priority	no restrictions apply

5.14.9 dynamic_object_metadata()

flag_absolute	no restrictions apply
has_object_metadata	no restrictions apply

5.14.10 single_dynamic_object_metadata()

position_azimuth	no restrictions apply
position_elevation	no restrictions apply
position_radius	no restrictions apply
gain_ factor	no restrictions apply
spread	no restrictions apply
spread_width	no restrictions apply
spread_height	no restrictions apply
spread_depth	no restrictions apply
dynamic_object_priority	no restrictions apply
nbits	no restrictions apply
flag_azimuth	no restrictions apply
position_azimuth_difference	no restrictions apply

flag_elevation	no restrictions apply
position_elevation_difference	no restrictions apply
flag_radius	no restrictions apply
position_radius_difference	no restrictions apply
flag_gain	no restrictions apply
gain_factor_difference	no restrictions apply
flag_spread	no restrictions apply
spread_difference	no restrictions apply
flag_spread_width	no restrictions apply
spread_width_difference	no restrictions apply
flag_spread_height	no restrictions apply
spread_height_difference	no restrictions apply
flag_spread_depth	no restrictions apply
spread_depth_difference	no restrictions apply
flag_dynamic_object_priority	no restrictions apply
dynamic_object_priority_difference	no restrictions apply

5.15 EnhancedObjectMetadataConfig()

hasDiffuseness	no restrictions apply
hasCommonGroupDiffuseness	no restrictions apply
hasExcludedSectors	no restrictions apply
hasCommonGroupExcludedSectors	no restrictions apply
useOnlyPredefinedSectors	no restrictions apply
hasClosestSpeakerCondition	no restrictions apply
closestSpeakerThresholdAngle	no restrictions apply
hasDivergence	no restrictions apply
divergenceAzimuthRange	no restrictions apply
useOnlyPredefinedSectors	no restrictions apply

5.16 EnhancedObjectMetadataFrame()

keepDiffuseness	no restrictions apply
diffuseness	no restrictions apply

keepExclusion	no restrictions apply
numExclusionSectors	no restrictions apply
excludeSectorIndex	shall be one of the non-reserved values as defined in ISO/IEC 23008-3:2019, Table 146
usePredefinedSector	no restrictions apply
excludeSectorIndex	no restrictions apply
excludeSectorMinAzimuth	no restrictions apply
excludeSectorMaxAzimuth	no restrictions apply
excludeSectorMinElevation	no restrictions apply
excludeSectorMaxElevation	no restrictions apply
closestSpeakerPlayout	no restrictions apply
keepDiffuseness	no restrictions apply
diffuseness	no restrictions apply
keepDivergence	no restrictions apply
divergence	no restrictions apply
keepExclusion	no restrictions apply
numExclusionSectors	no restrictions apply
excludeSectorIndex	no restrictions apply
usePredefinedSector	no restrictions apply
excludeSectorIndex	no restrictions apply
excludeSectorMinAzimuth	no restrictions apply
excludeSectorMaxAzimuth	no restrictions apply
excludeSectorMinElevation	no restrictions apply
excludeSectorMaxElevation	no restrictions apply

5.17 SAOC 3D Config

5.17.1 SAOC3DSpecificConfig()

bsSamplingFrequencyIndex	shall be equal to **usacSamplingFrequencyIndex**
bsSamplingFrequency	shall be equal to **usacSamplingFrequency**
bsFreqRes	shall not be 0
bsDoubleFrameLengthFlag	no restrictions apply

bsNumSaocDmxChannels	shall be equal to (numSAOCTransportChannels – bsNumSaocDmxObjects)
bsNumSaocDmxObjects	shall be equal to (numSAOCTransportChannels – bsNumSaocDmxChannels)
bsDecorrelationMethod	no restrictions apply
bsNumSaocObjects	no restrictions apply
bsRelatedTo	no restrictions apply
bsOneIOC	no restrictions apply
bsSaocDmxMethod	no restrictions apply
bsDualMode	no restrictions apply
bsBandsLow	no restrictions apply
bsDcuFlag	shall be 0
bsDcuMandatory	shall be 0
bsDcuDynamic	shall be 0
bsDcuMode	shall be 0
bsDcuParam	shall be 0

5.17.2 SAOC3DgetNumChannels()

No restrictions are applicable to this bitstream element.

5.17.3 SAOC3DExtensionConfig()

No restrictions are applicable to this bitstream element.

5.17.4 SAOC3DExtensionConfigData()

No restrictions are applicable to this bitstream element.

5.17.5 SAOCExtensionConfig()

The SAOCExtensionConfig() bitstream structure and all sub bitstream structures shall be restricted as specified in ISO/IEC 23003-2.

5.18 SAOC 3D frame

5.18.1 Saoc3DFrame()

bsIndependencyFlag	shall be 1 if **usacIndependencyFlag** == 1
bsDcuDynamicUpdate	no restrictions apply
bsDcuMode	no restrictions apply
bsDcuParam	no restrictions apply

5.18.2 SAOC3DFramingInfo()

bsFramingType	no restrictions apply
bsNumParamSets	no restrictions apply
bsParamSlot	no restrictions apply

5.18.3 EcDataSaoc()

The SAOCExtensionConfig() bitstream structure and all sub bitstream structures shall be restricted as specified in ISO/IEC 23003-2.

5.18.4 ByteAlign()

The SAOCExtensionConfig() bitstream structure and all sub bitstream structures shall be restricted as specified in ISO/IEC 23003-2.

5.18.5 SAOC3DExtensionFrame()

No restrictions are applicable to this bitstream element.

5.18.6 SAOC3DExtensionFrameData()

No restrictions are applicable to this bitstream element.

5.18.7 SAOCExtensionFrame()

The SAOCExtensionConfig() bitstream structure and all sub bitstream structures shall be restricted as specified in ISO/IEC 23003-2.

5.18.8 HOAConfig()

HoaOrder	shall NOT have values [30, ..., 38]
IsScreenRelative	no restrictions apply
UsesNfc	no restrictions apply
NfcReferenceDistance	no restrictions apply

5.18.9 HOADecoderConfig()

MinAmbHoaOrder	shall be smaller than or equal to $\min(floor(\sqrt{numHOATransportChannels} - 1), HoaOrder)$ shall be according to ISO/IEC 23008-3:2019, Table 14
SingleLayer	no restrictions apply
codedLayerCh	shall be smaller than NumOfAdditionalCoders
CodedSpatialInterpolationTime	shall be coded according to ISO/IEC 23008-3:2019, Table 209
SpatialInterpolationMethod	no restrictions apply
CodedVVecLength	shall NOT be 3

MaxGainCorrAmpExp	no restrictions apply
HOAFrameLengthIndicator	shall NOT be 3 according to ISO/IEC 23008-3:2019, Table 208
DiffOrder	the value of **DiffOrder** shall not result in a MaxHoaOrderToBeTransmitted > HOAOrder. **DiffOrder** < **HOAOrder** - **MinAmbHoaOrder**
NumVVecVqElementsBits	restricted to values [0..7]
UsePhaseShiftDecorr	no restrictions apply

5.18.10 HOAEnhConfig()

LayerIdx	shall be [0, ..., NumLayers – 1]

5.18.11 HOADecoderEnhConfig ()

MaxNoOfDirSigsForPrediction	no restrictions apply
NoOfBitsPerScalefactor	no restrictions apply
PredSubbandsIdx	no restrictions apply
bsNumOfPredSubbands	no restrictions apply
FirstSBRSubbandIdx	no restrictions apply
MaxNumOfPredDirsLog2	no restrictions apply
MaxNumOfPredDirsPerBand	no restrictions apply
DirGridTableIdx	shall NOT be 3
ParSubbandTableIdx	no restrictions apply
bsNumOfParSubbands	no restrictions apply
LastFirstOrderSubbandIdx	no restrictions apply
UseRealCoeffsPerParSubband	no restrictions apply

5.18.12 getSubbandWidths ()

CodedBwFirstBand	no restrictions apply
bw_diff	no restrictions apply

5.19 HOA frame

5.19.1 HOAFrame()

hoaIndependencyFlag	shall be 1 if **usacIndependencyFlag** == 1

5.19.2 HOAEnhFrame ()

No restrictions are applicable to this bitstream element.

5.19.3 ChannelSideInfoData()

ChannelType	shall be coded according to ISO/IEC 23008-3:2019, Table 195
ActiveDirsIds	shall be [0, …, 900] according to ISO/IEC 23008-3:2019, Table F.9
NewChannelTypeOne	no restrictions apply
NbitsQ	if **hoaIndependencyFlag** == 1 then the two MSBs of NbitsQ shall not be 00 binary
CodebkIdx	shall NOT be 4, 5, 6
CbFlag	no restrictions apply
bA	no restrictions apply
bB	no restrictions apply
uintC	no restrictions apply
PFlag	no restrictions apply
CbFlag	no restrictions apply

5.19.4 AddAmbHoaInfoChannel()

AmbCoeffTransitionState	shall NOT be 3
CodedAmbCoeffIdx	shall be smaller than MaxNumAddActiveAmbCoeffs
AmbCoeffIdxTransition	no restrictions apply

5.19.5 HOAGainCorrectionData()

GainCorrPrevAmpExp	no restrictions apply
CodedGainCorrectionExp	no restrictions apply
GainCorrectionException	no restrictions apply

5.19.6 VVectorData()

VvecIdx

shall be smaller than or equal to	dependent on	
	CodebkIdx	HoaOrder
899	0	N/A
33	1, 2	N/A
63	3	N/A
31	7	= 4
$(\text{HoaOrder} + 1)^2 - 1$	7	≠ 4

SgnVal	no restrictions apply
WeightIdx	no restrictions apply
VecVal	no restrictions apply
huffVal	shall be Huffman-coded according to ISO/IEC 23008-3:2019, Tables F.15 – F.24
intAddVal	no restrictions apply

5.19.7 HOAPredictionInfo()

PSPredictionActive	no restrictions apply
KindOfCodedPredIds	no restrictions apply
NumActivePredIds	no restrictions apply
PredIds	no restrictions apply
ActivePred	no restrictions apply
PredDirSigIds	no restrictions apply
PredGains	no restrictions apply

5.19.8 HOADirectionalPredictionInfo()

UseDirectionalPrediction	no restrictions apply
KeepPreviousPredDirsFlag	no restrictions apply
NumOfGlobalPredDirs	no restrictions apply
GlobalPredDirsIds	no restrictions apply
KeepPreviousDirPredMatrixFlag	no restrictions apply
UseHuffmanCodingDiffAngle	no restrictions apply
DirIsActive	no restrictions apply
RelDirGridIdx	no restrictions apply

5.19.9 readDirPredDiffValues()

DecodedMagDiff	no restrictions apply
DecodedAngleDiff	no restrictions apply

5.19.10 HOAParInfo ()

UsePar	no restrictions apply
KeepPreviousParMatrixFlag	no restrictions apply
ParDecorrSigsSelectionTableIdx	no restrictions apply

UseReducedNoOfUpmixSigs	no restrictions apply
UseParUpmixSig	no restrictions apply
UseParHuffmanCodingDiffAbs	no restrictions apply
UseParHuffmanCodingDiffAngle	no restrictions apply

5.19.11 readParDiffValues ()

HuffmanCodedParMagDiff	shall be Huffman-coded according to ISO/IEC 23008-3:2019, Table F.37
HuffmanCodedRealParMagDiff	shall be Huffman-coded according to ISO/IEC 23008-3:2019, Table F.38
runLengthCodedVal	no restrictions apply
CodedParMagDiff	no restrictions apply
runLengthCodedVal	no restrictions apply
HuffCodedParAngleDiff	shall be Huffman-coded according to ISO/IEC 23008-3:2019, Table F.39
CodedParAngleDiff	no restrictions apply

5.20 FMT converter frame, FormatConverterFrame()

rendering3DType	no restrictions apply

5.21 Multi-channel coding tool config, MCTConfig ()

mctChanMask	the number of all non-zero entries in mctChanMask[] shall be greater than or equal to 2

5.22 Multi-channel coding tool frame

5.22.1 MultichannelCodingBoxRotation ()

channelPairIndex	shall have a value between and including 0 and *nMCTChannels*·(*nMCTChannels*–1)/2–1, with nMCTChannels = number of all non-zero entries in mctChanMask[]
hasMctMask	no restrictions apply
hasBandwiseAngles	no restrictions apply
isMCTShort	no restrictions apply
numMaskBands	no restrictions apply
mctMask	no restrictions apply

mct_delta_time	**mct_delta_time** shall be 0 if **MCTSignalingType** of the previous and the current frame changes between one of the following values: 0 and 1, 0 and 3, 1 and 2, or 2 and 3 (and vice versa)
hcod_angle	no restrictions apply

5.22.2 MultichannelCodingBoxPrediction ()

channelPairIndex	shall have a value between and including 0 and *nMCTChannels*·(*nMCTChannels*–1)/2–1, with nMCTChannels = number of all non-zero entries in mctChanMask[]
hasMctMask	no restrictions apply
hasBandwiseCoeff	no restrictions apply
isMCTShort	no restrictions apply
numMaskBands	no restrictions apply
mctMask	no restrictions apply
pred_dir	no restrictions apply
mct_delta_time	**mct_delta_time** shall be 0 if **MCTSignalingType** of the previous and the current frame changes between one of the following values: 0 and 1, 0 and 3, 1 and 2, or 2 and 3 (and vice versa)
hcod_sf	no restrictions apply

5.22.3 MultichannelCodingFrame()

MCTSignalingType	no restrictions apply
keepTree	no restrictions apply
numPairs	no restrictions apply
hasStereoFilling	no restrictions apply

5.23 Tonal component coding configuration, TccConfig ()

tccMode[]	shall not be 3

5.24 Tonal component coding frame

5.24.1 General

Conformance test bitstream shall contain tonal component coding frame compressed data stored in TccGroupOfSegments() as defined in ISO/IEC 23008-3.

Each compressed data shall meet the syntactic and semantic requirements specified in ISO/IEC 23008-3.

5.24.2 TccGroupOfSegments()

tccDataPresent	no restrictions apply
numTrajectories	no restrictions apply
isContinued	no restrictions apply
segLength	no restrictions apply
amplQuant	no restrictions apply
amplTransformCoeffDC	no restrictions apply
amplTransformIndex	The corresponding **huffWord s**hall be encoded with the values listed in the Huffman table, `huff_idxTab`, defined in ISO/IEC 23008-3:2019, Table 119
amplTransformCoeffAC	The corresponding **huffWord s**hall be encoded with the values listed in the Huffman table, `huff_acTab`, defined in ISO/IEC 23008-3:2019, Table 120
amplSgn	no restrictions apply
freqQuant	no restrictions apply
freqTransformCoeffDC	no restrictions apply
freqTransformIndex	The corresponding **huffWord s**hall be encoded with the values listed in the Huffman table, `huff_idxTab`, defined in ISO/IEC 23008-3:2019, Table 119
freqTransformCoeffAC	The corresponding **huffWord s**hall be encoded with the values listed in the Huffman table, `huff_acTab`, defined in ISO/IEC 23008-3:2019, Table 120
freqSgn	no restrictions apply

5.25 HREP config, HREPConfig()

extendedGainRange	no restrictions apply
extendedBetaFactorPrecision	no restrictions apply
isHREPActive	no restrictions apply
lastFFTLine	lastFFTLine shall be greater than or equal to transitionWidthLines
transitionWidthLines	transitionWidthLines shall be smaller than or equal to lastFFTLine
defaultBetaFactorIdx	no restrictions apply
useCommonSettings	no restrictions apply

5.26 HREP frame, HREPFrame()

useRawCoding	no restrictions apply
gainIdx	no restrictions apply

useDefaultBetaFactorIdx	no restrictions apply
betaFactorIdx	no restrictions apply

5.27 ICG config, ICGConfig ()

ICPresent	no restrictions apply
ICinCPE	no restrictions apply
ICGPreAppliedPresent	no restrictions apply
ICGPreAppliedCPE	no restrictions apply

5.28 SignalGroupInformation Config, SignalGroupInformation ()

groupPriority	no restrictions apply
fixedPosition	no restrictions apply

5.29 DownmixMatrix

5.29.1 downmixConfig()

downmixConfigType	shall not be 3
passiveDownmixFlag	no restrictions apply
phaseAlignStrength	no restrictions apply
immersiveDownmixFlag	no restrictions apply

5.29.2 DownmixMatrixSet()

downmixIdCount	no restrictions apply
downmixId	shall not be 0x00, shall not be 0x7F. All values of downmixID within one DownmixMatrixSet() shall be pairwise distinct
downmixType	shall be 0 or 1
CICPspeakerLayoutIdx (downmixType == 0)	shall be encoded in accordance with ChannelConfiguration as defined in ISO/IEC 23091-3
CICPspeakerLayoutIdx (downmixType == 1)	shall be encoded in accordance with ChannelConfiguration as defined in ISO/IEC 23091-3
bsDownmixMatrixCount	no restrictions apply
bsNumAssignedGroupIDs	no restrictions apply
Signal_groupID	no restrictions apply
DmxMatrixLenBits	no restrictions apply

5.29.3 DownmixMatrix()

equalizerPresent	no restrictions apply
precisionLevel	shall not be 3
maxGain	no restrictions apply
minGain	no restrictions apply
isAllSeparable	no restrictions apply
isSeparable	no restrictions apply
isAllSymmetric	no restrictions apply
isSymmetric	no restrictions apply
mixLFEOnlyToLFE	no restrictions apply
rawCodingCompactMatrix	no restrictions apply
compactDownmixMatrix	no restrictions apply
useCompactTemplate	no restrictions apply
runLGRParam	no restrictions apply
zeroRunLength	no restrictions apply
compactDownmixMatrix	no restrictions apply
fullForAsymmetricInputs	no restrictions apply
rawCodingNonzeros	no restrictions apply
gainLGRParam	no restrictions apply

5.29.4 DecoderGainValue()

gainValueIndex	no restrictions apply

5.29.5 ReadRange()

range	no restrictions apply
rangeExtra	no restrictions apply

5.29.6 EqualizerConfig()

numEqualizers	no restrictions apply
eqPrecisionLevel	no restrictions apply
eqExtendedRange	no restrictions apply
numSections	no restrictions apply
qFactorIndex	no restrictions apply
qFactorExtra	no restrictions apply

centerGainIndex	no restrictions apply
scalingGainIndex	no restrictions apply
hasEqualizer	no restrictions apply

5.30 Loudness info

5.30.1 mpegh3daLoudnessInfoSet()

loudnessInfoCount	no restrictions apply
loudnessInfoType	no restrictions apply
mae_groupID	shall match the mae_groupID value of a present mae_GroupDefinition() structure
mae_groupPresetID	shall match the mae_groupPresetID value of a present mae_GroupPresetDefinition() structure
loudnessInfoAlbumPresent	no restrictions apply
loudnessInfoAlbumCount	no restrictions apply
loudnessInfoSetExtensionPresent	no restrictions apply

5.30.2 loudnessInfo()

The uniDrcGainExtension() bitstream structure shall be restricted as specified in ISO/IEC 23003-4:2020, Clause 9.

5.30.3 loudnessInfoSetExtension()

The uniDrcGainExtension() bitstream structure shall be restricted as specified in ISO/IEC 23003-4:2020, Clause 9.

5.31 Audioscene info

5.31.1 mae_AudioSceneInfo

mae_isMainStream	no restrictions apply
mae_audioSceneInfoIDPresent	no restrictions apply
mae_audioSceneInfoID	no restrictions apply
mae_numGroups	no restrictions apply
mae_numSwitchGroups	no restrictions apply
mae_numGroupPresets	no restrictions apply
mae_bsMetaDataElementIDoffset	no restrictions apply
mae_metaDataElementIDmaxAvail	no restrictions apply

5.31.2 mae_Data()

mae_numDataSets	no restrictions apply
mae_dataType	each data type shall only occur once at most
mae_dataLength	no restrictions apply
tmp	no restrictions apply

5.31.3 mae_GroupDefinition()

mae_groupID	shall not be the same value for different groups
mae_allowOnOff	no restrictions apply
mae_defaultOnOff	no restrictions apply
mae_allowPositionInteractivity	shall be identical for groups contained in the same switch group
mae_interactivityMinAzOffset	shall be identical for groups contained in the same switch group
mae_interactivityMaxAzOffset	shall be identical for groups contained in the same switch group
mae_interactivityMinElOffset	shall be identical for groups contained in the same switch group
mae_interactivityMaxElOffset	shall be identical for groups contained in the same switch group
mae_interactivityMinDistFactor	shall be identical for groups contained in the same switch group
mae_interactivityMaxDistFactor	shall be identical for groups contained in the same switch group
mae_allowGainInteractivity	shall be identical for groups contained in the same switch group
mae_interactivityMinGain	shall be identical for groups contained in the same switch group
mae_interactivityMaxGain	shall be identical for groups contained in the same switch group
mae_bsGroupNumMembers	no restrictions apply
mae_hasConjunctMembers	no restrictions apply
mae_startID	no restrictions apply
mae_metaDataElementID	groups that are part of switch groups shall consist of one or multiple complete signalGroups

5.31.4 mae_SwitchGroupDefinition()

mae_switchGroupID	shall not be the same value for different switch groups

mae_switchGroupAllowOnOff	no restrictions apply
mae_switchGroupDefaultOnOff	no restrictions apply
mae_bsSwitchGroupNumMembers	no restrictions apply
mae_switchGroupMemberID	shall only reference groups which have a value of mae_allowOnOff of 1
mae_switchGroupDefaultGroupID	no restrictions apply

5.31.5 mae_Description()

mae_bsNumDescriptionBlocks	no restrictions apply
mae_descriptionGroupID	shall only address values of mae_groupID once at the most
mae_descriptionSwitchGroupID	shall only address values of mae_switchGroupID once at the most
mae_descriptionGroupPresetID	shall only address values of mae_groupPresetID once at the most
mae_bsNumDescLanguages	no restrictions apply
mae_bsDescriptionLanguage	no restrictions apply
mae_bsDescriptionDataLength	no restrictions apply
mae_descriptionData	no restrictions apply

5.31.6 mae_ContentData()

mae_bsNumContentDataBlocks	no restrictions apply
mae_ContentDataGroupID	no restrictions apply
mae_contentKind	no restrictions apply
mae_hasContentLanguage	no restrictions apply
mae_contentLanguage	no restrictions apply

5.31.7 mae_CompositePair()

mae_bsNumCompositePairs	no restrictions apply
mae_CompositeElementID	no restrictions apply

5.31.8 mae_GroupPresetDefinition()

mae_groupPresetID	shall not be the same value for different presets
mae_groupPresetKind	no restrictions apply
mae_bsGroupPresetNumConditions	no restrictions apply

mae_groupPresetReferenceID	shall only reference groups or switch groups which have a value of mae_allowOnOff of 1 or mae_switchGroupAllowOnOff of 1, respectively not more than one condition shall be defined for a group not more than one condition shall be defined for a switch group not more than one condition shall be defined for groups which are part of the same switch group for a group which is part of a switch group no condition shall be defined if a condition for the switch group is already defined
mae_groupPresetConditionOnOff	no restrictions apply
mae_groupPresetDisableGainInteractivity	shall be 1 if mae_allowGainInteractivity of the referenced group is 0
mae_groupPresetGainFlag	shall be 0 if mae_allowGainInteractivity of the referenced group is 0
mae_groupPresetGain	no restrictions apply
mae_groupPresetDisablePositionInteractivity	shall be 1 if mae_allowPositionInteractivity of the referenced group is 0
mae_groupPresetPositionFlag	shall be 0 if mae_allowPositionInteractivity of the referenced group is 0
mae_groupPresetAzOffset	no restrictions apply
mae_groupPresetElOffset	no restrictions apply
mae_groupPresetDistFactor	no restrictions apply

5.31.9 mae_ProductionScreenSizeData()

hasNonStandardScreenSize	no restrictions apply
bsScreenSizeAz	no restrictions apply
bsScreenSizeTopEl	no restrictions apply
bsScreenSizeBottomEl	no restrictions apply

5.31.10 mae_LoudnessCompensationData ()

mae_loudnessCompGroupLoudnessPresent	no restrictions apply
mae_bsLoudnessCompGroupLoudness	no restrictions apply
mae_loudnessCompDefaultParamsPresent	no restrictions apply
mae_loudnessCompDefaultIncludeGroup	no restrictions apply

mae_loudnessCompDefaultMinMaxGainPresent	no restrictions apply
mae_bsLoudnessCompDefaultMinGain	no restrictions apply
mae_bsLoudnessCompDefaultMaxGain	no restrictions apply
mae_loudnessCompPresetParamsPresent	no restrictions apply
mae_loudnessCompPresetIncludeGroup	no restrictions apply
mae_loudnessCompPresetMinMaxGainPresent	no restrictions apply
mae_bsLoudnessCompPresetMinGain	no restrictions apply
mae_bsLoudnessCompPresetMaxGain	no restrictions apply

5.31.11 mae_ProductionScreenSizeDataExtension()

mae_overwriteProductionScreenSizeData	no restrictions apply
bsScreenSizeLeftAz	no restrictions apply
bsScreenSizeRightAz	no restrictions apply
mae_NumPresetProductionScreens	no restrictions apply
mae_productionScreenGroupPresetID	no restrictions apply
mae_hasNonStandardScreenSize	no restrictions apply
isCenteredInAzimuth	no restrictions apply
bsScreenSizeAz	no restrictions apply
bsScreenSizeLeftAz	no restrictions apply
bsScreenSizeRightAz	no restrictions apply
bsScreenSizeTopEl	no restrictions apply
bsScreenSizeBottomEl	no restrictions apply

5.31.12 mae_GroupPresetDefinitionExtension()

mae_hasSwitchGroupConditions	no restrictions apply
mae_isSwitchGroupCondition	no restrictions apply
mae_hasDownmixIdGroupPresetExtensions	no restrictions apply
mae_numDownmixIdGroupPresetExtensions	no restrictions apply
mae_groupPresetDownmixId	no restrictions apply
mae_bsGroupPresetNumConditions	no restrictions apply
mae_isSwitchGroupCondition	no restrictions apply
mae_groupPresetSwitchGroupID	shall only reference switch groups which have a value of mae_switchGroupAllowOnOff of 1

	not more than one condition shall be defined for a switch group
mae_groupPresetGroupID	shall only reference groups which have a value of mae_allowOnOff of 1 not more than one condition shall be defined for a group not more than one condition shall be defined for groups which are part of the same switch group for a group which is part of a switch group no condition shall be defined if a condition for the switch group is already defined
mae_groupPresetConditionOnOff	no restrictions apply
mae_groupPresetDisableGainInteractivity	shall be 1 if mae_allowGainInteractivity of the referenced group is 0
mae_groupPresetGainFlag	shall be 0 if mae_allowGainInteractivity of the referenced group is 0
mae_groupPresetGain	no restrictions apply
mae_groupPresetDisablePositionInteractivity	shall be 1 if mae_allowPositionInteractivity of the referenced group is 0
mae_groupPresetPositionFlag	shall be 0 if mae_allowPositionInteractivity of the referenced group is 0
mae_groupPresetAzOffset	no restrictions apply
mae_groupPresetElOffset	no restrictions apply
mae_groupPresetDistFactor	no restrictions apply

5.31.13 mae_DrcUserInterfaceInfo()

version	no restrictions apply
bsNumTargetLoudnessConditions	no restrictions apply
bsTargetLoudnessValueUpper	no restrictions apply
drcSetEffectAvailable	no restrictions apply

5.32 HOA matrix

5.32.1 HoaRenderingMatrixSet()

numOfHoaRenderingMatrices	no restrictions apply
HoaRenderingMatrixId	no restrictions apply
CICPSpeakerLayoutIdx	shall be encoded in accordance with ChannelConfiguration as defined in ISO/IEC 23091-3

HoaMatrixLenBits	no restrictions apply

5.32.2 HoaRenderingMatrix()

precisionLevel	no restrictions apply
isNormalized	no restrictions apply
gainLimitPerHoaOrder	no restrictions apply
maxGain	no restrictions apply
minGain	no restrictions apply
isFullMatrix	no restrictions apply
firstSparseOrder	shall be smaller than or equal to maxHoaOrder
hasLfeRendering	no restrictions apply
numPairs	no restrictions apply
zerothOrderAlwaysPositive	no restrictions apply
isAllValueSymmetric	no restrictions apply
isAnyValueSymmetric	no restrictions apply
boolVal (valueSymmetricPairs)	no restrictions apply
isAllSignSymmetric	no restrictions apply
isAnySignSymmetric	no restrictions apply
boolVal (signSymmetricPairs)	no restrictions apply
isAllSignSymmetric	no restrictions apply
isAnySignSymmetric	no restrictions apply
boolVal (signSymmetricPairs)	no restrictions apply
hasVerticalCoef	no restrictions apply

5.32.3 DecoderHoaMatrixData()

hasValue	no restrictions apply
signVal	no restrictions apply

5.32.4 DecoderHoaGainValue()

No restrictions are applicable to this bitstream element.

5.33 CompatibleProfileLevelSet()

bsNumCompatibleSets	no restrictions apply
reserved	shall be 0

CompatibleSetIndication	no restrictions apply

5.34 Restrictions depending on profiles and levels

5.34.1 General

Depending on the profile and level associated with the MPEG-H 3DA bitstream, further restriction may apply.

5.34.2 Low complexity profile

5.34.2.1 General

Additional to the restrictions specified in this document, all MPEG-H bitstreams which claim to comply to the low complexity profile, shall fulfil the profile restrictions defined in ISO/IEC 23008-3:2019, 4.8.2.2.

5.34.2.2 mpegh3daConfig()

The variables usacSamplingFrequencyIndex, usacSamplingFrequency and coreSbrFrameLengthIndex shall be encoded with a value specified in Table 5.

Table 5 — Specification of mpegh3daConfig()

Restriction applies to	Level				
	1	2	3	4	5
usacSamplingFrequencyIndex	0x03 - 0x08, 0x1f	0x03 - 0x08, 0x1f	0x03 - 0x08, 0x1f	0x03 - 0x08, 0x1f	0x00 - 0x08, 0x1f
usacSamplingFrequency	29400, 14700	29400, 14700	29400, 14700	29400, 14700	58800, 29400, 14700
coreSbrFrameLengthIndex	1	1	1	1	1

5.34.2.3 mpegh3daDecoderConfig()

The variable **elementLengthPresent** shall be 1, if the Configuration Extension type ID_CONFIG_EXT_AUDIOSCENE_INFO exists and the value **mae_numSwitchGroups** in bitstream structure *mae_AudioSceneInfo*() is larger than 0.

5.34.2.4 Mpegh3daChannelPairElementConfig()

The variable **qceIndex** shall be 0.

5.34.2.5 mpegh3daCoreConfig()

The variable **tw_mdct** shall be 0.

In case **igfStartIndex** signals an audio frequency higher than 8 kHz, **igfUseEnf** shall be 0.

5.34.2.6 StereoCoreToolInfo()

If the independent noise filling (INF) of the intelligent gap filling (IGF) is activated (i.e. if **igfUseEnf**==1), then the complex prediction tool shall be restricted to real-only prediction, i.e. **complex_coef** shall be 0.

If stereo filling is activated (i.e. if **stereo_filling**==1), then the complex prediction tool shall be restricted to real-only prediction, i.e. **complex_coef** shall be 0.

5.34.2.7 MultichannelCodingFrame()

The variable **numPairs** shall be encoded with a value specified in Table 6.

Table 6 — Specification of numPairs in MultichannelCodingFrame()

Restriction applies to	Level				
	1	2	3	4	5
numPairs	<=5	<=9	<=16	<=28	<=28

5.34.2.8 mpegh3daExtElement() for extension payload type ID_EXT_ELE_AUDIOPREROLL

The value **usacExtElementPresent** for extension payload type ID_EXT_ELE_AUDIOPREROLL shall be 1 at maximum once per 0.5 seconds of audio data.

5.34.2.9 AudioPreRoll()

The value **numPreRollFrames** shall be encoded with 0 or 1.

5.34.2.10 mpegh3daUniDrcConfig()

The value **drcCoefficientsUniDrcCount** shall be encoded with a value not larger than 4. The value **drcInstructionsUniDrcCount** shall be encoded with a value specified in Table 7. **loudnessInfoSetPresent** shall be encoded with a value of 0.

Table 7 — Specification of drcInstructionUniDrcCount in mpegh3daUniDrcConfig()

Restriction applies to	Level				
	1	2	3	4	5
drcInstructionsUniDrcCount	<=16	<=16	<=32	<=32	<=32

5.34.2.11 drcCoefficientsUniDrc()

The values **drcFrameSizePresent** and **timeDeltaMinPresent** shall be encoded with 0. The value for **gainInterpolationType** shall be encoded with 1. The value **bandCount** shall be encoded with a value specified in Table 8.

Table 8 — Specification of bandCount in drcCoefficientsUniDrc()

Restriction applies to	Level				
	1	2	3	4	5
bandCount	<=2	<=4	<=4	<=4	<=4

5.34.2.12 drcInstructionsUniDrc()

The value **dependsOnDrcSetPresent** shall be encoded with 0 if **downmixId** is equal to 0. Additionally, **bsSequenceIndex** shall be unique in simultaneously applied DRC sets except for **bsSequenceIndex** equal to 0. **bsSequenceIndex** shall be identical for all channels if **downmixId** is not equal to 0.

5.34.2.13 ObjectMetadataConfig() in extension config type ID_EXT_ELE_OBJ_METADATA

The value **lowDelayMetadataCoding** shall be encoded with 1. The variable **frameLength** shall be encoded with 3, 7 or 15.

5.34.2.14 HOAConfig()

HoaOrder — shall be encoded as defined in Table 9

Table 9 — Restrictions for HoaOrder

Restriction applies to HoaOrder	Level				
	1	2	3	4	5
in case UsesNfc == 0	<=2	<=4	<=6	<=6	<=6
in case UsesNfc == 1	N/A	<=1	<=2	<=3	<=3

UsesNfc — shall be encoded as defined in Table 10

Table 10 — Restrictions for UsesNfc

Restriction applies to	Level				
	1	2	3	4	5
UsesNfc	0	no restrictions			
There shall not be more than one occurrence of **UsesNfc** == 1 in signal groups of type SignalGroupTypeHOA.					

5.34.2.15 HOADecoderConfig()

HOAFrameLengthIndicator — shall not be 3

NOTE The value of this bitstream syntax element has no effect in LC Profile, because the outputFrameLength of the core decoder is always 1024 (see ISO/IEC 23008-3:2019, Table 208)

UsePhaseShiftDecorr — shall not be 1

5.34.2.16 HOADecoderEnhConfig()

MaxNoOfDirSigsForPrediction — shall be encoded according to Table 11

Table 11 — Restrictions for bsMaxNoOfDirSigsForPrediction

Restriction applies to	Level				
	1	2	3	4	5
bsMaxNoOfDirSigsForPrediction	<=1	<=2	no restrictions		

PredSubbandsIdx — shall be 0

ParSubbandTableIdx — shall be 0

5.34.2.17 mae_AudioSceneInfo()

The values **mae_numGroups**, **mae_numSwitchGroups** and **mae_numGroupPresets** shall be encoded with a value specified in Table 12.

Table 12 — Specification of mae_AudioSceneInfo()

Restriction applies to	Level				
	1	2	3	4	5
mae_numGroups	<=5	<=9	<=16	<=28	<=28
mae_numSwitchGroups	<=2	<=4	<=8	<=14	<=14
mae_numGroupPresets	<=4	<=4	<=8	<=16	<=31

5.34.2.18 mae_GroupPresetDefinition()

The value **mae_bsGroupPresetNumConditions** shall be encoded with a value specified in Table 13.

Table 13 — Specification of mae_bsGroupPresetNumConditions in mae_GroupPresetDefinition()

Restriction applies to	Level				
	1	2	3	4	5
mae_bsGroupPresetNumConditions	<5	<9	<16	<16	<16

5.34.2.19 mae_Description()

The values **mae_bsNumDescLanguages** and **mae_bsDescriptionDataLength** shall be encoded with a value specified in Table 14.

Table 14 — Specification of mae_Description()

Restriction applies to	Level				
	1	2	3	4	5
mae_bsNumDescLanguages	<4	<4	<4	<8	<16
mae_bsDescriptionDataLength	<256	<256	<256	<256	<256

5.34.2.20 mae_GroupPresetDefinitionExtension()

The value **mae_numDownmixIdGroupPresetExtensions** shall be encoded with a value specified in Table 15.

Table 15 — Specification of mae_numDownmixIdGroupPresentExtensions in mae_GroupPresetDefinitionExtension()

Restriction applies to	Level				
	1	2	3	4	5
mae_numDownmixIdGroupPresetExtensions	<=4	<=4	<=8	<=16	<=31

5.34.2.21 downmixConfig()

phaseAlignStrength shall be 0

5.34.2.22 EnhancedObjectMetadataConfig

If **spread**[o] or **spread_width**[o] of a given object o is not 0, then **divergence**[o] shall be 0.

If **divergence**[o] of a given object o is not 0, then **spread**[o] shall be 0 (in case of uniform spread) and **spread_width**[o] shall be 0 (in case of non-uniform spread).

6 MPEG-H 3D audio interfaces to the MPEG-H 3D audio decoder

6.1 Characteristics and test procedure

Characteristics of the decoder interfaces specify the constraints that are applicable to the interface parameters which are provided to the decoder. These syntactic and semantic constraints may, for example, restrict the range or the values of parameters that are provided directly or indirectly to the decoder.

Each MPEG-H 3DA audio decoder interface shall meet the syntactic and semantic requirements specified in this document. The present clause defines the conformance criteria that shall be fulfilled by compliant decoder interface implementations. These criteria are specified for the syntactic elements of the decoder interface. For interfaces, which can also be realised as part of a bitstream (such as the interface for user interaction) the same constraints apply.

For each interface a set of semantic tests to be performed on the interface parameters is described. To verify whether the syntax is correct is straightforward and therefore not defined herein after. 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.

6.2 Interface for local setup information

6.2.1 mpegh3daLocalSetupInformation()

bsRenderingType	no restrictions apply
bsNumWIREoutputs	shall not be larger than (numAudioChannels + numAudioObjects)
WireID	no restrictions apply
hasLocalScreenSizeInformation	no restrictions apply

6.2.2 LoudspeakerRendering()

bsNumLoudspeakers	no restrictions apply
hasLoudspeakerDistance	no restrictions apply
hasLoudspeakerCalibrationGain	no restrictions apply
useTrackingMode	no restrictions apply
hasKnownPosition	no restrictions apply
loudspeakerAzimuth	no restrictions apply
loudspeakerElevation	no restrictions apply
loudspeakerDistance	shall not be 0
loudspeakerCalibrationGain	no restrictions apply
externalDistanceCompensation	no restrictions apply

6.2.3 BinauralRendering()

No specific restrictions apply to the elements of BinauralRendering() and its further referenced bitstream elements, such as: BinauralFirData(), FdBinauralRendererParam(), VoffBrirParam(), SfrBrirParam(), QtdlBrirParam(), TdBinauralRendererParam().

6.2.4 LocalScreenSizeInformation()

isCenteredInAzimuth	no restrictions apply
bsLocalScreenSizeAz	no restrictions apply
bsLocalScreenSizeLeftAz	no restrictions apply
bsLocalScreenSizeRightAz	no restrictions apply
hasLocalScreenElevationInformation	no restrictions apply
bsLocalScreenSizeTopEl	no restrictions apply
bsLocalScreenSizeBottomEl	no restrictions apply

6.3 Interface for user interaction

6.3.1 mpegh3daElementInteraction()

ei_InteractionSignatureDataLength	no restrictions apply
ei_InteractionSignatureDataType	no restrictions apply
ei_InteractionSignatureData	no restrictions apply
hasLocalZoomAreaSize	no restrictions apply

6.3.2 ElementInteractionData ()

ei_interactionMode	in case mae_numGroupPresets equals zero ei_interactionMode shall be zero, otherwise ei_interactionMode shall be one
ei_numGroups	shall be the same number as mae_numGroups
ei_groupPresetID	shall contain a valid preset ID (ID of a preset defined in the bitstream)

6.3.3 ei_GroupInteractivityStatus ()

ei_groupID	shall contain a valid groupID (ID of a group defined in the bitstream)
ei_onOff	no restrictions apply
ei_routeToWIRE	no restrictions apply
routeToWireID	no restrictions apply
ei_changePosition	shall be 1 if the value mae_allowPositionInteractivity of the corresponding group equals 1

ei_azOffset	no restrictions apply
ei_elOffset	no restrictions apply
ei_distFact	no restrictions apply
ei_changeGain	shall be set to 1 if the value mae_allowGainInteractivity of the corresponding group equals 1
ei_gain	no restrictions apply

6.3.4 LocalZoomAreaSize()

bsZoomAzCenter	no restrictions apply
bsZoomAz	no restrictions apply
bsZoomElCenter	no restrictions apply
bsZoomEl	no restrictions apply

6.4 Interface for loudness normalization and dynamic range control

The uniDrcInterface() bitstream structure and all sub-structures shall be restricted as specified in ISO/IEC 23003-4:2020, Clause 9.

6.5 Interface for scene displacement data, mpegh3daSceneDisplacementData()

sd_yaw	no restrictions apply
sd_pitch	no restrictions apply
sd_roll	no restrictions apply

6.6 Interface for positional scene displacement data, mpegh3daPositionalSceneDisplacementData()

sd_azimuth	no restrictions apply
sd_elevation	no restrictions apply
sd_radius	no restrictions apply

7 MPEG-H 3D audio decoders

7.1 General

The set of conformance test conditions described in this document shall be applied to verify that a given MPEG-H 3DA decoder implementation complies with this document. Test conditions are designed to test each tool in isolation and thus, determine the requirements for the corresponding conformance criteria. However, some tools show interactions and dependencies. To cover such dependencies, test cases are defined that can be composed of one or more test conditions.

Every conformance test case results in one or more conformance test sequences. One line in the spreadsheet accessible at https://standards.iso.org/iso-iec/23008/-9/ed-2/en represents one conformance test sequence. The tool or tool combination tested by a given test sequence can be deduced from its filename, as it follows the nomenclature defined in 4.3.2.

To claim conformance for the MPEG-H 3DA LC Profile, the output of the implementation under test shall meet the conformance test criteria against the reference waveform by applying the appropriate conformance test tools. The required conformance test tools as well as the conformance test criteria of each conformance test sequence are listed in the spreadsheet accessible at https://standards.iso.org/iso-iec/23008/-9/ed-2/en. All conformance test tools are defined in 4.3.3.

If a test case defines default behaviour, this should be fulfilled by all conformance test sequences if not specified different explicitly. For example, default behaviour for a tool can be that it is switched off.

The frequency of occurrence for independent frames (frames with **usacIndependencyFlag** == 1) should be at least once every 50 frames in each conformance test bitstream.

If not otherwise specified, the audio signal representation shall not leave the allowed value range of the chosen data representation at any point in the decoder processing chain. In other words, no "clipping" shall occur at any point in the decoder processing chain.

The conformance test sequences can be found at https://standards.iso.org/iso-iec/23008/-9/ed-2/en.

7.2 Basic conformance test conditions

7.2.1 Element configuration test condition

7.2.1.1 General

This test condition shall be applied to verify the proper decoder behaviour in case of several signal groups, reference layouts, speaker configurations, signals of channels, objects, SAOC, HOA and the definitions for SCE, CPE and LFE.

7.2.1.2 Test sequences

The test bitstream shall contain the configuration which is given by a specific 3daC_ElemConfIdx as shown in Table 16. This specifies a reference Layout, which shall be one of the 3daC_SpeakerConfIdx defined in Table 17. Additionally the 3daC_ElemConfIdx defines the Signalgroups and their order. If the SignalGroupType is **SignalGroupTypeChannels**, it can be either the reference Layout ("r") or a 3daC_SpeakerConfIdx, which is defined in Table 17. If the signalGroupType is **SignalGroupTypeObject**, it is specified with Obj<numOfChannels>. If the signalGroupType is **SignalGroupTypeHOA,** it is specified with Hoa<numOfChannels>. Also the 3daC_ElemConfIdx specifies the order of the usacElementTypes ID_USAC_SCE, ID_USAC_CPE, ID_USAC_LFE. Please note that there might be some ID_USAC_EXT Elements added, which are not defined in this test case.

Table 16 — 3daC_ElemConfIdx for conformance

3daC_ElemConfIdx	Reference layout (defined by 3daC_SpeakerConfIdx)	Signalgroups (defined by 3daC_SpeakerConfIdx)	Core coder elements with order in the element loop in mpegh3daFrame()
C0	0	r	16 × SCE
C1	1	r	SCE
C2	2	r	CPE
C6	6	r	CPE – SCE –LFE – CPE
C19	19	r	CPE – SCE –LFE – CPE – CPE –CPE – CPE
C100	6	r	6 × SCE
C102	6	Two signal groups: r, 2	6 × SCE – CPE
C103	2	Two signal groups: r, r	2 × CPE

Table 16 *(continued)*

3daC_Elem ConfIdx	**Reference layout (defined by 3daC_ SpeakerConfIdx)**	**Signalgroups (defined by 3daC_ SpeakerConfIdx)**	**Core coder elements with order in the element loop in mpegh3daFrame()**
C104	104	r	CPE – SCE – CPE – CPE – CPE
O1	6	Obj<1>	SCE
O9	6	Obj<9>	9 × SCE
O20	19	Obj<20>	20 x SCE
O24	19	Obj<24>	24 x SCE
H1	6	Hoa<1>	SCE
H4	6	Hoa<4>	4 × SCE
H8	6	Hoa<8>	8 × SCE
H9	6	Hoa<9>	9 × SCE
H16	6	Hoa<16>	16 × SCE
M1	6	Three signal groups: r , Obj<2>, Hoa<8>	SCE – CPE –LFE – CPE – 2 × SCE – 8 × SCE

Table 17 — 3daC_SpeakerConfIdx for conformance

3daC_SpeakerConfIdx	**Description**
0	speakerLayoutType == 3 numSpeakers = 16
1	speakerLayoutType == 0 CICPSpeakerLayoutIdx=1
2	speakerLayoutType == 0 CICPSpeakerLayoutIdx=2
6	speakerLayoutType == 0 CICPSpeakerLayoutIdx=6
7	speakerLayoutType == 0 CICPSpeakerLayoutIdx=7
15	speakerLayoutType == 0 CICPSpeakerLayoutIdx=15
16	speakerLayoutType == 0 CICPSpeakerLayoutIdx=16
19	speakerLayoutType == 0 CICPSpeakerLayoutIdx=19

Table 17 *(continued)*

3daC_SpeakerConfIdx	Description
104	speakerLayoutType == 1 numSpeakers = 9 CICPSpeakerIdx[0] = 2 CICPSpeakerIdx[1] = 0 CICPSpeakerIdx[2] = 1 CICPSpeakerIdx[3] = 4 CICPSpeakerIdx[4] = 5 CICPSpeakerIdx[5] = 17 CICPSpeakerIdx[6] = 18 CICPSpeakerIdx[7] = 30 CICPSpeakerIdx[8] = 31

7.2.2 Sampling rate

7.2.2.1 General

This test condition shall be applied to verify the proper decoder behaviour for different sampling frequencies.

7.2.2.2 Test sequences

The test bitstream shall contain the configuration which is given by a specific 3daC_SampFreqIdx defined in Table 18.

Table 18 — 3daC_SampFreqIdx for conformance

3daC_SampFreqIdx	Description	Corresponding sampling frequency as defined in ISO/IEC 23003-3 Hz
0	usacSamplingFrequencyIndex=0x0	96 000
1	usacSamplingFrequencyIndex=0x1	88 200
2	usacSamplingFrequencyIndex=0x2	64 000
3	usacSamplingFrequencyIndex=0x3	48 000
4	usacSamplingFrequencyIndex=0x4	44 100
5	usacSamplingFrequencyIndex=0x5	32 000
6	usacSamplingFrequencyIndex=0x6	24 000
7	usacSamplingFrequencyIndex=0x7	22 050
8	usacSamplingFrequencyIndex=0x8	16 000
32	usacSamplingFrequencyIndex=0x1F usacSamplingFrequency=14700	N/A
33	usacSamplingFrequencyIndex=0x1F usacSamplingFrequency=29 400	N/A
34	usacSamplingFrequencyIndex=0x1F usacSamplingFrequency=58 800	N/A

7.2.3 Core mode tests [Fd|Lpd|Cct]

7.2.3.1 General

This test condition shall be applied to verify the proper decoder behaviour for the core modes.

7.2.3.2 Test sequences

The test condition shall contain one of the following 3 conditions:

— FD only [Fd];

— LPD only [Lpd];

— toggle between FD and LPD [Cct] (short for "common core coding tools").

In case of FD only, the core_mode in bitstream structure mpegh3daCoreData() shall always be 0.

In case of LPD only, the core_mode in bitstream structure mpegh3daCoreData() shall always be 1. Note that if an MPEG-H 3DA LFE exists, this channel will be encoded only with FD.

In case of switching between FD and LPD, the following conditions shall be fulfilled:

— If a MPEG-H 3DA CPE exists, at least one MPEG-H 3DA CPE shall toggle between FD and LPD simultaneously.

— If at least two MPEG-H 3DA CPE exists, at least one MPEG-H 3DA CPE shall toggle between FD and LPD differently for each channel, so that at least 25 % of the frames are coded differently.

— For each pair of channels (without the two channels of one MPEG-H 3DA CPE and MPEG-H 3DA LFE) there values for core_mode shall be differently for at least one frame.

— Each channel (without MPEG-H 3DA LFE) shall toggle at least 10 times from FD to LPD and 10 times from LPD to FD.

7.3 Additional test conditions

7.3.1 3D audio core (FD)

7.3.1.1 Basic FD window test condition [Win]

7.3.1.1.1 General

This test condition shall be applied to verify the proper decoder for all meaningful FD window sequence transitions. Meaningful window sequence transitions are listed in ISO/IEC 23003-3:2020, 7.9.3.1 (Table 138).

7.3.1.1.2 Test sequences

This test condition shall contain the following conditions:

— For all channels (except MPEG-H 3DA LFE) every meaningful FD window sequence transition shall be triggered.

— For all channels (except MPEG-H 3DA LFE) every meaningful FD window sequence transition shall be triggered with sine window (window_shape 0) and KBD window (window_shape 1).

— If there is at least one MPEG-H 3DA CPE, **common_window** shall be at least one time 1 and at least one time 0.

— If there are at least two MPEG-H 3DA CPEs, for at least one of them **window_sequence** shall be different for both channels in at least 50 % of the frames and **window_shape** shall be different for at least 25 % of the frames and all combinations of **window_sequence** and **window_shape** for **ch** equal 1 and **window_sequences** and **window_shape** for **ch** equal 2 shall be signalled at least once.

7.3.1.1.3 Default behaviour

If this test condition is not active, the default behaviour should be that all FD frames use always ONLY_LONG_SEQUENCE and sine window shape (window_shape equals 0).

7.3.1.2 Non-meaningful FD window switching test condition [Nmf]

7.3.1.2.1 General

This test condition should be applied to monitor the decoder behaviour in case FD window sequence transitions not specified in ISO/IEC 23003-3:2020, 7.9.3.1 (Table 138) that occur in a given bitstream.

7.3.1.2.2 Test sequences

All non-meaningful FD window transitions shall be signalled at least once with sine window shape (window_shape equals 0) and at least once with KBD window shape (window_shape equals 1) in every channel (except MPEG-H 3DA LFE).

For every channel where FD mode and LPD mode is used, there shall be at least one transition from FD to LPD where the FD-window has signalled LONG_WINDOW and LONG_STOP_WINDOW with window shape sine and KBD and at least one transition from LPD to FD where FD has signalled LONG_WINDOW and LONG_START_WINDOW with window shape sine and KBD.

For every channel where FD mode and LPD mode is used, there shall be at least one transition from FD to LPD where either **prev_aliasing_symmetry** or **curr_aliasing_symmetry** or both equal 1.

It shall be assured that the decoder does not crash during decoding.

The decoder behaviour at non-meaningful FD window transitions is not covered by this document, hence no decoded waveforms are provided.

7.3.1.2.3 Default behaviour

If this test condition is not active, the default behaviour should be that all FD frames use always ONLY_LONG_SEQUENCE and sine window shape (window_shape equals 0).

7.3.1.3 Aliasing symmetries test condition [Asy]

7.3.1.3.1 General

This test condition shall be applied to verify the proper behaviour of the extended aliasing symmetries of MPEG-H 3D audio.

7.3.1.3.2 Test sequences

This test condition shall contain the following conditions:

— For all channels (except MPEG-H 3DA LFE) every combination of **prev_aliasing_symmetry** and **curr_aliasing_symmetry** shall be triggered.

— **usacIndependencyFlag** shall be 1 for at least 10 % of the frames.

7.3.1.3.3 Default behaviour

If this test condition is not active, the default behaviour should be that all FD frames trigger **curr_aliasing_symmetry** and **prev_aliasing_symmetry** equal to 0.

7.3.1.4 Noise filling test condition [Nf]

7.3.1.4.1 General

This test condition shall be applied to verify the proper behaviour of the noise filling tool of MPEG-H 3D audio and the correct signalling of its parameters.

7.3.1.4.2 Test sequences

This test condition shall contain the following conditions:

— For all audio elements **noiseFilling** in *mpegh3daCoreConfig()* shall be 1 (except MPEG-H 3DA LFE).

— The values of **noise_level** and **noise_offset** vary from frame to frame. All possible combinations of **noise_filling** and **noise_offset** shall be triggered in every audio channel stream at least once, without the combination of **noise_filling** equal to 0 and **noise_offset** not equal to 0.

— Each pair of two different audio channel streams (except MPEG-H 3DA LFE) shall use a different value of **noise_filling** or **noise_offset** for at least one frame.

7.3.1.4.3 Default behaviour

If this test condition is not active, the default behaviour should be to set **noiseFilling** to 0 in *mpegh3daCoreConfig()* for all audio elements.

7.3.1.5 Varying max_sfb test condition [Sfb]

7.3.1.5.1 General

This test condition shall be applied to ensure the correct decoder behaviour in case varying values of **max_sfb** are signalled by the bitstream.

7.3.1.5.2 Test sequences

The value of **max_sfb** transmitted in ics_info() varies in the range from 0 to the maximum. The upper bound is determined by the given sampling rate.

If at least one MPEG-H 3DA CPE exists, at least one MPEG-H 3DA CPE shall signal different values of **max_sfb** in more than 50 % of the frames for each channel. This MPEG-H 3DA CPE shall use **common_window** equals 1 at least once while transmitting two different **max_sfb**.

7.3.1.5.3 Default behaviour

This conformance test condition does not have default behaviour.

7.3.1.6 TNS test condition [Tns]

7.3.1.6.1 General

This test condition shall be applied to verify the proper behaviour of the temporal noise shaping (TNS) tool of MPEG-H 3D audio and the correct signalling of its parameters.

7.3.1.6.2 Test sequences

All bitstreams containing TNS data are indicated by the bit **tns_active** in *StereoCoreToolInfo()* and **tns_data_present** in *mpegh3daCoreCoderData().* All TNS parameters mentioned in Table 19 shall be applied at least once for each audio channel element. For MPEG-H 3DA CPE, additionally the parameters in Table 20 shall be applied. Each pair of audio channel streams (except MPEG-H 3DA LFE) shall use at least for one frame different values.

Table 19 — TNS parameters

Bitstream field	Values
n_filt	0...3(0, 1)
coef_res	0, 1
length	1, maxSfb
order	15 (7), 7(3), 1
direction	0, 1
coef_compress	0, 1
Coef	0, 15 (7)
NOTE The values in parenthesis are applied to short blocks.	

Table 20 — TNS stereo parameters

Bitstream field	Values
Tns_data_present[1]	0, 1
Tns_on_lr	0, 1
Tns_present_both	0, 1
Common_tns	0, 1

7.3.1.6.3 Default behaviour

If this test condition is not active, the default behaviour should be to set **tns_active** to 0 in *StereoCoreToolsInfo()* and to set **tns_data_present** to 0 in *mpegh3daCoreCoderData()* for all audio channel streams.

7.3.1.7 M/S stereo test condition [Ms]

7.3.1.7.1 General

This test condition shall be applied to verify the proper behaviour of the M/S stereo tool of the MPEG-H 3DA decoder.

7.3.1.7.2 Test sequences

Bitstreams make use of the M/S stereo tool. An overview of affected bitstream parameters is shown in Table 21.

Table 21 — M/S stereo parameters

Bitstream element	Value	Description
ms_mask_present	0	M/S not active
	1	M/S active on some scale factor bands
	2	M/S active on all scale factor bands
ms_used	0, 1	Indicates the use of M/S stereo per scale factor band

All bitstreams activating the M/S stereo tool shall cover the values as described above for every channel. Additionally, for every scale factor band **ms_used** shall be set to 0 at least once and set to 1 at least once.

7.3.1.7.3 Default behaviour

If this test condition is not active, the default behaviour should be that **ms_mask_present** is set to 0.

7.3.1.8 Complex prediction stereo test condition [Cp]

7.3.1.8.1 General

This test condition shall be applied to ensure the functionality of the complex prediction stereo tool of the MPEG-H 3DA decoder.

7.3.1.8.2 Test sequences

Bitstreams activate the complex prediction stereo tool of MPEG-H 3DA. The affected bitstream values are listed in Table 22.

Table 22 — Complex prediction stereo parameters

Bitstream element	Value	Description
ms_mask_present	0	Complex prediction not active
	3	Complex prediction active
cplx_pred_used	0, 1	Indicates the use of complex prediction per prediction band
cplx_pred_all	0, 1	Complex prediction on all prediction bands
complex_coef	0, 1	Transmit complex coefficients (1) or real only coefficients(0)
delta_code_time	0, 1	Time differential coding (1) or frequency differential coding (0)
use_prev_frame	0, 1	Use only current frame (0) or use both current and previous frame (1) for MDST estimation
pred_dir	0, 1	Prediction from mid to side (0) or from side to mid (1)

All bitstreams activating the complex prediction stereo tool shall cover all values as described above for every channel. Additionally **cplx_pred_used** shall be for every sfb at least one time 0 and one time 1.

7.3.1.8.3 Default behaviour

If this test condition is not active, the default behaviour should be that **ms_mask_present** is set to 0.

7.3.1.9 Multichannel coding tool (MCT)

7.3.1.9.1 General

The MCT tool is very complex to define in only one conformance test condition. That is why there are several conformance test conditions for this tool, which all start with "M-". In case no MCT related conformance test condition is active (respectively no conformance test condition is active, which is defined in 7.3.1.9), no extension payload with ID_EXT_ELE_MCT shall be written.

7.3.1.9.2 MCT global configuration

7.3.1.9.3 MCT channel signalling [M-chM-<x>]

7.3.1.9.3.1 General

This test condition shall be applied to verify the proper behaviour of the bitstream syntax element **mctChanMask[]** defined in MCTConfig().

7.3.1.9.3.2 Test sequences

Bitstreams make use of the MCT and cover all possible activation states of distinct channels for MCT processing. Activation of MCT via **mctChanMask[]** is reflected by a hexadecimal value (<x>) that is appended to M-chM as follows: The bit representation of the hexadecimal value consists of all audio signals being part of the bit stream, irrespective of any grouping via signal groups and element type, where the least significant bit (LSB) represents the last audio signal and the most significant bit (MSB) represents the first audio signal. Note that an LFE element shall always be set to 0 and that if all elements of mctChanMask[] are zero for one signal group, no extension element with MCT payload is allowed.

7.3.1.9.3.3 Default behaviour

If this test condition is not active, the default behaviour should be to set **mctChanMask[]** to 1 for all audio channels **bsNumberOfSignals[grp]**+1 of a signal group grp except for LFE channels where it is 0.

7.3.1.9.4 MCT payload

7.3.1.9.4.1 MCT signalling Type [M-Typ-<x>]

General

This test condition shall be applied to verify the proper behaviour of the decoder for the bitstream syntax element **MCTSignalingType**.

Test sequences

This test shall contain test conditions according to Table 23:

Table 23 — MCTSignalingType

M-Typ	MCTSignalingType	Description
0	0	Prediction only
1	1	Rotation only
2	2	Prediction with Stereo Filling
3	3	Rotation with Stereo Filling
10	0,1,2,3	Switching of all MCTSignalingTypes (every transition shall occur at least once)

Table 23 *(continued)*

M-Typ	MCTSignalingType	Description
11	2,3	MCTSignalingType indicating Stereo Filling (this value shall toggle at least 20 times)

Additionally, **mct_delta_time** shall toggle at least 10 times in both directions. In case of several MCTSignalingTypes within one bitstream, **mct_delta_time** shall be 1 when switching from 0 to 2 or 1 to 3 or vice versa at least once if possible.

Default behaviour

If this condition is not active, the default behaviour should be to use MCTSignalingType=0, exclusively.

7.3.1.9.4.2 MCT stereo filling [M-SFi-<x>]

General

This test condition shall be applied to verify the proper behaviour of the decoder for the bitstream syntax element **hastStereoFilling[]** when **MCTSignalingType** is larger than 1.

Test sequences

This test shall contain test conditions according to Table 24:

Table 24 — hasStereoFilling[]

MCTSteFi	hasStereoFilling	Description
0	0	Stereo Filling inactive
1	1	Stereo Filling active
2	0,1	Stereo Filling shall be active and inactive

Additionally **isMCTShort** shall toggle between 0 and 1 (note that this means that not only ONLY_LONG_SEQUENCE is used)

Default behaviour

If this test condition is not active, the default behaviour should be to use **hasStereoFilling**[pair]=0, for all pairs pair.

7.3.1.9.4.3 MCT mechanics [M-Mec]

General

This test condition shall be applied to verify the proper behaviour of the decoder for the bitstream syntax element **numPairs** and **keepTree** for different window sequences.

Test sequences

This test condition shall contain the following conditions:

— **numPairs** shall cover all valid values and remain constant over multiple frames;

— **keepTree** shall be 1 at least once for 10 consecutive frames.

Default behaviour

If this test condition is not active, the default behaviour should be to use a constant value for **numPairs** and **keepTree**=0.

7.3.1.9.4.4 MCT rotation content [M-Rot]

General

This test condition shall be applied to verify the proper behaviour of the decoder for bitstream syntax elements used for MCT rotation processing.

Test sequences

This test condition shall contain the following conditions:

— all combinations of **hasMctMask** and **hasBandwiseCoeff** shall be covered;

— **mct_delta_time** shall toggle between 0 and 1;

— **numMaskBands** shall cover all valid values;

— **mctMask[]** shall be 0 and 1 for any pair of scale factor bands throughout the test sequences where transmitted;

— **dpcm_beta[]** shall cover all possible values throughout the test sequences;

— **isMCTShort** shall toggle between 0 and 1 (note that this means that not only ONLY_LONG_SEQUENCE is used).

Default behaviour

If this condition is not active, the default behaviour should be to transmit **hasMctMask**=0, **hasBandwiseCoeff**=0 and a constant **dpcm_beta**[].

7.3.1.9.4.5 MCT prediction content [M-Pre]

General

This test condition shall be applied to verify the proper behaviour of the decoder for bitstream syntax elements used for MCT prediction processing.

Test sequences

This test condition shall contain the following conditions:

— all combinations of **hasMctMask** and **hasBandwiseCoeff** shall be covered;

— **pred_dir** shall toggle between 0 and 1;

— **mct_delta_time** shall toggle between 0 and 1;

— **numMaskBands** shall cover all valid values;

— **mctMask[]** shall be 0 and 1 for any pair of scale factor bands throughout the test sequences where transmitted;

— **dpcm_alpha_q_re[]** shall cover all possible values throughout the test sequences;

— **isMCTShort** shall toggle between 0 and 1 (note that this means that not only ONLY_LONG_SEQUENCE is used).

Default behaviour

If this condition is not active, the default behaviour should be to transmit **hasMctMask**=0, **hasBandwiseCoeff**=0 and a constant **dpcm_alpha_q_re**[].

7.3.1.10 Baseline Profile test condition [Bl-<l0-l1-...>]

7.3.1.10.1 General

This test condition shall be applied to verify the proper behaviour of the Baseline Profile integration of the MPEG-H 3DA decoder as specified in ISO/IEC 23008-3:2019/Amd 2:2020.

7.3.1.10.2 Test sequences

This test condition shall contain the following condition:

— mpegh3daConfig() shall include an **mpegh3daProfileLevelIndication** with a value between 0x10 ... 0x14, corresponding to Baseline Profile level 1 ... 5. The included Baseline Profile level indicated with **mpegh3daProfileLevelIndication** is depending on the parameter signalled with <l0-l1-...>, which is interpreted as follows: “1” = L1, “2” = L2, “3” = L3, “4” = L4, “5” = L5

7.3.1.10.3 Default behaviour

If this test condition is not active, the default behaviour should be that **mpegh3daProfileLevelIndication** is not set to a value between 0x10 ... 0x14.

7.3.2 3D audio core (LPD)

7.3.2.1 LPC coding test condition [Lpc]

7.3.2.1.1 General

The test condition shall be applied to verify the functionality of the linear predictive coding (LPC) filter and the proper decoding of LPC parameters in the bitstream.

7.3.2.1.2 Test sequences

The test bitstream shall be designed such that:

— If not specified differently (with a combination of several conformance test conditions), all LPD frames shall be encoded using MDCT-based TCX.

— For each LPC filter (LPC1, LPC2, LPC3 and LPC4 for no Full band LPD and LPC1 and LPC4 for Full band LPD), every possible absolute and relative quantization mode in ISO/IEC 23008-3:2019, Table 91 (fullbandLPD==1) or in ISO/IEC 23003-3:2020, Table 148 (fullbandLPD==0) is to be used at least once.

— All entries in the first stage approximation codebook are to be used at least once.

7.3.2.1.3 Default behaviour

If this conformance test condition is not active, the default behaviour should be that the LPC filters exhibit weak resonances.

7.3.2.2 ACELP core mode test condition [Ace]

7.3.2.2.1 General

This test condition shall be applied to verify the correct decoding of frames encoded with the ACELP coding scheme.

7.3.2.2.2 Test sequences

The test bitstream shall be designed such that:

— Every possible value of the bitfields mean_energy (4 possibilities, see ISO/IEC 23003-3:2020, Table 152), acb_index[·] (512 or 64 possibilities, depending on the subframe position), ltp_filtering_flag[·] (two possibilities) and gains[·] (128 possibilities) is to be used at least once.

— In order to guarantee a complete and balanced coverage of all 8 algebraic codebooks defined in ISO/IEC 23003-3:2020, 7.14.5.2, each algebraic codebooks shall be employed at least once and the frequency of occurrence of all codebooks should be evenly distributed.

— If several channels are encoded in a bitstream, each pair of channels shall be encoded with two different codebook types for at least one frame.

7.3.2.2.3 Default behaviour

If this conformance test condition is not active, the default behaviour should be that all frames are encoded using TCX mode without restriction regarding the length of TCX. If ACELP is employed nonetheless, the pitch gain should be less than 0.1.

7.3.2.3 TCX and noise filling test condition [Tcx]

7.3.2.3.1 General

This test condition shall be applied to verify the correct decoding of frames encoded with the TCX coding scheme. Furthermore, the TCX noise filling is covered.

7.3.2.3.2 Test sequences

The test bitstream shall be designed such that:

— noiseFilling is set to 1.

— Every possible value of the bitfields noise_factor (8 possibilities) and global_gain (128 possibilities) is used at least once in each channel. Additionally each pare of the channels shall be encoded with a different noise_factor and different global_gain at least for one frame.

— In order to guarantee a complete and balanced coverage of all MDCT window lengths and all transitions between these, the usage of the various MDCT window lengths is as follows:

— If Fullband LPD is active, where [x x] represents the two LPD coding modes mod[0..2] for one frame and 1, and 2 are the mode values that determine the MDCT window length as described in ISO/IEC 23008-3:2019, Table 87:

— [1 1] for at least 100 frames and at most 150 frames;

— [2 2] for at least 100 frames and at most 150 frames;

— [1 1] [2 2] for at least 100 frames and at most 150 frames.

— If Fullband LPD is inactive, where [x1 x2 x3 x4] represents the four LPD coding modes mod[0..3] for one frame and 1, 2 and 3 are the mode values that determine the MDCT window length as described in ISO/IEC 23003-3:2020, Table 97:

— [1 1 1 1] for at least 100 frames and at most 150 frames;

— [2 2 2 2] for at least 100 frames and at most 150 frames;

— [3 3 3 3] for at least 100 frames and at most 150 frames;

— [1 1 1 1] [1 1 2 2] [1 1 2 2] [2 2 2 2] [2 2 1 1] [2 2 1 1] [3 3 3 3] [2 2 2 2] [3 3 3 3] [3 3 3 3] for at least 100 frames and at most 150 frames.

— If several channels are encoded in the bitstream, each pare of the channels shall be encoded with two different coding modes for at least one frame.

7.3.2.3.3 Default behaviour

If this conformance test condition is not active, the default behaviour should be that all frames are encoded with tcx (no restriction regarding the length of tcx).

7.3.2.4 fullband LPD test condition [fbL-<x1>-<x2>-..]

7.3.2.4.1 General

This test condition shall be applied to verify the proper behaviour of the fullband LPD tool of the MPEG-H 3DA decoder.

7.3.2.4.2 Test sequences

The name of the test includes parameters (<x1>, <x2>, ..) corresponding to the elements present in *mpegh3daDecoderConfig()*, except for ID_USAC_LFE and ID_USAC_EXT. The parameter x<id> denotes if fullband LPD tool is active or inactive, when set to 0 or 1 respectively, in the valid channel elements of order <id>. All parameters are connected by "-".

The test bitstream shall be designed such that, all elements set **fullbandLpd** to 0 or 1 in *mpegh3daCoreConfig() corresponding to the parameter mentioned above*.

7.3.2.4.3 Default behaviour

If this test condition is not active, the default behaviour should be that fullband LPD is active for all elements i.e. **fullbandLpd** is set to 1 in *mpegh3daCoreConfig()*.

7.3.2.5 LPD mode coverage and FAC test condition [Lpd]

7.3.2.5.1 General

This test condition shall be applied to ensure the proper decoding of frames encoded in LPD mode. It also covers all allowed transitions between LPD coding schemes (ACELP / TCX).

7.3.2.5.2 Test sequences

The test bitstream shall be designed such that the following sequence is encoded:

— if Fullband LPD is active, all transitions of the different lpd_modes from ISO/IEC 23008-3:2019, Table 87 shall occur at least one time at every possible position.

— if Fullband LPD is inactive, all transitions of the different lpd_mode from ISO/IEC 23003-3:2020, Table 97 shall occur at least one time at every possible position.

7.3.2.5.3 Default behaviour

If this conformance test condition is not active, the default behaviour should be that all frames are encoded with TCX (no restriction regarding the length of tcx).

7.3.2.6 AVQ test condition [Avq]

7.3.2.6.1 General

This test condition shall be applied to test the AVQ quantization tool of the USAC decoder.

7.3.2.6.2 Test sequences

The test bitstream is designed such that:

— All frames are encoded using ACELP and short MDCT-based TCX. The sequence shall be as follows: [0 0] [1 1] [0 1] [1 0] (fullbandLPD==1) or [0 0 1 1] [1 1 0 0] [0 1 0 1] [1 0 1 0] (fullbandLPD==0).

— For quantization of the FAC information, every absolute leader from ISO/IEC 23003-3:2020, Table 146 is to be used at least once.

7.3.2.6.3 Default behaviour

This conformance test condition does not have default behaviour.

7.3.2.7 Stereo LPD test condition [sLP-<x1>-<x2>-..]

7.3.2.7.1 General

This test condition shall be applied to verify the proper behaviour of the stereo LPD tool of the MPEG-H 3DA decoder.

7.3.2.7.2 Test sequences

The name of the test includes parameters (<x1>, <x2>, ..) corresponding to the ID_USAC_CPE elements present in *mpegh3daDecoderConfig()*. The parameter x<id> denotes if stereo LPD tool is active or inactive, when set to 0 or 1 respectively, in the channel pair element of order <id>. All parameters are connected by "-".

The test bitstream shall be designed such that:

— All ID_USAC_CPE shall set the syntax element **lpdStereoIndex** to 0 or 1 in *mpegh3daChannelPairElementConfig()* corresponding to the parameter mentioned above.

— If **lpdStereoIndex** is set to 1, all possible values of the tool parameters as shown in Table 25 shall be tested.

Table 25 — LPD stereo parameters

Bitstream field	Values
res_mode	0,1
q_mode	0, 1
ipd_mode	0,1,2,3
pred_mode	0,1
cod_mode	0, 1,2,3

7.3.2.7.3 Default behaviour

If this test condition is not active, the default behaviour should be that LPD stereo is inactive i.e. **lpdStereoIndex** is set to 0 for all elements.

7.3.2.8 Time domain bandwidth extension test condition [Tbe]

7.3.2.8.1 General

This test condition shall be applied to test the time domain bandwidth extension from ACELP of the USAC decoder.

7.3.2.8.2 Test sequences

This test condition only affects channel elements where **fullbandLPD** is set to 1.

The affected bitstream values are listed in Table 26.

Table 26 — Time domain bandwidth extension parameters

Bitstream field	Values
tbe_heMode	0, 1
idxFrameGain	0...31
idxSubGains	0...31
lfs_idx[0]	0...127
lfs_idx[1]	0...127
tbe_hrConfig	0, 1
tbe_nlConfig	0, 1
idxMixConfig	0...3
idxShbFrGain	0...63
idxResSubGains	0...31
idxShbExcResp[0]	0...127
idxShbExcResp[1]	0...15

All bitstreams activating the time domain bandwidth extension test condition shall cover all values as described above for every channel where **fullbandLPD** is set to 1. Additionally all possible combinations of the values **tbe_nlConfig** and **idxMixConfig** shall be triggered. And for **tbe_heMode** equal to 0 the value **idxSubGains** shall be at least one time odd and one time even.

7.3.2.8.3 Default behaviour

If this conformance test condition is not active, the value of **tbe_heMode** should be 1 and **idxFrameGain** 0.

7.3.3 3D audio core (FD and LPD)

7.3.3.1 Frequency domain prediction test condition [Fdp]

7.3.3.1.1 General

This test condition shall be applied to verify the proper behaviour of the frequency domain prediction tool of MPEG-H 3D audio and the correct signalling of its parameters for the FD mode as well as for the LPD mode.

7.3.3.1.2 Test sequences

The test bitstream shall be designed such that:

— The value **fdp_data_present** shall be set to 1 for 20 consecutive frames at least once for each channel (except MPEG-H 3DA LFE).

— Every possible value (0 ... 255) shall be set for **fdp_spacing_index** for each channel (except MPEG-H 3DA LFE).

— For every channel **fdp_data_present** shall be set at least one time to 1 in the first frame after **usacIndependencyFlag** is set to 1.

— For every channel where FD mode and LPD mode is used, there shall be at least one transition from FD to LPD and at least one transition from LPD to FD where for the first frame after the transition **fdp_data_present** is set to 1.

— For every channel where LPD mode is used and sometimes **lg** is equal to **ccfl**, **fdp_data_present** shall be set at least one time to 1 in the first frame after **lg** is not equal to **ccfl**.

— For each pair of channels the **fdp_spacing_index** shall be different for at least one frame.

7.3.3.1.3 Default behaviour

If this conformance test condition is not active, the default behaviour should be that **fdp_data_present** is set to 0. Note that this value can occur in *fd_channel_stream()*, and *tcx_coding()*.

7.3.3.2 Long-term postfilter test condition [Lpf]

7.3.3.2.1 General

This test condition shall be applied to verify the proper behaviour of the long-term postfilter tool of MPEG-H 3D audio and the correct signalling of its parameters for the FD mode as well as for the LPD mode

7.3.3.2.2 Test sequences

The test bitstream shall be designed such that:

— the value **ltpf_data_present** shall be set to 1 for at least 80 % of the frames for each channel (except MPEG-H 3DA LFE);

— every possible value (0 ... 3) shall be set for **ltpf_gain_index** for each channel (except MPEG-H 3DA LFE);

— every possible value (0 ... 511) shall be set for **ltpf_pitch_lag_index**;

— if at least one MPEG-H 3DA CPE exists, **common_ltpf** shall be set at least once to 1 and at least once to 0;

— for every channel where FD mode and LPD mode is used , there shall be at least one transition from FD to LPD and at least one transition from LPD to FD where both frames of this channel have **ltpf_data_present** set to 1;

— for each pair of channels the **ltpf_gain_index** and **ltpf_pitch_lag_index** shall be different for at least one frame.

7.3.3.2.3 Default behaviour

If this conformance test condition is not active, the default behaviour should be that **ltpf_data_present** is set to 0. Note that this value can occur in *fd_channel_stream()*, *StereoCoreToolInfo()* and *tcx_coding()*.

7.3.3.3 Bass-post filter test condition [Bpf]

7.3.3.3.1 General

This test condition shall be applied to verify the behaviour of the bass-post filter of the USAC decoder in LPD coding mode.

7.3.3.3.2 Test sequences

This test condition only affects channel Elements where fullbandLPD is set to 0.

The test bitstream shall be designed such that for the channel Elements where fullbandLPD is set to 0:

— The frames are encoded using alternately the MDCT-based TCX coding mode and the ACELP coding mode. From time to time there shall be 5 to 10 consecutive frames TCX coding mode and 20 to 25 consecutive frames ACELP coding mode.

— The bass-post filter is switched on (bpf_control_info=1) and off (bpf_control_info=0). From time to time this shall toggle frequently every 2-3 frames.

— Every possible value of the acb_index parameter (512 or 64 possibilities depending on the subframe position) is used at least once for the ACELP frames where the bass-post filter is enabled.

— For each pair of channels (except MPEG-H 3DA LFE) the bpf_control_info shall be different for both channels for at least one frame.

— for every channel where FD mode and LPD mode is used, there shall be at least one transition from FD to LPD and at least one transition from LPD to FD, where bpf_control_info is set to 1 and one where bpf_control_info is set to 0 for each LPDmode. Note that the bpf_control_info can only be set in the LPD-channelstream.

7.3.3.3.3 Default behaviour

If this conformance test condition is not active, the default behaviour should be that bpf_control_info is set to 0. Note this is only possible if Fullband LPD is deactivated.

7.3.3.4 Enhanced noise filling test conditions

7.3.3.4.1 General

The enhanced noise filling tool is very versatile. Therefore several conformance test conditions are defined for this tool. Each condition is named with "E-" at the beginning. In case no Enhanced Noise Filling related conformance test condition is active, the syntax elements **noiseFilling** and **enhancedNoiseFilling** shall be 0. If no Enhanced Noise Filling related conformance test condition is active without the Stereo Filling Test condition, **enhancedNoiseFilling** shall be 0.

If **enhancedNoiseFilling** is set to 1, then igf_AllZero shall be 0 in the majority of occurrences if not otherwise prohibited.

7.3.3.4.2 IGF range signalling [E-ran-<x>-<y>]

7.3.3.4.2.1 General

This test condition shall be applied to verify the proper behaviour of the syntax element **igfStartIndex** and **igfStopIndex** as defined in mpegh3daCoreConfig().

7.3.3.4.2.2 Test sequences

The scalefactor bands where ENF is active are signalled by **igfStartIndex** and **igfStopIndex**. Each possible combination of these indices is reflected by two decimal values appended to E-ran, one for **igfStartIndex** (<x>) and one for **igfStopIndex** (<y>), where <x> is in the range of [00, 01, ..., 31] and <y> in the range of [00, 01, ..., 15].

7.3.3.4.2.3 Default behaviour

If this test condition is not active, the default behaviour shall be that **igfStartIndex** equals 21 and **igfStopIndex** equals 15.

7.3.3.4.3 IGF tiling [E-Cti]

7.3.3.4.3.1 General

This test condition shall be applied to verify the proper behaviour of the syntax element **igf_UsePrevTileIdx** and **igfCurrTileIdx** as defined in igf_data().

7.3.3.4.3.2 Test sequences

This test shall contain test conditions according to Table 27. The number of tiles igfNTiles corresponds to the **igfStartIndex** and **igfStopIndex**. Those variables can be changed with the test condition "E-ran".

Table 27 — igf_UsePrevTileIdx and igfCurrTileIdx

Bitstream field	Values
igf_UsePrevTileIdx	0, 1
igfCurrTileIdx[0]	0...3
igfCurrTileIdx[1]	0...3, if igfNTiles > 1
igfCurrTileIdx[2]	0...3, if igfNTiles > 2
igfCurrTileIdx[3]	0...3, if igfNTiles > 3

Additionally, all pairs of **igfCurrTileIdx[]** shall be different at least one time.

7.3.3.4.3.3 Default behaviour

If this test condition is not active, the default behaviour shall be that **igfCurrTileIdx** is equal to 3 in each accessible element.

7.3.3.4.4 IGF whitening [E-Wht]

7.3.3.4.4.1 General

This test condition shall be applied to verify the proper behaviour of the syntax element **igf_UsePrevWhiteningLevel**, **remainingTilesDifferent** and **igf_WhiteningLevel** as defined in igf_data().

7.3.3.4.4.2 Test sequences

In order to use this test condition the syntax element **igfUseWhitening** shall equal 1.

This test shall contain test conditions according to Table 28. The number of tiles igfNTiles corresponds to the **igfStartIndex** and **igfStopIndex**. Those variables can be changed with the test condition "E-ran".

Table 28 — igf_UsePrevWhiteningLevel, remainingTilesDifferent and igf_WhiteningLevel

Bitstream field	Values
igf_UsePrevWhiteningLevel	0, 1
remainingTilesDifferent	0, 1
igf_WhiteningLevel[0]	0...2
igf_WhiteningLevel[1]	0...2, if igfNTiles > 1
igf_WhiteningLevel[2]	0...2, if igfNTiles > 2
igf_WhiteningLevel[3]	0...2, if igfNTiles > 3

Additionally, all pairs of **igf_WhiteningLevel[]** shall be different at least one time.

7.3.3.4.4.3 Default behaviour

If this test condition is not active, the default behaviour shall be that **igfUseWhitening** is equal to 0.

7.3.3.4.5 IGF envelope noise flattening [E-Enf]

7.3.3.4.5.1 General

This test condition shall be applied to verify the proper behaviour of the syntax element **igfApplyTNF** as defined in igf_data().Besides the temporal noise flattening (TNF) tool this test condition will also test the independent noise filling (INF) which does not feature a dedicated payload syntax element but will be activated with the syntax element **igfUseEnf**, which is also mandatory for the temporal noise flattening.

7.3.3.4.5.2 Test sequences

If this test condition is active, the value **igfUseEnf** shall be equal to 1.

Beside that this test shall contain test conditions according to Table 29.

Table 29 — igfApplyTNF

Bitstream field	Values
igfApplyTNF	0, 1

Additionally, **igfApplyTNF** shall toggle at least 10 times in both directions.

7.3.3.4.5.3 Default behaviour

If this test condition is not active, the default behaviour shall be that **igfUseEnf** equals 0.

7.3.3.4.6 IGF After TNS Synth [E-Ats]

7.3.3.4.6.1 General

This test condition shall be applied to verify the proper behaviour of the syntax element **igfAfterTnsSynth** as defined in mpegh3daCoreConfig().

7.3.3.4.6.2 Test sequences

If this test condition is active, the value **igfAfterTnsSynth** shall be equal to 1.

7.3.3.4.6.3 Default behaviour

If this test condition is not active, the default behaviour shall be that **igfAfterTnsSynth** is equal to 0.

7.3.3.4.7 IGF no high resolution [E-Nhr]

7.3.3.4.7.1 General

This test condition shall be applied to verify the proper behaviour of the use of not high resolution.

7.3.3.4.7.2 Test sequences

If this test condition is active, the value **igfUseHighRes** in mpegh3daCoreConfig() shall be equal to 0.

7.3.3.4.7.3 Default behaviour

If this test condition is not active, the default behaviour shall be that **igfUseHighRes** in mpegh3daCoreConfig() is equals to 1.

7.3.3.4.8 IGF no independent tiling [E-Nit]

7.3.3.4.8.1 General

This test condition shall be applied to verify the proper behaviour of not using independent tiling.

7.3.3.4.8.2 Test sequences

To test the behaviour of not independent Tiling, the value **igfIndependentTiling** in mpegh3daChannelPairElementConfig() shall be equal to 0.

This test shall contain values according to Table 30.

Table 30 — Not independent tiling parameters

Bitstream field	Values
igf_ms_mask_present	0, 1, 2, 3
ms_used (related to igf_ms_mask_present ==1)	0, 1
igf_stereo_pred_all	0, 1
cplx_pred_used (in igf_stereo_pred_data())	0, 1
igf_pred_dir	0, 1
igf_delta_code_time	0, 1

7.3.3.4.8.3 Default behaviour

If this test condition is not active, the default behaviour shall be that **igfIndependentTiling** in mpegh3daChannelPairElementConfig() is equal to 1.

7.3.3.4.9 Stereo filling [E-SFi]

7.3.3.4.9.1 General

This test condition shall be applied to verify the proper behaviour of the stereo filling tool.

7.3.3.4.9.2 Test sequences

In order to use this test condition the syntax element **noiseFilling** shall be 1.

This test shall contain test conditions according to Table 31:

Table 31 — Stereo filling

Bitstream field	Values
common_window	0, 1
noise_level	[0]
noise_offset	[1...31]

Additionally the following conditions shall be fulfilled:

— For at least one MPEG-H 3DA CPE, **common_window** shall be always set to 1.

— Mostly if **common_window** is set to 1 (at least 80 %), **noise_level** shall be set to 0 and **noise_offset** unequal to 0 for the second channel (activates stereo filling).

— All valid combinations of **noise_level** equal to 0 and **noise_offset** unequal to 0 shall be triggered at least once.

7.3.3.4.9.3 Default behaviour for stereo filling

If this test condition is not active, the default behaviour should be that **noiseFilling** is set to 0.

7.3.3.5 Channel pair element configuration [cpc-<x1>-<x2>-...]

7.3.3.5.1 General

This test condition shall be applied to verify the proper decoder behaviour for different CPE configurations.

7.3.3.5.2 Test sequences

The name of this test condition contains as many parameters (<x1>,<x2>,...) as ID_USAC_CPE. The order of the parameter shall be in sync with the order of the ID_USAC_CPE in *mpegh3daDecoderConfig()*. All parameters are connected by "-". If the parameter is an integer, the corresponding **shiftIndex1** of the specific ID_USAC_CPE shall be 1 and **shiftChannel1** shall be set to the value of the corresponding parameter. If the parameter is "x" **shiftIndex1** of the specific ID_USAC_CPE shall be set to 0.

7.3.3.5.3 Default behaviour

If this test condition is not active, the default behaviour should be that **shiftIndex1** is set to 0.

7.3.3.6 Tonal component coding [Tcc-<x1>-<x2>-...]

7.3.3.6.1 General

This test condition shall be applied to verify the correct decoder behaviour for the tonal component coding tool.

7.3.3.6.2 Test sequences

For the purpose of conformance testing the variable **tccConfig** is defined as shown in Table 32.