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

**Part 3:
3D audio**

*Technologies de l'information — Codage à haute efficacité et livraison
des médias dans des environnements hétérogènes —*

Partie 3: Audio 3D



Reference number
ISO/IEC 23008-3:2022(E)

Documents
AVA Document Sharing Hub
© ISO/IEC 2022



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2022

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

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

Published in Switzerland

Contents

Page

Foreword.....	xiii
Introduction.....	xiv
1 Scope	1
2 Normative references	1
3 Terms, definitions, symbols, abbreviated terms and mnemonics.....	2
3.1 Terms, definitions, symbols and abbreviated terms	2
3.2 Mnemonics	2
4 Technical overview.....	2
4.1 Decoder block diagram	2
4.2 Overview over the codec building blocks	3
4.3 Efficient combination of decoder processing blocks in the time domain and QMF domain.....	6
4.4 Rule set for determining processing domains	9
4.4.1 Audio core codec processing domain.....	9
4.4.2 Mixing	10
4.4.3 DRC-1 Operation domains (DRC in rendering context).....	10
4.4.4 Audio core codec interface domain to rendering	10
4.4.5 Rendering context.....	10
4.4.6 Post-processing context	11
4.4.7 End-of-chain context.....	11
4.5 Sample rate converter	11
4.6 Decoder delay	11
4.7 Contribution mode of MPEG-H 3D audio.....	12
4.8 MPEG-H 3D audio profiles and levels	12
4.8.1 General	12
4.8.2 Profiles	13
5 MPEG-H 3D audio core decoder.....	27
5.1 Definitions.....	27
5.1.1 Joint stereo.....	27
5.1.2 MPEG surround based stereo (MPS 212)	28
5.2 Syntax.....	28
5.2.1 General	28
5.2.2 Decoder configuration	28
5.2.3 MPEG-H 3D audio core bitstream payloads	51
5.3 Data structure	72
5.3.1 General	72
5.3.2 General configuration data elements.....	72
5.3.3 Loudspeaker configuration data elements	75
5.3.4 Core decoder configuration data elements.....	77
5.3.5 Downmix matrix data elements	81
5.3.6 HOA rendering matrix data elements	84
5.3.7 Signal group information elements	87
5.3.8 Low frequency enhancement (LFE) channel element, mpegh3daLfeElement().....	87
5.3.9 Compatible profile and levels sets.....	88
5.4 Configuration element descriptions	88
5.4.1 General	88
5.4.2 Downmix configuration.....	88
5.4.3 HOA rendering matrix configuration	94

5.5	Tool descriptions	98
5.5.1	General.....	98
5.5.2	Quad channel element	98
5.5.3	Transform splitting.....	100
5.5.4	MPEG surround for mono to stereo upmixing.....	107
5.5.5	Enhanced noise filling	110
5.5.6	Audio pre-roll	134
5.5.7	Fullband LPD	137
5.5.8	Time-domain bandwidth extension.....	148
5.5.9	LPD stereo coding	161
5.5.10	Multichannel coding tool.....	169
5.5.11	Filterbank and block switching	179
5.5.12	Frequency domain prediction.....	180
5.5.13	Long-term postfilter.....	183
5.5.14	Tonal component coding.....	188
5.5.15	Internal channel on MPS212 for low complexity format conversion	198
5.5.16	High resolution envelope processing (HREP) tool	210
5.6	Buffer requirements.....	216
5.6.1	Minimum decoder input buffer	216
5.6.2	Bit reservoir.....	216
5.6.3	Maximum bit rate.....	217
5.7	Stream access point requirements and inter-frame dependency.....	217
6	Dynamic range control and loudness processing	218
6.1	General.....	218
6.2	Description.....	218
6.3	Syntax.....	219
6.3.1	Loudness metadata	219
6.3.2	Dynamic range control metadata.....	219
6.3.3	Data elements.....	220
6.4	Decoding process	222
6.4.1	General.....	222
6.4.2	Dynamic range control.....	224
6.4.3	Usage of downmixId in MPEG-H	224
6.4.4	DRC set selection process	225
6.4.5	DRC-1 for SAOC 3D Content.....	227
6.4.6	DRC-1 for HOA content.....	228
6.4.7	Loudness normalization.....	229
6.4.8	Peak limiter.....	230
6.4.9	Time-synchronization of DRC gains	230
6.4.10	Default parameters	230
7	Object metadata decoding	230
7.1	General.....	230
7.2	Description.....	230
7.3	Syntax.....	231
7.3.1	Object metadata configuration	231
7.3.2	Top level object metadata syntax	232
7.3.3	Subsidiary payloads for efficient object metadata decoding	233
7.3.4	Subsidiary payloads for object metadata decoding with low delay	239
7.3.5	Enhanced object metadata configuration	244
7.4	Data structure	247
7.4.1	Definition of ObjectMetadataConfig() payloads	247
7.4.2	Efficient object metadata decoding	248
7.4.3	Object metadata decoding with low delay	256

7.4.4	Enhanced object metadata	261
8	Object rendering.....	264
8.1	Description.....	264
8.2	Terms and definitions.....	264
8.3	Input data	264
8.4	Processing	265
8.4.1	General remark	265
8.4.2	Imaginary loudspeakers	266
8.4.3	Dividing the loudspeaker setup into a triangle mesh	267
8.4.4	Rendering algorithm	269
9	SAOC 3D.....	273
9.1	Description.....	273
9.2	Definitions.....	273
9.3	Delay and synchronization.....	275
9.4	Syntax.....	275
9.4.1	Payloads for SAOC 3D.....	275
9.4.2	Definition of SAOC 3D payloads.....	279
9.5	SAOC 3D processing	281
9.5.1	Compressed data stream decoding and dequantization of SAOC 3D data	281
9.5.2	Time/frequency transforms.....	281
9.5.3	Signals and parameters	282
9.5.4	SAOC 3D decoding.....	284
9.5.5	Dual mode	289
10	Generic loudspeaker rendering/format conversion	289
10.1	Description.....	289
10.2	Definitions.....	291
10.2.1	General remarks	291
10.2.2	Variable definitions	291
10.3	Processing	291
10.3.1	Application of transmitted downmix matrices.....	291
10.3.2	Application of transmitted equalizer settings	296
10.3.3	Downmix processing involving multiple channel groups.....	296
10.3.4	Initialization of the format converter	297
10.3.5	Audio signal processing.....	313
11	Immersive loudspeaker rendering/format conversion	319
11.1	Description.....	319
11.2	Syntax.....	321
11.3	Definitions.....	321
11.3.1	General remarks	321
11.3.2	Variable definitions	322
11.4	Processing	323
11.4.1	Initialization of the format converter	323
11.4.2	Audio signal processing.....	366
12	Higher order ambisonics (HOA).....	373
12.1	Technical overview	373
12.1.1	Block diagram	373
12.1.2	Overview of the decoder tools	374
12.2	Syntax.....	376
12.2.1	Configuration of HOA elements	376
12.2.2	Payloads of HOA elements.....	380
12.3	Data structure	393
12.3.1	Definitions of HOA Config	393

12.3.2	Syntax of getSubbandBandwidths()	397
12.3.3	Definitions of HOA payload	398
12.4	HOA tool description	406
12.4.1	HOA frame converter	406
12.4.2	Spatial HOA decoding	423
12.4.3	HOA renderer	452
12.4.4	Layered coding for HOA	461
13	Binaural renderer	464
13.1	General	464
13.2	Frequency-domain binaural renderer	464
13.2.1	General	464
13.2.2	Definitions	466
13.2.3	Parameterization of binaural room impulse responses	470
13.2.4	Frequency-domain binaural processing	482
13.3	Time-domain binaural renderer	489
13.3.1	General	489
13.3.2	Definitions	490
13.3.3	Parameterization of binaural room impulse responses	492
13.3.4	Time-domain binaural processing	497
14	MPEG-H 3D audio stream (MHAS)	498
14.1	Overview	498
14.2	Syntax	498
14.2.1	Main MHAS syntax elements	498
14.2.2	Subsidiary MHAS syntax elements	500
14.3	Semantics	501
14.3.1	mpeghAudioStreamPacket()	501
14.3.2	MHASPacketPayload()	502
14.3.3	Subsidiary MHAS packets	503
14.4	Description of MHASPacketTypes	503
14.4.1	PACTYP_FILLDATA	503
14.4.2	PACTYP_MPEGH3DACFG	504
14.4.3	PACTYP_MPEGH3DAFRAME	504
14.4.4	PACTYP_SYNC	504
14.4.5	PACTYP_SYNCGAP	504
14.4.6	PACTYP_MARKER	504
14.4.7	PACTYP_CRC16 and PACTYP_CRC32	505
14.4.8	PACTYP_DESCRIPTOR	505
14.4.9	PACTYP_USERINTERACTION	506
14.4.10	PACTYP_LOUDNESS_DRC	506
14.4.11	PACTYP_BUFFERINFO	506
14.4.12	PACTYP_GLOBAL_CRC16 and PACTYP_GLOBAL_CRC32	506
14.4.13	PACTYP_AUDIOTRUNCATION	507
14.4.14	PACTYP_AUDIOSCENEINFO	508
14.4.15	PACTYP_EARCON	508
14.4.16	PACTYP_PCMCONFIG	508
14.4.17	PACTYP_PCMDATA	508
14.4.18	PACTYP_LOUDNESS	509
14.4.19	MHASPacketType specific requirements for MHASPacketLabel	509
14.5	Application examples	510
14.5.1	Light-weighted broadcast	510
14.5.2	MPEG-2 transport stream	510
14.5.3	CRC error detection	511
14.5.4	Audio sample truncation	511

14.6	Multi-stream delivery and interface	512
14.7	Carriage of generic data	514
14.7.1	Syntax.....	515
14.7.2	Semantics.....	515
14.7.3	Processing at the MPEG-H 3D audio decoder	516
15	Metadata audio elements (MAE)	516
15.1	General	516
15.2	Syntax.....	518
15.3	Semantics.....	527
15.4	Definition of mae_metaDataElementIDs.....	540
15.5	Loudness compensation after gain interactivity.....	541
16	Loudspeaker distance compensation.....	543
17	Interfaces to the MPEG-H 3D audio decoder.....	544
17.1	General	544
17.2	Interface for local setup information	544
17.2.1	General	544
17.2.2	WIRE output.....	544
17.2.3	Syntax for local setup information	545
17.2.4	Semantics for local setup information	545
17.3	Interface for local loudspeaker setup and rendering.....	546
17.3.1	General	546
17.3.2	Syntax for local loudspeaker signalling.....	546
17.3.3	Semantics for local loudspeaker signalling.....	547
17.4	Interface for binaural room impulse responses (BRIRs).....	548
17.4.1	General	548
17.4.2	Syntax of binaural renderer interface	548
17.4.3	Semantics.....	553
17.5	Interface for local screen size information	557
17.5.1	General	557
17.5.2	Syntax.....	557
17.5.3	Semantics.....	557
17.6	Interface for signaling of local zoom area.....	558
17.6.1	General	558
17.6.2	Syntax.....	558
17.6.3	Semantics.....	559
17.7	Interface for user interaction.....	559
17.7.1	General	559
17.7.2	Definition of user interaction categories.....	559
17.7.3	Definition of an interface for user interaction.....	560
17.7.4	Syntax of interaction interface	561
17.7.5	Semantics of interaction interface	562
17.8	Interface for loudness normalization and dynamic range control (DRC).....	564
17.9	Interface for scene displacement data.....	565
17.9.1	General	565
17.9.2	Definition of an interface for scene-displacement data	565
17.9.3	Syntax of the scene displacement interface	566
17.9.4	Semantics of the scene displacement interface	566
17.10	Interfaces for channel-based, object-based, and HOA metadata and audio data	567
17.10.1	General.....	567
17.10.2	Expectations on external renderers.....	567
17.10.3	Object-based metadata and audio data (object output interface)	567
17.10.4	Channel-based metadata and audio data	575
17.10.5	HOA metadata and audio data.....	580

17.10.6	Audio PCM data	584
17.11	Interface for positional scene displacement data	585
17.11.1	General	585
17.11.2	Syntax of the positional scene displacement interface	585
17.11.3	Semantics of the positional scene displacement interface	585
17.11.4	Processing	586
18	Application and processing of local setup information and interaction data and scene displacement data	586
18.1	Element metadata preprocessing	586
18.2	Interactivity limitations and restrictions	592
18.2.1	General information	592
18.2.2	WIRE interactivity	592
18.2.3	Position interactivity	593
18.2.4	Screen-related element remapping and object remapping for zooming	593
18.2.5	Closest loudspeaker layout	594
18.3	Screen-related element remapping	594
18.4	Screen-related adaptation and zooming for higher order ambisonics (HOA)	597
18.5	Object remapping for zooming	599
18.6	Determination of the closest loudspeaker	601
18.7	Determination of a list of loudspeakers for conditioned closest loudspeaker playback	601
18.8	Processing of scene displacement angles for channels and objects (CO)	603
18.9	Processing of scene displacement angles for scene-based content (HOA)	605
18.10	Determination of a reduced reproduction layout based on excluded sectors	607
18.11	Diffuseness rendering	608
19	MPEG-H 3D audio profile definition	610
20	Carriage of MPEG-H 3D audio in ISO base media file format	610
20.1	General	610
20.2	Random access and stream access	611
20.3	Overview of new box structures	611
20.4	MHA decoder configuration record	611
20.4.1	Definition	611
20.4.2	Syntax	612
20.4.3	Semantics	612
20.5	MPEG-H audio sample entry	612
20.5.1	Definition	612
20.5.2	Syntax	613
20.5.3	Semantics	613
20.6	MPEG-H audio MHAS sample entry	614
20.6.1	Definition	614
20.6.2	Syntax	615
20.7	MHA dynamic range control and loudness	615
20.7.1	Definition	615
20.7.2	Syntax	615
20.7.3	Semantics	617
20.8	MHA multi-stream signalling	617
20.8.1	Definition	617
20.8.2	Syntax	618
20.8.3	Semantics	618
20.9	Audio scene information	619
20.9.1	MHA group definition	619
20.9.2	MHA switch group definition	620
20.9.3	MHA group preset definition	621
20.9.4	MHA group description text label	623

20.9.5	MHA scene information	625
20.10	Track references	626
20.11	MPEG-H Audio profile and level compatibility sets	626
20.11.1	Definition	626
20.11.2	Syntax	626
20.11.3	Semantics	626
21	Sub-parameters for the MIME type 'Codecs' parameter	627
21.1	General	627
21.2	'Codecs' parameter for MPEG-H 3D audio	627
22	Timing considerations and decoder behaviour	627
23	Multi-stream handling	627
23.1	Restrictions on extension payloads	627
24	Low complexity generic loudspeaker rendering/format conversion	629
24.1	Description	629
24.2	Definitions	630
24.2.1	General remarks	630
24.2.2	Variable definitions	621
24.3	Processing	631
24.3.1	Application of transmitted downmix matrices	631
24.3.2	Application of transmitted equalizer settings	636
24.3.3	Downmix processing involving multiple channel groups	636
24.3.4	Initialization of the format converter	637
24.3.5	Audio signal processing	654
25	Low complexity immersive loudspeaker rendering/format conversion	656
25.1	Description	656
25.2	Syntax	657
25.3	Definitions	658
25.3.1	General remarks	658
25.3.2	Variable definitions	658
25.4	Processing	659
25.4.1	Initialization of the format converter	659
25.4.2	Audio signal processing	703
26	MPEG surround	706
26.1	Technical overview	706
26.2	Syntax and data structure	707
26.3	Tool description	707
27	Production metadata decoding	707
27.1	General	707
27.1.1	Object distance coding	707
27.1.2	Direct headphone signalling	707
27.1.3	Reference distance coding	708
27.2	Syntax	709
27.2.1	Production metadata configuration	709
27.2.2	Production metadata frame	710
27.3	Semantics	711
27.3.1	Production metadata configuration	711
27.4	Decoding process	713
28	Earcon metadata	713
28.1	General	713
28.2	Syntax	713

28.3	Semantics	715
28.4	Decoding process	719
Annex A (normative) Tables for arithmetic decoding of IGF information		721
A.1	cf_se01[27]	721
A.2	cf_se10[27]	721
A.3	cf_se02[7][27]	721
A.4	short cf_se20[7][27]	721
A.5	short cf_se11[7][7][27]	722
A.6	cf_off_se01	723
A.7	cf_off_se10	723
A.8	cf_off_se02[7]	723
A.9	short cf_off_se20[7]	724
A.10	cf_off_se11[7][7]	724
Annex B (normative) SAOC 3D Decorrelator pre-mixing matrices		725
B.1	Premixing matrix for 22.2 output configuration	725
B.2	Algorithm for generating pre-mixing matrices	727
B.2.1	Input to the algorithm and representations	727
B.2.2	Algorithm steps	727
Annex C (informative) Encoder tools		732
C.1	General overview	732
C.1.1	Encoder block diagram	732
C.1.2	Overview of the encoder and decoder building blocks	732
C.2	Core encoder tools	733
C.2.1	Quad channel element	733
C.2.2	Transform splitting	734
C.2.3	Calculation of residual signal for MPEG surround with hybrid residual coding	735
C.2.4	Enhanced noise filling	735
C.3	Object metadata encoding	736
C.3.1	Pre-processing of the object metadata	736
C.3.2	Efficient object metadata encoding	737
C.3.3	Object metadata encoding with low delay	737
C.3.4	Spatially skipping objects	738
C.4	SAOC 3D encoder	738
C.4.1	Overview	738
C.4.2	Calculation of the SAOC 3D parameters	738
C.4.3	Time/frequency transform	739
C.4.4	Framing	739
C.4.5	Parameter quantization and coding	739
C.5	HOA encoder	739
C.5.1	Specification of value ranges for HOA coefficients	739
C.5.2	Encoder block diagram	742
C.5.3	Spatial HOA encoding	743
C.6	MPEG surround encoder tool	779
Annex D (normative) Peak limiter for unguided clipping prevention		781
Annex E (normative) Compact template downmix matrices		782
Annex F (normative) HOA tables		783
F.1	HOA format description	783
F.1.1	Spherical coordinate system for HOA	783
F.1.2	General	783
F.1.3	Definition of real valued spherical harmonics	785
F.1.4	Definition of the HOA signal matrix	785

F.1.5	Definition of the mode matrix.....	785
F.2	Uniformly distributed positions $\Omega q(N = 1) = \theta qN = 1, \phi qN = 1, 1 \leq j \leq O = 4$	786
F.3	Uniformly distributed positions $\Omega q(N = 2) = \theta qN = 2, \phi qN = 2, 1 \leq j \leq O = 9$	786
F.4	Uniformly distributed positions $\Omega q(N = 3) = \theta qN = 3, \phi qN = 3, 1 \leq j \leq O = 16$	786
F.5	Uniformly distributed positions $\Omega q(N = 4) = \theta qN = 4, \phi qN = 4, 1 \leq j \leq O = 25$	787
F.6	32 Uniformly distributed positions in spherical coordinates.....	787
F.7	Uniformly distributed positions $\Omega q(N = 5) = \theta qN = 5, \phi qN = 5, 1 \leq j \leq O = 36$	788
F.8	Uniformly distributed positions $\Omega q(N = 6) = \theta qN = 6, \phi qN = 6, 1 \leq j \leq O = 49$	789
F.9	Uniformly distributed positions $\Omega q(N = 29) = \theta qN = 29, \phi qN = 29, 1 \leq j \leq O = 900$	790
F.10	Uniformly distributed positions $\Omega q(N = 17) = \theta qN = 17, \phi qN = 17, 1 \leq j \leq O = 324$	797
F.11	Uniformly distributed positions $\Omega q(N = 15) = \theta qN = 15, \phi qN = 15, 1 \leq j \leq O = 256$	800
F.12	Table of loudspeaker directions	802
F.13	Table of 64 horizontal-only directions	803
F.14	Table of 256x8 weighting values, WeightValCdbk	804
F.15	Huffman tables for HuffTabIndex=6.....	810
F.16	Huffman tables for HuffTabIndex=7.....	810
F.17	Huffman tables for HuffTabIndex=8.....	810
F.18	Huffman tables for HuffTabIndex=9.....	810
F.19	Huffman tables for HuffTabIndex=10	811
F.20	Huffman tables for HuffTabIndex=11	811
F.21	Huffman tables for HuffTabIndex=12	811
F.22	Huffman tables for HuffTabIndex=13	812
F.23	Huffman tables for HuffTabIndex=14	812
F.24	Huffman tables for HuffTabIndex=15	812
F.25	HOA Spherical grid for DRC DSHT for order N=1	813
F.26	Spherical grid for DRC DSHT for order N=2	813
F.27	Spherical grid for DRC DSHT for order N=3	813
F.28	Spherical grid for DRC DSHT for order N=4	814
F.29	Spherical grid for DRC DSHT for order N=5	814
F.30	Spherical grid for DRC DSHT for order N=6	815
F.31	Spherical grid for DRC DSHT for order N=7	816
F.32	Spherical grid for DRC DSHT for order N=8	817
F.33	Spherical grid for DRC DSHT for order N=9	818
F.34	Huffman table for decoding HuffmanMagDiffNoSbr	820
F.35	Huffman table for decoding HuffmanMagDiffSbr	820
F.36	Huffman table for decoding DecTableAngleDiff.....	821
F.37	Huffman table for decoding ParHuffmanMagDiffNoSbr	821
F.38	Huffman table for decoding ParHuffmanMagDiffSbr	822
F.39	Huffman table for decoding ParDecTableAngleDiff.....	822
F.40	Table for ParDecorrSigsSelectionTableIdx referring to NumOfDecorrSigsPerParSubbandTable and ParSelectedDecorrSigsIdxMatrixTable	823
F.41	Table for ParDecorrSigsSelectionTableIdx referring to ParPermlIdxVectorTable.....	824
F.42	Table for ParDecorrSigsSelectionTableIdx referring to NumOfDecorrSigsPerFirstOrderParSubbandTable and ParFirstOrderSelectedDecorrSigsIdxMatrixTable	824
F.43	Table for ParDecorrSigsSelectionTableIdx referring to ParFirstOrderPermlIdxVectorTable.....	824
Annex G	(informative) Low complexity HOA rendering.....	825
G.1	General	825
G.2	Detailed description	827
G.2.1	Realization of combined HOA synthesis and rendering.....	828
G.2.2	Further remarks	837

Annex H (informative) Information on delay and complexity of time-domain	
binauralization	839
H.1 Complexity and latency.....	839
H.1.1 Algorithm description.....	839
H.1.2 Complexity.....	840
H.1.3 Latency.....	840
H.2 Experimental results	841
H.3 Alternative low-delay implementations.....	842
Annex I (informative) Determination of a rotation matrix for processing of scene	
displacement data	844
Annex J (informative) Decorrelation filtering for ‘diffuseness’ processing.....	846
Annex K (informative) Distance and depth spread rendering	847
Annex L (informative) HREP encoder description.....	849
L.1 Computation of the gains and the beta factor.....	849
L.2 Computation of the LP part and the HP part.....	850
L.3 Computation of the output signal	850
L.4 Encoding of gains using arithmetic coding.....	851
Annex M (informative) Screen-related adaptation of HOA content in complexity	
constrained implementations.....	853
Annex N (normative) Retaining original file length with MPEG-H 3D audio.....	854
N.1 General.....	854
N.2 Avoiding leading zero samples	854
N.3 Avoiding trailing zero samples	855
Annex O (normative) Codebook tables used to de-quantize high band time domain	
bandwidth extension parameters.....	856
0.1 Codebook table for theLSFCB1_7b	856
0.2 Codebook table for theLSFCB2_7b	858
0.3 Codebook table for theExcFilterCB1_7b	860
0.4 Codebook table for theExcFilterCB2_4b	863
0.5 Codebook table for SHBCB_SubGain5bit	863
0.6 Codebook table for SHBCB_GainFrame5bit.....	863
0.7 Codebook table for win_flatten	863
0.8 Codebook table for RVEC.....	864
Annex P (informative) Implementation and usage guidelines for signalling of profile and	
level compatibility sets	865
P.1 Baseline profile signalling.....	865
Bibliography	867

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 <https://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 Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This third edition cancels and replaces the second edition (ISO/IEC 23008-3:2019), which has been technically revised. It also incorporates Amendments ISO/IEC 23008-3:2019/Amd 1:2019 and ISO/IEC 23008-3:2019/Amd 2:2020.

The main changes are as follows:

- specification of audio metadata enhancements;
- carriage of Earcon metadata and PCM data in MHAS packets;
- specification of baseline profile;
- signaling of compatibility to multiple profiles and levels.

A list of all parts in the ISO/IEC 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

3D sound systems are able to realize a significantly enhanced sound experience relative to current widespread 5.1 channel audio programs and playback systems. These systems demand high quality audio coding and error-free transmission in order to keep the timbre, sound localization and sound envelopment of the original audio program. Presentation over headphones with suitable spatialization are also considered.

This document provides means for all scenarios where there is a need to compress a multi-channel audio program (e.g. 22.2 channel program) and to render it to the native target number of loudspeakers. In order to reach a wide market, a 3D audio program is able to be downmixed to a lower hierarchy of loudspeakers, for example 10.1 or 8.1 channels. In addition, all scenarios support a level of random access to facilitate broadcast break-in, and “trick modes” such as fast forward when playing from stored media.

This document focuses on applications such as audio for home theatres where the audio presentation is immersive, involving many loudspeakers (e.g. from 10 to more than 20) surrounding the listener and placed below, at and above ear-level. Moreover, applications as personal TV, TV for smartphones and multi-channel audio-only programs are envisioned. These require that 3D audio encoding/decoding systems are able to operate at low bitrates appropriate for efficient transmission over a cellular channel. At the same time, the sense of envelopment and accurate sonic localization even for systems having a tablet-sized visual displays with loudspeakers built into the device and headphone listening are maintained. The definition of the main profile, its associated bitstream syntax, semantics, and decoding process description was provided in the first edition of this document (ISO/IEC 23008-3:2015).

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents or patents.iec.ch.

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

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

Part 3: 3D audio

1 Scope

This document specifies technology that supports the efficient transmission of immersive audio signals and flexible rendering for the playback of immersive audio in a wide variety of listening scenarios. These include home theatre setups with 3D loudspeaker configurations, 22.2 loudspeaker systems, automotive entertainment systems and playback over headphones connected to a tablet or smartphone.

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 13818-1, *Information technology — Generic coding of moving pictures and associated audio information — Part 1: Systems*

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

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

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

ISO/IEC 23003-1:2007, *Information technology — MPEG audio technologies — Part 1: MPEG Surround*

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

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*

IETF RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace*

3 Terms, definitions, symbols, abbreviated terms and mnemonics

3.1 Terms, definitions, symbols and abbreviated terms

For the purposes of this document, the terms, definitions, symbols and abbreviated terms given in ISO/IEC 14496-3 and ISO/IEC 23003-3 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/>

3.2 Mnemonics

The following mnemonics are defined to describe the different data types used in the coded bitstream payload.

bslbf	Bit string, left bit first, where “left” is the order in which bit strings are written in ISO/IEC 14496 (all parts). Bit strings are written as a string of 1s and 0s within single quote marks, for example '1000 0001'. Blanks within a bit string are for ease of reading and have no significance.
uimsbf	Unsigned integer, most significant bit first.
vlclbf	Variable length code, left bit first, where “left” refers to the order in which the variable length codes are written.
tcimsbf	Two’s complement integer, most significant (sign) bit first.

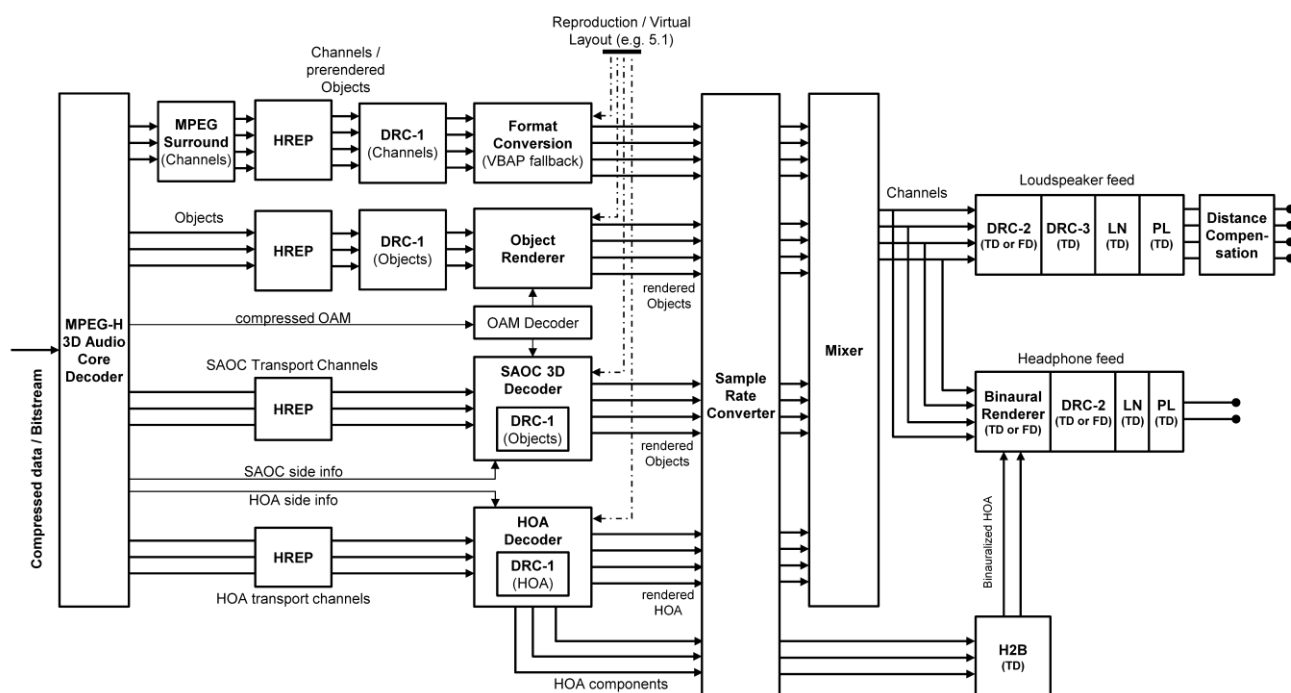
4 Technical overview

4.1 Decoder block diagram

The 3D audio codec system consists of an MPEG-H 3D audio core codec for coding of channel, object and higher order ambisonics (HOA) signals. The core codec is based on the MPEG-D USAC codec. To increase the efficiency for coding a large amount of objects, MPEG SAOC technology has been adopted. Several types of renderers perform the tasks of rendering objects to channels, rendering channels to a different loudspeaker setup, rendering HOA signals to the loudspeaker setup or rendering virtual loudspeaker channels or HOA components to headphones.

When object signals are explicitly transmitted or parametrically encoded using SAOC, the corresponding object metadata information is compressed and multiplexed into the 3D audio bitstream.

Figure 1 shows the different algorithmic blocks of the 3D audio system.



Key

DRC	dynamic range control	HREP	high resolution envelope processing
SAOC	spatial audio object coding	OAM	object audio metadata
HOA	higher order ambisonics	H2B	HOA to binaural
LN	loudness normalization	PL	peak limiter

Figure 1 — 3D audio decoder

4.2 Overview over the codec building blocks

The MPEG-H 3D audio core codec for loudspeaker-channel signals, discrete object signals, object downmix signals and pre-rendered signals is based on MPEG-D USAC technology. It handles the coding of the multitude of signals by creating channel- and object-mapping information based on the geometric and semantic information of the input's channel and object assignment. This mapping information describes how input channels and objects are mapped to channel elements (CPEs, SCEs, LFEs) and the corresponding information is transmitted to the decoder.

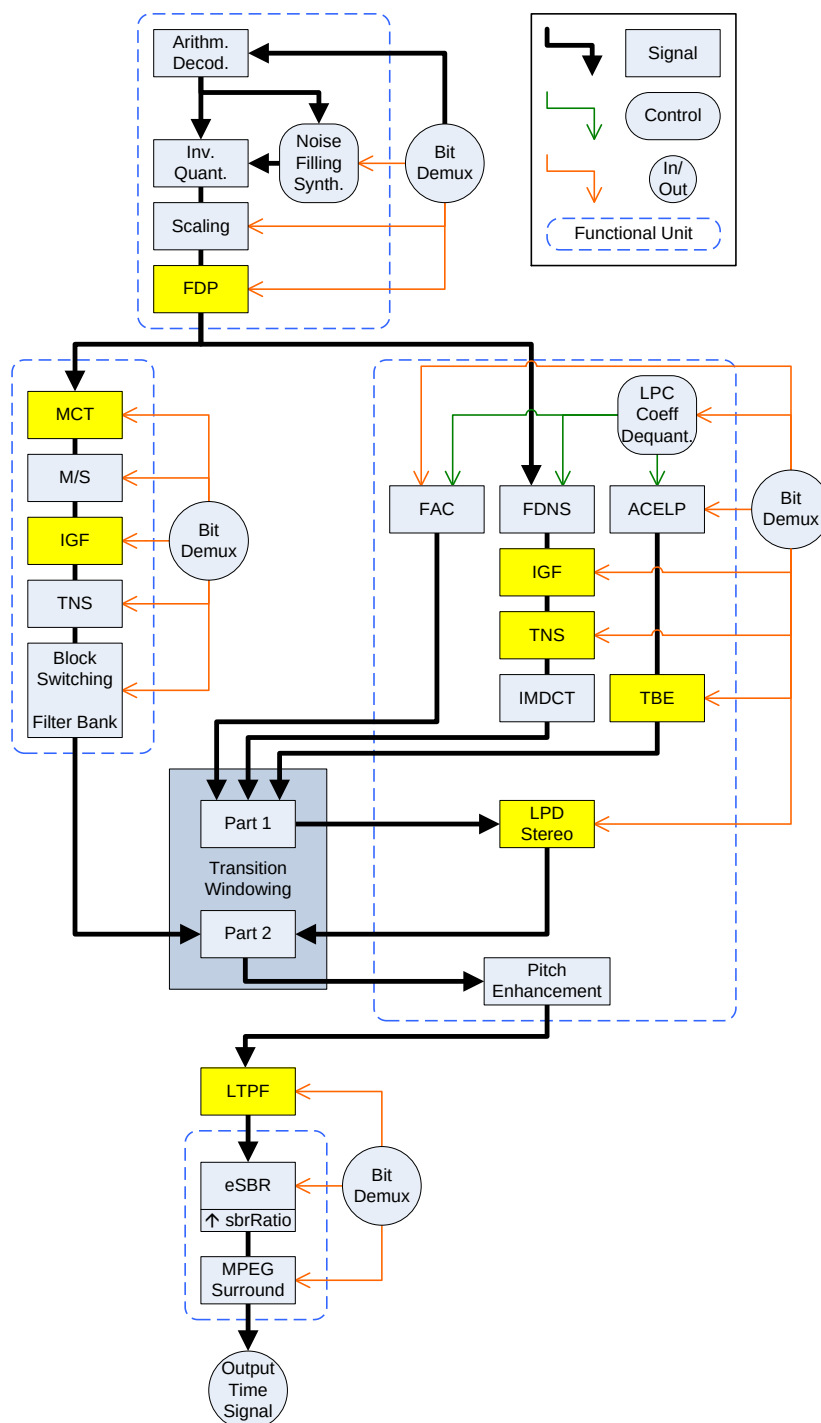


Figure 2 — Simplified typical MPEG-H core decoder configuration

Figure 2 shows a simplified block diagram of the typical MPEG-H core decoder building blocks. The major differences compared to MPEG-D USAC technology are highlighted in yellow. The highlighted tools are described in detail in Clause 5.

The coding of objects is possible in different ways, depending on the rate/distortion requirements and the interactivity requirements for the renderer. The following object coding variants are possible.

- Prerendered objects: Object signals are pre-rendered and mixed to multi-channel or HOA signals before encoding, as appropriate. The subsequent coding chain then operates on multi-channel or HOA signals.
- Discrete object waveforms: Objects are supplied as monophonic waveforms to the encoder. The encoder uses single channel elements SCEs to transmit the objects in addition to the channel signals. The decoded objects are rendered and mixed at the receiver side. Compressed object metadata information is transmitted to the receiver/renderer alongside.
- Parametric object waveforms: Object properties and their relation to each other are described by means of SAOC parameters. The downmix of the object signals is coded with the MPEG-H 3D audio core codec. The parametric information is transmitted alongside. The number of downmix channels is chosen depending on the number of objects and the overall data rate. Compressed object metadata information is transmitted to the SAOC renderer.

The SAOC encoder and decoder for object signals are based on MPEG SAOC technology. The system is capable of recreating, modifying and rendering a number of audio objects based on a smaller number of transmitted channels and additional parametric data (OLDs, IOCs, DMGs).

The SAOC decoder reconstructs the object/channel signals from the decoded SAOC transport channels and parametric information, and generates the output audio scene based on the reproduction layout, the decompressed object metadata information and optionally on the user interaction information.

The object metadata codec efficiently codes the associated metadata that specifies the geometrical position and volume of each object in 3D space by quantization of the object properties in time and space. The compressed object metadata is transmitted to the receiver as side information.

The object renderer utilizes the compressed object metadata to generate object waveforms according to the given reproduction format. Each object is rendered to certain output channels according to its metadata. The output of this block results from the sum of the partial results.

The loudspeaker renderer converts between the transmitted channel configuration and the desired reproduction format. It is thus called 'format converter'. In case of conversions to lower numbers of output channels it creates downmixes. The system automatically generates optimized downmix matrices for the given combination of input and output formats and applies these matrices in a downmix process. The format converter allows for standard loudspeaker configurations as well as for random configurations with non-standard loudspeaker positions.

The higher order ambisonics (HOA) decoder/renderer reconstructs the HOA coefficient signals based on the HOA transport channels decoded by the 3D audio core decoder and the HOA specific side information. The coding principle is based on a separate transmission of so-called predominant sounds and ambient sound scene components. Subsequently the HOA renderer generates the loudspeaker channel feeds based on the reproduction layout.

If two or more groups of channel based content, discrete/parametric objects or HOA based content are decoded, the corresponding waveforms are delay-aligned and sample-wise added by the mixer before providing the resulting waveforms (or before feeding them to a postprocessor module such as the binaural renderer, DRC-2, DRC-3, the peak limiter PL, or the loudspeaker distance compensation).

The sample rate converter block converts between the core decoder sampling rate and the decoder output sampling rate. It enables a constant output sampling rate in case of varying core coding sampling rates and allows for resampling of audio scenes, which may be coded in multiple sub-streams with different core sampling rates, to a common output sampling rate.

Note that in case DRC-2/3 gains need to be applied in the post-processing context (see Figure 2), the DRC-2/3 gains should be resampled similar to the audio samples.

The MPEG surround decoder takes the downmix signals coming from the MPEG-H 3D audio decoder and performs the guided MPEG surround upmix using the MPEG surround side information to reproduce the multichannel signal for the transmitted loudspeaker layout.

The binaural renderer module produces a binaural downmix of the multichannel audio material, such that each input channel is represented by a virtual sound source. The processing is conducted frame-wise in the QMF domain or in the time domain. The binauralization is based on measured binaural room impulse responses (BRIRs).

Virtual layout information fed from an application shall be consistent with the corresponding BRIR set provided (expressed with MeasurementSetup, see Table 253), in the sense that the set of positions corresponding to the virtual layout is a subset of the set of positions corresponding to the BRIRs.

4.3 Efficient combination of decoder processing blocks in the time domain and QMF domain

There are numerous processing blocks in the MPEG-H 3D audio decoding, rendering and processing framework. In general these blocks can be classified into different classes:

- 1) Block operates in the time domain (**TD**)
- 2) Block operates in frequency domain (**FD**) – also called QMF domain
- 3) Block is “**neutral**” and can operate in FD or TD.
Signal-processing-wise the same operation is carried out in either FD or TD. Operation in a different domain may cause no difference or only a small but perceptually negligible difference to the output signal.

Table 1 lists all blocks with their corresponding processing domain. For the sake of clarification, the functional blocks are grouped into semantically differentiable contexts, which follow the MPEG-H 3D audio signal processing as shown in Figure 3.

Table 1 — MPEG-H 3D audio functional blocks and internal processing domain

Processing context	Functional block	Processing domain	Delay samples [1/f _{s,core}] or [1/f _{s,out}]	Contribution to maximum delay high profile samples [1/f _{s,out}]	Contribution to maximum delay low complexity profile samples [1/f _{s,out}]
Audio core		TD , Core frame length = 1024	0		
	SBR / MPS212	FD , Core frame length = 2048 or 4096	0		
	SBR/MPS212, stereoConfigIndex==3	FD , Core frame length = 2048 or 4096	384	384 × RSR_{max}	
	MPEG-H 3D audio core coder	FD or TD			
	MPEG surround	FD	640	640 × RSR_{max}	
	HREP	TD , Core frame length = 1024	64		
	HREP, QMF-synthesis and QMF-analysis pair and alignment to 64 sample grid	FD TD FD	64 + 257 + 320 + 63	(64 + 257 + 320 + 63) × RSR_{max}	
	QMF-analysis	TD, FD	320	320 × RSR_{max}	
	QMF-synthesis	FD, TD	257	257 × RSR_{max}	
	Hybrid filter/SBR	FD	384	384 × RSR_{max}	
Rendering	DRC-1	if multiband: TD (STFT) or FD else: neutral	0		
	Format converter, core frame length = 1024	FD	3072	3072 × RSR_{max}	
	Format converter, core frame length = 2048 or 4096	FD	2048		
	STFT format converter	TD (STFT)	256		256 × RSR_{max}
	Object renderer	Neutral	0		
	SAOC 3D decoder, bsDoubleFrameLengthFlag=0	FD	0		
	SAOC 3D decoder, bsDoubleFrameLengthFlag=1 Core frame length = 1024	FD	1024		

Processing context	Functional block	Processing domain	Delay samples [1/f _{s,core}] or [1/f _{s,out}]	Contribution to maximum delay high profile samples [1/f _{s,out}]	Contribution to maximum delay low complexity profile samples [1/f _{s,out}]
	SAOC 3D decoder, bsDoubleFrameLengthFlag=1 Core frame length = 2048	FD	2048		
	HOA decoder	TD, (FD)^c	0 577		
	HOA decoder with multiband DRC-1	TD, FD or TD (STFT), TD	577 256		
	Sample rate converter	FD, TD	256	256	256
Mixing	Mixer	FD, TD	0		
Maximum accumulated delay to mixer:				5761 × RSR_{max} + 256	256 × RSR_{max} + 256
Post-processing	QMF-analysis	TD, FD	320	320	320
	QMF-synthesis	FD, TD	257	257	257
	DRC-2	if multiband: FD else: neutral	0		
	FD binauralizer	FD	0		
	TD binauralizer	TD	0		
End of chain	DRC-3 (only singleband)	TD	0		
	Loudness normalization	TD	0		
	Peak limiter ^a	TD	240	240	240
	LS distance compensation	TD	0		
Maximum accumulated delay to decoder output ^{a, b}:				5761 × RSR_{max} + 1073 = 18356	256 × RSR_{max} + 1073 = 1841
^a The given values are valid for f _{s,out} = 48 kHz. The peak limiter shall introduce a delay of 5ms, i.e. the limiter delay in samples shall be determined as floor(f _{s,out} · 5/kHz). The maximum accumulated delay to the decoder output has to be adapted accordingly for output sampling rates that differ from f _{s,out} = 48 kHz. ^b The maximum resampling factor RSR _{max} of the sampling rate converter is 3. ^c HOA Decoder: FD in case subband directional prediction or PAR is used (only high profile), low complexity profile only operates in TD.					

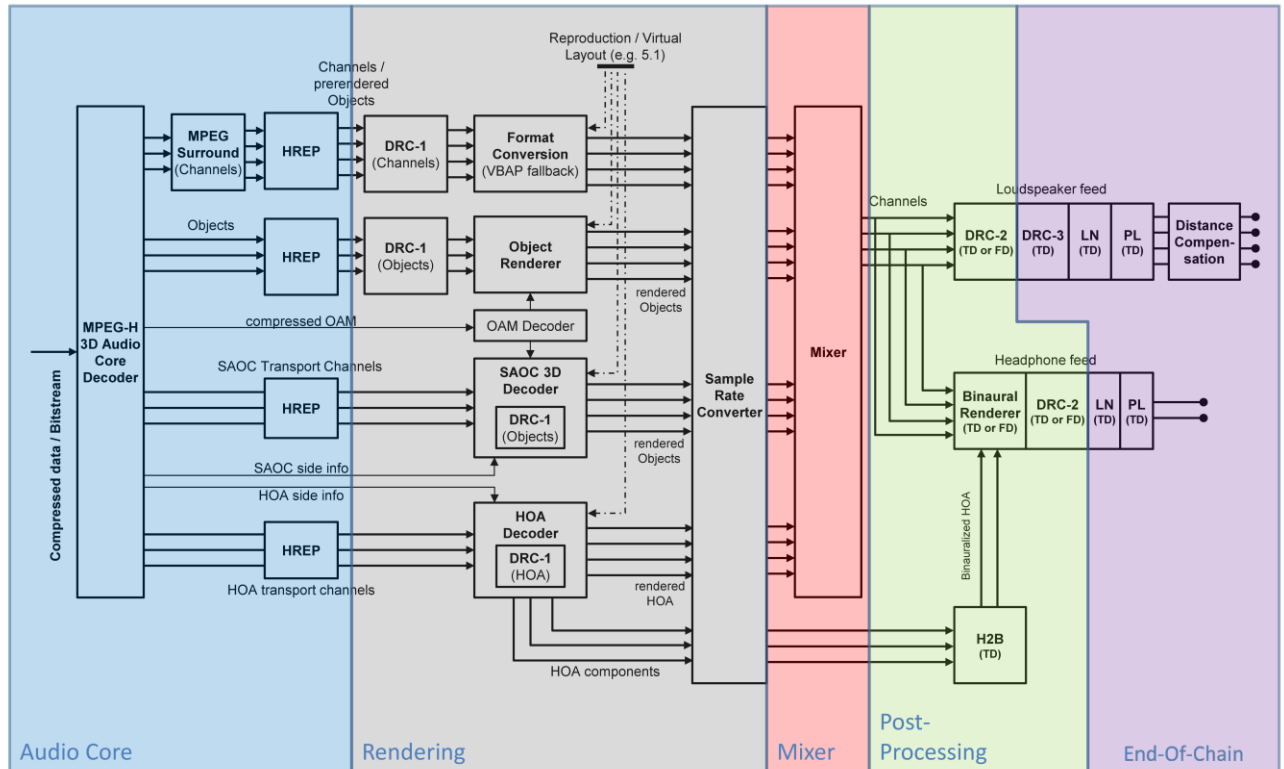


Figure 3 — MPEG-H 3D audio decoder overview with signal processing context

Operating blocks that are directly connected in the signal chain and which operate in the *same* domain can interface in that common domain.

Operating blocks that are directly connected in the signal chain and which operate in *different* domains require a transformation block which turns one signal representation into another, i.e. a QMF analysis (TD to FD) or a QMF synthesis (FD to TD). This transform causes additional delay and hence needs to be taken into account when determining the overall signal processing delay.

The following subclause describes formal generic rules which determine how the various functional blocks can be connected.

4.4 Rule set for determining processing domains

4.4.1 Audio core codec processing domain

The incoming bitstream configuration determines which core decoder tools are active and whether parts of the core codec operate in QMF domain / FD.

- a) If last decoding stage is in QMF domain (SBR and/or MPS212 active) (typical for mid to low bit rate coding)
 - 1) formal audio core codec processing domain is **FD**
- b) If last decoding stage is **not** in QMF domain (neither SBR nor MPS212 active) (typical for high rate coding)
 - 1) formal audio core codec processing domain is **TD**

4.4.2 Mixing

- The mixer may operate in **TD** or in **FD**

4.4.3 DRC-1 Operation domains (DRC in rendering context)

- If multiband DRC-1 data is present in the bitstream for a DRC-1 module,
 - the corresponding DRC-1 processing block should operate in **FD** or **TD (STFT)**

4.4.4 Audio core codec interface domain to rendering

- If the audio core codec processing domain is **TD**
 - If the rendering block requires operation in **FD**
 - apply QMF analysis and output **FD** for this rendering block
 - else
 - output **TD**
- If the audio core codec processing domain is **FD**
 - If the subsequent rendering block requires operation in **TD**
 - apply QMF synthesis and output **TD** for this rendering block
 - else
 - output **FD**

4.4.5 Rendering context

Rendering paths are considered individually:

- If the rendering block operates in **FD** and mixer operates in **TD**
 - apply QMF synthesis after renderer and output **TD**
- If the rendering block operates in **TD** and mixer operates in **FD**
 - apply QMF analysis after renderer and output **FD**
- else
 - keep processing domain

In case the core decoder operates in the time domain (TD), i.e. if the core decoder does not apply QMF domain processing like SBR or MPS-212, the low-complexity STFT domain downmix as specified in clauses 24 and 25 shall be applied for format conversion if format conversion is requested.

4.4.6 Post-processing context

- If mixer operates **TD**
 - If **any** block in post-processing context requires operation in **FD**
 - convert to **FD** immediately after mixer;
 - after the **FD** operation convert to **TD**, i.e. prior to end-of-chain context
 - else
 - stay in **TD**
- If Mixer operates in **FD**
 - If **no** block in post-processing context requires operation in **FD**
 - convert to **TD** prior to DRC-2
 - else
 - convert to **TD** prior to end-of-chain context

4.4.7 End-of-chain context

All blocks shall operate in **TD**.

4.5 Sample rate converter

The sample rate converter performs sample rate conversion between the core coder sampling rate and the output sampling rate of the decoder. The signal delay introduced by the sample rate converter shall not exceed 256 samples, measured in samples at the output sampling rate of the decoder. The sample rate converter shall be able to perform sampling rate conversions with resampling ratios (defined as decoder output sampling rate divided by core decoder output sampling rate) of 3/2, 2, and 3. The sample rate converter block shall further be able to pass through the signal(s) if the core decoder sampling rate matches the decoder output sampling rate.

4.6 Decoder delay

If a constant decoder delay application is signalled in the bitstream (**receiverDelayCompensation**==1, as defined in subclause 5.2.2) then the decoding and rendering delay from the IMDCT output of the core decoder to the mixing block of the MPEG-H 3D audio decoder shall be kept constant by introducing delay lines where required. Similarly, the overall delay from the IMDCT output to the decoder output shall be kept constant by introducing delay lines where required. The constant delay values that shall be fulfilled are determined by the *maximum accumulated delay* numbers in Table 1.

Enforcing a constant delay from the IMDCT output to the mixer implies that all signals at the IMDCT output shall be aligned. Accordingly, the decoder shall compensate for different delays that may occur in the processing of the IMDCT output signals, e.g. different SBR/MPS processing delays for SCEs and CPEs in case of **stereoConfigIndex**==3.

Further, all side information utilized in the rendering blocks as well as for DRC processing shall be sent aligned to the IMDCT output waveforms independent of the value of receiverDelayCompensation. The rendering and DRC side information shall be applied delayed in the rendering/DRC blocks. The side information delays are determined by the delays the waveforms encounter in the processing pipeline from the IMDCT output until reaching the corresponding rendering/DRC blocks. The following kinds of side information shall be aligned to the IMDCT output waveforms and shall be applied delayed as defined above:

- OAM object metadata;
- DRC side information;
- SAOC side information;
- HOA side information.

In case a bitstream contains multiple signal groups with different signalGroupTypes, the bitstream element receiverDelayCompensation shall be set to 1.

4.7 Contribution mode of MPEG-H 3D audio

The contribution mode of MPEG-H 3D audio specifies a generic transport mechanism for audio signals with accompanying metadata and it is designed to be unaware of the signal type and of the content and structure of the associated metadata.

A value of **speakerLayoutType** == 3 as defined in Table 69 in the signalling of the **referenceLayout** in the **mpegh3daConfig()** indicates that MPEG-H 3D audio shall operate in contribution mode. In contribution mode the rendering context shall operate in a pass-through mode, i.e. the format converter shall apply an identity matrix to the signal. contribution mode bitstreams shall have the following additional restrictions:

- bsNumSignalGroups == 0 (one single signal group);
- SignalGroupType == 0 (channel signal group);
- differsFromReferenceLayout[0] == 0 (no audioChannelLayout in the signal group);
- core coder delay lines for compensation of SBR, MPS212, shall be applied;
- Content of extension elements shall not be processed in the decoder but shall be made available to an external framework.

4.8 MPEG-H 3D audio profiles and levels

4.8.1 General

This subclause defines profiles and their levels for MPEG-H 3D audio.

Complexity units are defined to give an approximation of the decoder complexity in terms of processing power required for the decoding process. The approximated processing power is given in processor complexity units (PCU), specified in millions operations per second (MOPS).

4.8.2 Profiles

The following audio profiles are defined:

- 1) The main profile of MPEG-H 3D audio provides a complete set of features for low-bitrate and high-quality coding, and rendering for all playback scenarios.
- 2) The high profile of MPEG-H 3D audio provides a complete set of features for low-bitrate and high-quality coding, and rendering for all playback scenarios. The high profile is a superset of the low-complexity profile.
- 3) The low complexity profile provides features for broadcasting and streaming with reduced decoder complexity.
- 4) The baseline profile is a subset of the low-complexity profile which supports channel and object signals.

Table 2 lists a summary of the location of and normative reference to the definitions of MPEG-H 3D audio profiles.

Table 2 — Summary of the location of and normative reference to the definitions of MPEG-H 3D audio profiles

Tool/Module		Defined in ISO/IEC	Sub-clause	USAC 23003-3	MPEG-H 3D audio High profile	MPEG-H 3D audio Low-complexity profile	MPEG-H 3D audio Baseline profile
block switching		14496-3	4.6.11	X	X	X	X
window shapes	AAC based	14496-3	4.6.11	X	X	X	X
	Additional windows	23003-3	6.2.9.3	X	X	X	X
filter bank	AAC based	14496-3	4.6.11	X	X	X	X
	additional USAC	23003-3	7.9	X	X	X	X
TNS		14496-3	4.6.9	X	X	X	X
intensity		14496-3	4.6.8.2				
coupling		14496-3	4.6.8.3				
perceptual noise synthesis	PNS	14496-3	4.6.13				
	noise filling	23003-3	7.2	X	X	X	X
MS	basic mid/side coding	14496-3	4.6.8.1	X	X	X	X
	MDCT based complex prediction	23003-3	7.7.2	X	X	X	X

Tool/Module		Defined in ISO/IEC	Sub-clause	USAC 23003-3	MPEG-H 3D audio High profile	MPEG-H 3D audio Low-complexity profile	MPEG-H 3D audio Baseline profile
quantization	non-uniform	14496-3	4.6.1	X	X	X	X
	uniform	23003-3	7.1	X	X	X	
entropy coding	Huffman	14496-3	4.6.3				
	context adaptive arithmetic coding	23003-3	7.4	X	X	X	X
SBR	base	14496-3	4.6.18	X	X		
	enhanced	23003-3	7.5	X	X		
parametric stereo extension	parametric stereo	14496-3	8.6.4 / 8.A				
	MPEG surround 2-1-2 (incl. residual coding)	23003-3	6.2.13	X	X		
	quad channel element	23008-3	5.5		X		
ACELP		23003-3	7.14	X	X	X	
frequency domain noise shaping	scale factor based	14496-3	4.6.2	X	X	X	X
	LPC based	23003-3		X	X	X	
intelligent gap filling	IGF for FD	23008-3			X	X	X
improved LPD coding	IGF for TCX and TBE in ACELP	23008-3			X	X	
	LPD stereo	23008-3			X	X	
predictors for FD	frequency-domain prediction and time-domain post-filtering	23008-3			X	X	X
predictors for TCX	frequency-domain prediction and time-domain post-filtering	23008-3			X	X	

Tool/Module		Defined in ISO/IEC	Sub-clause	USAC 23003-3	MPEG-H 3D audio High profile	MPEG-H 3D audio Low-complexity profile	MPEG-H 3D audio Baseline profile
discrete multi-channel coding	MCT	23008-3			X	X	X
format converter	generic downmix	23008-3	10, 24		X	X ^d	X ^d
immersive rendering	immersive rendering within format converter	23008-3	11, 25		X	X ^d	X ^d
static metadata	metadata audio elements (MAE) and audio scene information (ASI)	23008-3	15		X	X	X
	decoder and renderer						
dynamic object metadata	object audio metadata (OAM)	23008-3	7, 8		X	X	X
	decoder and renderer						
MPS	MPEG surround extension	23003-1	10		X		
SAOC-3D	decoder and renderer	23008-3	9		X		
HOA	decoder and renderer	23008-3	12		X	X ^e	
	near field compensation	23008-3			X	X ^a	
	subband directional prediction	23008-3			X		
	parametric ambiance replication (PAR)	23008-3			X		
	phase-based decorrelation	23008-3			X		

Tool/Module		Defined in ISO/IEC	Sub-clause	USAC 23003-3	MPEG-H 3D audio High profile	MPEG-H 3D audio Low-complexity profile	MPEG-H 3D audio Baseline profile
Binaural	FD-binaural, TD-binaural	23008-3	13		X	X ^b	X ^b
	HOA2Binaural H2B	23008-3			X	X ^b	
DRC	DRC-1	23003-4			X	X ^c	X ^c
	DRC-2 (single band)	23003-4			X	X	X
	DRC-2 (multi band)	23003-4					
	DRC-3 (single band)	23003-4			X	X	X
sample rate converter		23008-3			X	X	X
peak limiter	unguided clipping prevention	23008-3 23003-4	D		X	X	X
loudness	loudness metadata and handling	23003-4	6		X	X	X
	loudness compensation	23008-3			X	X	X
MHAS	MPEG-H 3D Audio stream	23008-3	14		X	X	X
	truncation message and CRC packet type, ASI packet type	23008-3			X	X	X
file format	carriage of MPEG-H 3D audio in ISO base media file format	23008-3			f		
interfaces and processing	interfaces and processing for interaction data and local setup info	23008-3	17,18		X	X	X

Tool/Module		Defined in ISO/IEC	Sub-clause	USAC 23003-3	MPEG-H 3D audio High profile	MPEG-H 3D audio Low-complexity profile	MPEG-H 3D audio Baseline profile
carriage of generic data	carriage of generic data for the interaction with system engine	23008-3			X	X	
TCC	tonal component coding	23008-3			X		
IC	internal channel	23008-3			X		
HREP	high resolution envelope processing	23008-3			X		
^a Restrictions apply dependent on the levels. ^b Implementation of binaural rendering is only mandated if headphone reproduction is supported. ^c Multi-band DRC-1 shall be applied in the STFT domain of the TD format converter. ^d The TD format converter downmix shall be applied for downmixing. ^e In order to achieve target complexity for the LC profile at a given level, study Annex G. ^f File format encapsulation is independent of the profile that is used for the bitstream. A profile level indicator is part of the file format specification (see subclause 20.4).							

The baseline profile is a subset of the low-complexity profile. If a decoder implementation supports decoding of low complexity profile level 3 bitstreams and supports the configuration extension `CompatibleProfileLevelSet()`, then the decoder shall support decoding of bitstreams encoded according to the baseline profile level 3. Bitstreams complying to the baseline profile may be signalled using:

- the `mpegh3daProfileLevelIndication` field set to indicate baseline profile as specified in Table 67, or alternatively,
- the `mpegh3daProfileLevelIndication` field set to indicate low complexity profile as specified in Table 67 and the `CompatibleProfileLevelSet` configuration extension for indicating compatibility to baseline profile, as described in Annex P;

Additionally, it is strongly recommended that low complexity profile bitstreams conforming to the baseline profile, are signalled using the profile and level values for `mpegh3daProfileLevelIndication` and `CompatibleSetIndication` given in Table P.1.

4.8.2.1 Levels of the low complexity profile

Table 3 — Levels and their corresponding restrictions for the low complexity profile

Level	Max. sampling rate	Max. no. of core ch. in compressed data stream	Max. no. of decoder processed core ch.	Max. no. of loud speaker output ch.	Example of max. loud speaker configuration	Max. no. of decoded objects	Example of a max. config C+O	Max. HOA order	Example of max. HOA order + O
1	48000	10	5	2	2.0	5	2 ch. + 3 static obj. ^a	2	2 nd order + 3 static obj. ^a
2	48000	18	9	8	7.1	9	6 ch. + 3 static obj. ^a	4	4 th order + 3 static obj. ^a
3	48000	32	16	12	11.1	16	12 ch. + 4 obj.	6	6 th order + 4 obj.
4	48000	56	28	24	22.2	28	24 ch. + 4 obj.	6	6 th order + 4 obj.
5	96000	56	28	24	22.2	28	24 ch. + 4 obj.	6	6 th order + 4 obj.

^a In this context “static objects” are understood as channel-based signals without accompanying OAM data which are not also associated to a channel bed.

- The use of switch groups determines the subset of core channels from the core channels in the bitstream that shall be decoded.
- If the `mae_AudioSceneInfo()` contains switch groups (`mae_numSwitchGroups>0`), then the `elementLengthPresent` flag shall be 1.
- The number of channels of the signalled `referenceLayout` shall not exceed the maximum number of loudspeaker output channels as defined in the levels Table 3.

Table 4 lists the approximated worst case processing power (PCU) of decoder modules and the whole decoder for the different levels of the low complexity profile given in MOPS.

Table 4 — Approximated worst case processing power (PCU) of decoder modules and the whole decoder for the different levels of the low complexity profile given in MOPS

Level	Core LC	Format converter	Object renderer	HOA ^a	Objects only renderer	DRC	Limiter	Binaural ^b	Worst case PCU
1	33	3	0	3	9	6	4	7	58
2	59	10	0	17	16	18	5	19	118
3	106	36	7	36	29	24	6	27	206
4	186	113	7	93	50	30	9	46	392
5	373	226	14	186	50	34	19	92	758

^a The complexity numbers for the HOA spatial decoding and rendering are based on the low complexity combined HOA spatial decoding and rendering described in Annex G.

^b The complexity numbers for binaural processing are calculated on the basis of BRIR filters of 1 second length measured in a BS.1116 compliant room.

4.8.2.2 Restrictions for the low complexity profile and levels

In the low complexity profile the core decoder, format converter, object renderer, HOA renderer and DRC and peak limiter operate in the time domain, MDCT-domain or STFT-domain.

The following restrictions apply for HOA renderer and decoder, according to Table 5.

Table 5 — Restrictions for the HOA spatial decoding and rendering according to the level of the low complexity profile

Restriction applies to	Maximum allowed value depending on MpegH3daProfileLevelIndication				
	Lvl 1	Lvl 2	Lvl 3	Lvl 4	Lvl 5
HOA order (max)	2	4	6	6	6
Number of predominant sounds (max)	3	5	7	8	8
Number of directional signals used in prediction (max): MaxNoOfDirSigsForPrediction	2	3	no restrictions apply	no restrictions apply	no restrictions apply
The near field compensation (NFC) processing may be applied to HOA content of an order which is smaller or equal to:	N/A (NFC not allowed)	1	2	3	3

NFC may be employed in not more than one signal group of type SignalGroupTypeHOA.

The following restrictions apply to MPEG-D DRC (ISO/IEC 23003-4) when employed as part of MPEG-H 3D audio.

- drcFrameSizePresent and timeDeltaMinPresent shall be set to 0.
- gainInterpolationType shall be set to 1.
- dependsOnDrcSetPresent shall be set to 0 for drcInstructionsUniDrc() with downmixId == 0.
- HOA signal groups shall be restricted to one drcChannelGroup and DRC gains shall be applied to the HOA core channels (HOATransportChannels).
- The values of bsSequenceIndex within drcInstructionsUniDrc() shall be unique in simultaneously applied DRC sets except for bsSequenceIndex == 0.
- Multiband DRC shall be restricted to drcInstructionsUniDrc() with downmixId == 0. If the bitstream should contain multiband DRC, the number of multiband DRC core channels shall be restricted as follows:

$$\begin{aligned}
 &(\text{numAudioChannels} + \\
 &\text{numAudioObjects} \quad + \text{numAudioObjectsMB} + \\
 &\text{numHOATransportChannels} \quad + \text{numHOATransportChannelsMB}) \\
 &\leq (\text{numCoreChannelsMax(Lvl)} - \text{dependsOnDrcSetPresentFlag} - 1),
 \end{aligned}$$

where

- numAudioChannels, numAudioObjects and numHOATransportChannels are the number of C, O and HOA core channels as specified in Table 17;
- numAudioObjectsMB and numHOATransportChannelsMB are the number of O and HOA core channels out of numAudioObjects and numHOATransportChannels that contain multiband DRC;
- numCoreChannelsMax is the maximum number of decoder processed core channels depending on the Mpeg3daProfileLevelIndication field as defined in Table 3;
- dependsOnDrcSetPresentFlag is set to one if the bitstream contains any configuration with dependsOnDrcSetPresent==1 (otherwise zero).
- nNodes shall be restricted to a maximum value of 32, where nNodes is the number of encoded gain values in the current DRC frame.
- loudnessInfoSetPresent within mpeg3daUniDrcConfig() shall be set to 0.
- nDrcChannelGroups shall be restricted to 1 for drcInstructionsUniDrc() with downmixId != 0. If in this case targetChannelCountFromDownmixId is greater than 1, then repeatSequenceIndex shall be set to 1, and bsRepeatSequenceCount shall be targetChannelCountFromDownmixId - 2.

Table 6 lists the restrictions applying to DRC processing according to the levels of the low complexity profile.

Table 6 — Restrictions applying to DRC processing according to the levels of the low complexity profile

Restriction applies to	Maximum allowed value depending on Mpeg3daProfileLevelIndication				
	Lvl 1	Lvl 2	Lvl 3	Lvl 4	Lvl 5
nDrcChannelGroupsTotal ^a	5	9	16	28	28
drcCoefficientsUniDrcCount	4	4	4	4	4
bandCount ^b	2	4	4	4	4
sequenceCountTotal ^c	24	28	32	48	63
drcInstructionsUniDrcCount	16	16	32	32	32
^a Maximum allowed number of simultaneously active DRC channel groups in all applied DRC sets. ^b Maximum allowed number of DRC bands for multiband DRC. ^c Sum of all nDrcBands in drcGainSequence() structures plus number of sequences with gainCodingProfile=3.					

The following tool specific restrictions apply:

- 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.
- The independent noise filling of the intelligent gap filling shall not be employed in cases where `igfBgn` corresponds to an audio frequency higher than 8 kHz.
- The LPD mode shall only be employed at 3D audio core coder sampling rates (as defined in Table 15) $\leq 32\,000$ Hz.

EXAMPLE For a 48 kHz input signal, the encoder resamples the signal to a 32 kHz core coder sampling rate and the LPD decoder operates at this lower sampling rate. After the core decoding the signal is resampled to 48 kHz.

- The multi-channel coding tool (MCT) shall not employ more stereo boxes than specified in Table 7.

Table 7 — Restrictions applying to MCT processing according to the levels of the low complexity profile

Restriction applies to	Maximum allowed value depending on MpegH3daProfileLevelIndication				
	Lvl 1	Lvl 2	Lvl 3	Lvl 4	Lvl 5
Number of stereo boxes in MCT	5	9	16	28	28

The following restrictions apply to coding of audio objects and the associated OAM data, according to Table 8.

Table 8 — Restrictions applying to object processing according to the levels of the low complexity profile

Restriction applies to	Maximum allowed value n depending on MpegH3daProfileLevelIndication				
	Lvl 1	Lvl 2	Lvl 3	Lvl 4	Lvl 5
$(\text{number of objects without divergence}) + 3 \cdot (\text{number of objects with divergence} > 0) \leq n$	5	9	16	28	28

- Efficient object metadata decoding is not permitted, i.e. `lowDelayMetadataCoding` shall be 1.
- Furthermore the OAM frame length shall comply to:

$$\text{OAMFrameLength} = \text{outputFrameLength} / n,$$
 with n being a positive integer in the range of $\{1, \dots, 4\}$

- Objects shall not employ divergence and spread at the same time.

- If an object is defined with a spatial extent (spread $\alpha > 0,0^\circ$ for uniform spread, spread_width $\alpha_{\text{width}} > 0,0^\circ$ for non-uniform spread) it shall have a divergence value equal to zero.
- If an object is defined with a divergence value > 0 , it shall not have a spatial extent (spread α shall be equal to $0,0^\circ$ for uniform spread, spread_width α_{width} shall be equal to $0,0^\circ$ for non-uniform spread).

The following restrictions apply to binaural rendering:

The value of bsBinauralDataFormatID in BinauralRendering() should be set to 1 (if the FD Binaural renderer is implemented) or to 2 (if the TD Binaural renderer is implemented). The value of bsBinauralDataFormatID can be set to 0 if the parameterization of binaural room impulse responses according to subclauses 13.2.3 or 13.3.3 is implemented.

The number of BRIR sets is restricted to a maximum number of 3.

In case of H2B filters, the number of BRIR filter pairs to be provided shall correspond to 'Maximum H2B filter order' column in Table 9. In the other cases, the following applies.

The number of BRIR pairs in each BRIR set shall correspond to the number indicated in the relevant level-dependent row of Table 9. The measured BRIR positions shall correspond to all nominal geometric positions corresponding to the list of LoudspeakerGeometry indices in Table 9. The correspondence between LoudspeakerGeometry index and nominal geometric position is defined in ISO/IEC 23091-3. Thereby, it is ensured that one BRIR pair is available for each possible regular input channel configuration that can be used within the indicated level.

An input channel configuration is regular if it is defined by means of an ISO/IEC 23091-3 ChannelConfiguration or a list of ISO/IEC 23091-3 LoudspeakerGeometry (CICPspeakerIdx).

If binaural rendering is activated, the measured BRIR positions shall be passed to the mpeg3daLocalSetupInformation(). Thus, all renderer stages are set to the target layout that is equal to the transmitted channel configuration. As one BRIR is available per regular input channel, the format converter can be passed through in case regular input channel positions are used.

Table 9 — The binaural restrictions for the low complexity profile

Level	Number of BRIR pairs	Maximum H2B filter order	BRIR positions by means of loudspeaker position abbreviation	BRIR positions by means of loudspeaker geometry according to ISO/IEC 23091-3: 2018
1	3	1	L, R, C	0, 1, 2
2	10	2	L, R, C, LS, Rs, Lc, Rc, Lsr, Rsr, Cs	0, 1, 2, 4, 5, 8, 9, 10, 15, 16
3	21	3	L, R, C, LS, Rs, Lc, Rc, Lsr, Rsr, Cs, Lss, Rss, Lv, Rc, Cv, Lvr, Rvr, Cvr, Rs, Lvs, Rvs	0, 1, 2, 4, 5, 8, 9, 10, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 25, 30, 31
4	28	5	L, R, C, LS, Rs, Lc, Rc, Lsr, Rsr, Cs, Lss, Rss, Lv, Rc, Cv, Lvr, Rvr, Cvr, Lvss, Rvss, Ts, Lb, Rb, Cb, Lvs, Rvs, Lbs, Rbs,	0, 1, 2, 4, 5, 8, 9, 10, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31, 37, 38
5	28	6	L, R, C, LS, Rs, Lc, Rc, Lsr, Rsr, Cs, Lss, Rss, Lv, Rc, Cv, Lvr, Rvr, Cvr, Lvss, Rvss, Ts, Lb, Rb, Cb, Lvs, Rvs, Lbs, Rbs,	0, 1, 2, 4, 5, 8, 9, 10, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31, 37, 38

The following additional parameter values restrictions apply.

- The value of *kMax* in *FdBinauralRendererParam()* shall be equal to or less than 48 (bands).
- The value of *kConv* in *FdBinauralRendererParam()* shall be equal to 32.
- The values of *rt60[k]* in *SfrBrirParam()* shall be less than or equal to 1,0 (sec).
- The average of the values of *nFilter[k]* shall be less than or equal to 64.
- The values of *nFilter[k]* in *VoffBrirParam()* should be less than or equal to 256.

The following coding tools, modules, or features shall not be employed:

- Time warped filterbank;
- 768 sample *outputFrameLength*, i.e. *coreSbrFrameLengthIndex* shall not be 0

The following text describes restrictions dependent on the length of the arithmetic coder codeword, *arith_data()*. For this text the following definitions apply.

F_{sOut}	core coder sampling rate as indicated by means of <i>usacSamplingFrequencyIndex</i> or <i>usacSamplingFrequency</i> in <i>mpegh3daConfig()</i>
F_{sMax}	maximum allowed sampling rate of a given level in this profile
N_{chMax}	maximum number of decoder processed core channels of a given level in this profile according to Table 3.
N_{chLtpf}	number of core coder channels in which the long term post filter (LTPF) is applied
N_{chInf}	number of core coder channels in which the independent noise filling (INF) is applied
$Nbits_{arith_data()}(ch)$	number of bits used for arithmetic coding of spectral data, <i>arith_data()</i> , for core coder channel <i>ch</i> for a given frame

$Nbits_{arith_all} = \sum_{\text{all channels}} Nbits_{arith_data()}(ch)$, i.e. the sum of all bits used for the arithmetic coding of spectral data of all decoder processed core coder channels

- In any given audio frame $Nbits_{arith_all}$ shall comply with the following restriction:

$$Nbits_{arith_all} < \frac{(3072 \cdot N_{chMax} - 2048 \cdot N_{chLtpf} - 2048 \cdot N_{chInf}) \cdot F_{sMax}}{F_{sOut}}$$

The following restrictions apply to the *AudioPreRoll()* extension.

- Decoders conforming to this profile shall support the full decoding and correct handling of the *AudioPreRoll()* extension.
- The number of pre-roll frames, *numPreRollFrames*, in an *AudioPreRoll()* extension payload shall not exceed 1 (one).

- In access units that are embedded as pre-roll in an AudioPreRoll() extension the usacExtElementPresent field for extensions of type ID_EXT_ELE_AUDIOPREROLL shall be 0.

The following restrictions apply to the employed sampling rate and the resampler block.

- The sampling rate that is signalled by means of usacSamplingFrequencyIndex as defined in ISO/IEC 23003-3:2020, Table 67 or usacSamplingFrequency shall be one of the values in the first column of Table 10.
- Depending on the above mentioned sampling rate and the profile level the resampler shall employ one of the resampling ratios indicated in Table 10.

Table 10 — Allowed sampling rates and resampling ratios

Allowed sampling rate	Allowed resampling ratio depending on MpegH3daProfileLevelIndication				
	Lvl 1	Lvl 2	Lvl 3	Lvl 4	Lvl 5
96 000	N/A	N/A	N/A	N/A	1
88 200	N/A	N/A	N/A	N/A	1
64 000	N/A	N/A	N/A	N/A	1.5
58 800	N/A	N/A	N/A	N/A	1.5
48 000	1	1	1	1	1 or 2
44 100	1	1	1	1	1 or 2
32 000	1.5	1.5	1.5	1.5	1.5 or 3
29 400	1.5	1.5	1.5	1.5	1.5 or 3
24 000	2	2	2	2	2
22 050	2	2	2	2	2
16 000	3	3	3	3	3
14 700	3	3	3	3	3

The following restrictions apply to the coding of the audio scene information structure, according to Table 11.

Table 11 — ProfileLevel dependent restrictions to selected fields of the mae_AudioSceneInfo()

Restriction applies to	Allowed maximum value depending on MpegH3daProfileLevelIndication				
	Lvl 1	Lvl 2	Lvl 3	Lvl 4	Lvl 5
mae_numGroups	5	9	16	28	28
mae_numSwitchGroups	2	4	8	14	14
mae_numGroupPresets	4	4	8	16	31
(mae_bsGroupPresetNumConditions +1)	5	9	16	16	16
mae_numDownmixIdGroupPresetExtensions per mae_groupPresetID	4	4	8	16	31
(mae_bsNumDescLanguages +1)	4	4	4	8	16
(mae_bsDescriptionDataLength +1)	256	256	256	256	256

— If mae_numSwitchGroups > 0, then elementLengthPresent shall be set to 1.

The following restriction applies to the usage of the contribution mode.

— For end user devices the contribution mode of MPEG-H 3D audio as defined in subclause 4.7 shall not be supported.

4.8.2.3 Levels of the main profile

The description of the main profile, its associated bitstream syntax, semantics, and decoding process is no longer needed for the implementation of this document.

4.8.2.4 Levels of the high profile

Placeholder for high profile.

4.8.2.5 Levels of the baseline profile

4.8.2.5.1 General

Table 12 — Levels and their corresponding restrictions for the baseline profile

Level	Max. sampling rate	Max. number of core channels in compressed data stream	Max. number of decoder processed core channels	Max. number of channels in referenceLayout
1	48000	10	5	5
2	48000	18	9	9
3	48000	32	16 ^a or 24 ^b	16 ^a or 24 ^b
4	48000	56	28	24
5	96000	56	28	24
^a No additional complexity restrictions are applied.				
^b Additional complexity restrictions given in 4.8.2.5.2 are applied.				

- The use of switch groups determines the subset of core channels from the core channels in the bitstream that shall be decoded.
- If the `mae_AudioSceneInfo()` contains switch groups (`mae_numSwitchGroups>0`), then the `elementLengthPresent` flag shall be 1.
- The number of channels of the signalled `referenceLayout` shall not exceed the values defined in the levels in Table 12.
- Object renderer and binaural renderer that perform at least as well as the object and binaural renderer specified in Clauses 8 and 13 may be integrated using the output interfaces for un-rendered channels and objects described in subclause 17.10.

NOTE The performance recommendation covers the behaviour of the decoder over the complete decoding and rendering chain, especially for the case of configuration changes as described in subclause 5.5.6, mixing of channel and object content or DRC processing, loudness compensation and user interactivity.

- For Level 3 the maximum number of decoder processed core channels and maximum number of channels signalled in `referenceLayout` is:
 - a) 16 if no additional complexity restrictions are applied,
 - b) 24 if all the complexity restrictions in 4.8.2.5.2 are applied.

4.8.2.5.2 Complexity restrictions for Level 3 with more than 16 decoder processed core channels

- **signalGroupType** in `Signals3d()` shall indicate `SignalGroupTypeObject` (Objects only).
- **usacElementType[elemIdx]** in `mpegh3daDecoderConfig()` shall indicate `ID_USAC_SCE` or `ID_USAC_EXT`.
- **noiseFilling** and **enhancedNoiseFilling** in `mpegh3daCoreConfig()` shall be set to "0".
- **usacExtElementType** in `mpegh3daExtElementConfig()` shall not be set to `ID_EXT_ELE_MCT`.
- Long term prediction filter shall not be used, i.e., **ltpf_data_present** and **common_ltpf** shall be set to "0".
- Frequency domain predictor shall not be used, i.e., **fdp_data_present** shall be set to "0".

4.8.2.6 Restrictions for the baseline profile and levels

All restrictions defined for low complexity profile in subclause 4.8.2.2 shall apply.

The LPD path of the core coder and HOA path are not supported.

Restrictions defined in Table 13 shall apply.

Table 13 — Baseline profile restrictions

MPEG-H 3D audio bit field	Structure	Use description
<i>phaseAlignStrength</i>	<i>downmixConfig()</i>	Shall have the value "0"
<i>SignalGroupType[grp]</i>	<i>Signals3d()</i>	Shall have the value "SignalGroupTypeChannels" or "SignalGroupTypeObject"
<i>qceIndex</i>	<i>mpegh3daChannelPairElementConfig()</i>	Shall have the value "0"
<i>lpdStereoIndex</i>	<i>mpegh3daChannelPairElementConfig()</i>	Shall have the value "0"
<i>tw_mdct</i>	<i>mpegh3daCoreConfig()</i>	Shall have the value "0"
<i>fullbandLpd</i>	<i>mpegh3daCoreConfig()</i>	Shall have the value "0"
<i>core_mode[ch]</i>	<i>mpegh3daCoreCoderData()</i>	Shall have the value "0"
<i>common_max_sfb</i>	<i>StereoCoreToolInfo()</i>	Shall have the value "1"
<i>tns_on_lr</i>	<i>StereoCoreToolInfo()</i>	Shall have the value "1"
<i>common_tw</i>	<i>StereoCoreToolInfo()</i>	Shall have the value "0"
<i>fac_data_present</i>	<i>fd_channel_stream()</i>	Shall have the value "0"

4.8.2.7 Signalling of profile and level compatibility sets

MPEG-H 3d audio bitstreams may comply with multiple profiles and levels and the `CompatibleProfileLevelSet()` syntax element defined in Table 14 may be used to signal the compatibility to multiple profiles.

The `CompatibleProfileLevelSet()` syntax element contains a list of profile-level numbers the content is compatible with. Only the lowest level per profile needs to be present, as higher level decoders are inherently compatible with lower level content.

Table 14 — Syntax of `CompatibleProfileLevelSet()`

Syntax	No. of bits	Mnemonic
<code>CompatibleProfileLevelSet()</code>		
{		
bsNumCompatibleSets;	4	uimsbf
numCompatibleSets = bsNumCompatibleSets + 1;		
reserved;	4	uimsbf
for (idx = 0; idx < numCompatibleSets; idx++) {		
CompatibleSetIndication;	8	uimsbf
}		
}		

5 MPEG-H 3D audio core decoder

5.1 Definitions

5.1.1 Joint stereo

The MDCT domain-based joint stereo coding tool with the possibility of complex stereo prediction is as defined in ISO/IEC 23003-3:2020, subclause 7.7.

5.1.2 MPEG surround based stereo (MPS 212)

The MPEG surround 2-1-2 based stereo tool working in QMF domain is as defined in ISO/IEC 23003-3:2020, subclause 7.11, with the possibility of using residual coding (unified stereo) as specified in ISO/IEC 23003-3:2020, B.21.

5.2 Syntax

5.2.1 General

The bitstream syntax is based on ISO/IEC 23003-3:2020, Clause 5.

Modifications and amendments to the existing bitstream syntax are listed below.

In environments that require byte alignment, MPEG-H 3D audio configuration elements or payload elements that are not an integer number of bytes in length are padded at the end to achieve an integer byte count.

5.2.2 Decoder configuration

5.2.2.1 General configuration syntax

Table 15 lists the syntax of `mpegh3daConfig()`.

Table 15 — Syntax of `mpegh3daConfig()`

Syntax	No. of bits	Mnemonic
<code>mpegh3daConfig()</code>		
{		
mpegh3daProfileLevelIndication;	8	uimsbf
usacSamplingFrequencyIndex;	5	bslbf
if (<code>usacSamplingFrequencyIndex == 0x1f</code>) {		
usacSamplingFrequency;	24	uimsbf
} else {		
<code>usacSamplingFrequency;</code>		a
}		
coreSbrFrameLengthIndex;	3	uimsbf
<code>outputFrameLength;</code>		b
cfg_reserved;	1	uimsbf
receiverDelayCompensation;	1	uimsbf
<code>referenceLayout = SpeakerConfig3d();</code>		
<code>frameworkConfig3d = FrameworkConfig3d(referenceLayout);</code>		
<code>decoderConfig = mpegh3daDecoderConfig(</code>		
<code>frameworkConfig3d.Signals3d.bsNumSignalGroups,</code>		
<code>frameworkConfig3d.Signals3d.bsNumberOfSignals,</code>		
<code>frameworkConfig3d.Signals3d.numAudioChannels,</code>		
<code>frameworkConfig3d.Signals3d.numAudioObjects,</code>		
<code>usacSamplingFrequency,</code>		
<code>outputFrameLength</code>		
);		

Syntax	No. of bits	Mnemonic
<pre> if (usacConfigExtensionPresent) { configExtension = mpeg3daConfigExtension(frameworkConfig3d); } </pre>	1	uimsbf
^a The value of usacSamplingFrequencyIndex and their associated sampling frequencies are described in ISO/IEC 23003-3:2020, subclause 6.1.1.1, Table 67. ^b The value of outputFrameLength is defined in ISO/IEC 23003-3:2020, subclause 6.1.1.1, Table 70.		

Table 16 lists the syntax of FrameworkConfig3d().

Table 16 — Syntax of FrameworkConfig3d()

Syntax	No. of bits	Mnemonic
<pre> FrameworkConfig3d(referenceLayout) { Signals3d(referenceLayout); } </pre>		

Table 17 lists the syntax of Signals3d().

Table 17 — Syntax of Signals3d()

Syntax	No. of bits	Mnemonic
<pre> Signals3d(referenceLayout) { numAudioChannels = 0; numAudioObjects = 0; numSAOTransportChannels = 0; numHOATransportChannels = 0; bsNumSignalGroups; for (grp = 0; grp < bsNumSignalGroups + 1 ; grp++) { signal_groupID[grp] = grp; differsFromReferenceLayout[grp] = 0; SignalGroupType[grp]; bsNumberOfSignals[grp] = escapedValue(5, 8, 16); if (SignalGroupType[grp] == SignalGroupTypeChannels) { numAudioChannels += bsNumberOfSignals[grp] + 1; differsFromReferenceLayout[grp]; if (differsFromReferenceLayout[grp]) { audioChannelLayout[grp] = SpeakerConfig3d(); } else { audioChannelLayout[grp] = referenceLayout; } } if (SignalGroupType[grp] == SignalGroupTypeObject) { numAudioObjects += bsNumberOfSignals[grp] + 1; } } } </pre>	5 3 1	uimsbf bslbf bslbf

Syntax	No. of bits	Mnemonic
<pre> if (SignalGroupType[grp] == SignalGroupTypeSAOC) { numSAOCTransportChannels += bsNumberOfSignals[grp] + 1; saocDmxLayoutPresent; if (saocDmxLayoutPresent == 1) { saocDmxChannelLayout = SpeakerConfig3d(); } } if (SignalGroupType[grp] == SignalGroupTypeHOA) { numHOATransportChannels += bsNumberOfSignals[grp] + 1; } } </pre>	1	bslbf

5.2.2.2 Loudspeaker configuration syntax

Table 18 lists the syntax of `SpeakerConfig3d()`.

Table 18 — Syntax of SpeakerConfig3d()

[illegible]

Table 19 lists the syntax of `mpegh3daFlexibleSpeakerConfig()`.

Table 19 — Syntax of mpeg3daFlexibleSpeakerConfig()

Syntax	No. of bits	Mnemonic
<pre> mpeg3daFlexibleSpeakerConfig(numSpeakers) { angularPrecision; for (i = 0; i < numSpeakers; i++) { mpeg3daSpeakerDescription(); if ((AzimuthAngle != 0°) && (AzimuthAngle != 180°)) { alsoAddSymmetricPair; if (alsoAddSymmetricPair) { i++; } } } } </pre>	1 1	uimsbf ^a uimsbf ^b
^a The value of AzimuthAngle can be calculated using Table 20.		
^b Also add the loudspeaker with the opposite AzimuthDirection.		

Table 20 lists the syntax of mpeg3daSpeakerDescription().

Table 20 — Syntax of mpeg3daSpeakerDescription()

Syntax	No. of bits	Mnemonic
<pre> mpeg3daSpeakerDescription(angularPrecision) { isCICPspeakerIdx; if (isCICPspeakerIdx) { CICPspeakerIdx; } else { ElevationClass; if (ElevationClass == 3) { ElevationAngleIdx; if (ElevationAngle != 0°) { ElevationDirection; } } AzimuthAngleIdx; if ((AzimuthAngle != 0°) && (AzimuthAngle != 180°)) { AzimuthDirection; } isLFE; } } </pre>	1 7 2 5, 7 1 6, 8 1 1	uimsbf uimsbf uimsbf ^a uimsbf uimsbf ^a uimsbf uimsbf
^a The number of bits for ElevationAngleIdx and AzimuthAngleIdx depends on the value of angularPrecision according to Table 70. The value of ElevationAngle and AzimuthAngle can be derived from Table 72 and Table 74, respectively. In case isCICPspeakerIdx is one, ElevationAngle and AzimuthAngle shall be signalled by means of a LoudspeakerGeometry as defined in ISO/IEC 23091-3.		

5.2.2.3 Core decoder configuration

Table 21 lists the syntax of mpeg3daDecoderConfig().

Table 21 — Syntax of mpeg3daDecoderConfig()

Syntax	No. of bits	Mnemonic
<pre> mpeg3daDecoderConfig(bsNumSignalGroups, bsNumberOfSignals[], numAudioChannels, numAudioObjects, sampleRate, outputFrameLength) { grp = 0; grpSignalCnt = 0; numElements = escapedValue(4,8,16) + 1; elementLengthPresent; for (elemIdx=0; elemIdx<numElements; elemIdx++) { usacElementType[elemIdx]; switch (usacElementType[elemIdx]) { case ID_USAC_SCE: usacElement[elemIdx] = mpeg3daSingleChannelElementConfig(sbrRatioIndex); grpSignalCnt++; break; case ID_USAC_CPE: usacElement[elemIdx] = mpeg3daChannelPairElementConfig(sbrRatioIndex, numAudioChannels, numAudioObjects); grpSignalCnt += 2; break; case ID_USAC_LFE: usacElement[elemIdx] = mpeg3daLfeElementConfig(); grpSignalCnt++; break; case ID_USAC_EXT: usacElement[elemIdx] = mpeg3daExtElementConfig(bsNumberOfSignals[grp], grp, sampleRate, outputFrameLength); break; } if(grpSignalCnt == bsNumberOfSignals[grp] + 1) { grp++; grp = min(grp, bsNumSignalGroups); grpSignalCnt = 0; } } } </pre>	1	uimsbf
	2	uimsbf
NOTE mpeg3daSingleChannelElementConfig(), mpeg3daChannelPairElementConfig(), mpeg3daLfeElementConfig() and mpeg3daExtElementConfig() signalled at position elemIdx refer to the corresponding elements in mpeg3daFrame() at the respective position elemIdx.		

Table 22 lists the syntax of mpeg3daSingleChannelElementConfig().

Table 22 — Syntax of mpeg3daSingleChannelElementConfig()

Syntax	No. of bits	Mnemonic
<pre> mpeg3daSingleChannelElementConfig(sbrRatioIndex) { mpeg3daCoreConfig(); if (sbrRatioIndex > 0) { SbrConfig(); } } </pre>		

Table 23 lists the syntax of mpeg3daChannelPairElementConfig().

Table 23 — Syntax of mpeg3daChannelPairElementConfig()

Syntax	No. of bits	Mnemonic
<pre> mpeg3daChannelPairElementConfig(sbrRatioIndex, numAudioChannels, numAudioObjects) { coreConfig = mpeg3daCoreConfig(); if (coreConfig.enhancedNoiseFilling) { igfIndependentTiling; } if (sbrRatioIndex > 0) { SbrConfig(); stereoConfigIndex; } else { stereoConfigIndex = 0; } if (stereoConfigIndex > 0) { Mps212Config(stereoConfigIndex); } qceIndex; if (qceIndex > 0) { shiftIndex0; if (shiftIndex0 > 0) { nBits = floor(log2(numAudioChannels + numAudioObjects + numHOATransportChannels + numSAOCTransportChannels - 1)) + 1; shiftChannel0; } } shiftIndex1; if (shiftIndex1 > 0) { nBits = floor(log2(numAudioChannels + numAudioObjects + numHOATransportChannels + numSAOCTransportChannels - 1)) + 1; } } </pre>	1 2 2 1 nBits 1	bslbf uimsbf Uimsbf uimsbf uimsbf

Syntax	No. of bits	Mnemonic
shiftChannel1;	nBits	
}		
if (sbrRatioIndex == 0 && qceIndex == 0) {		
 lpdStereoIndex;	1	bslbf
} else {		
 lpdStereoIndex = 0;		
}		
}		

Table 24 lists the syntax of mpeg3daCoreConfig().

Table 24 — Syntax of mpeg3daCoreConfig()

Syntax	No. of bits	Mnemonic
mpeg3daCoreConfig()		
{		
 tw_mdct;	1	bslbf
 fullbandLpd;	1	bslbf
 noiseFilling;	1	bslbf
 if (enhancedNoiseFilling) {	1	bslbf
 igfUseEnf;	1	bslbf
 igfUseHighRes;	1	bslbf
 igfUseWhitening;	1	bslbf
 igfAfterTnsSynth;	1	bslbf
 igfStartIndex;	5	uimsbf
 igfStopIndex;	4	uimsbf
 }		
}		

Table 25 lists the syntax of mpeg3daLfeElementConfig().

Table 25 — Syntax of mpeg3daLfeElementConfig()

Syntax	No. of bits	Mnemonic
mpeg3daLfeElementConfig()		
{		
 tw_mdct = 0;		
 fullbandLpd = 0;		
 noiseFilling = 0;		
 enhancedNoiseFilling = 0;		
}		

Table 26 lists the syntax of mpeg3daExtElementConfig().

Table 26 — Syntax of mpeg3daExtElementConfig()

Syntax	No. of bits	Mnemonic
<pre> mpeg3daExtElementConfig(bsNumberOfSignals[], outputFrameLength, grp) { usacExtElementType = escapedValue(4, 8, 16); usacExtElementConfigLength = escapedValue(4, 8, 16); if (usacExtElementDefaultLengthPresent) { usacExtElementDefaultLength = escapedValue(8, 16, 0) + 1; } else { usacExtElementDefaultLength = 0; } usacExtElementPayloadFrag; If (usacExtElementConfigLength > 0) { switch (usacExtElementType) { case ID_EXT_ELE_FILL: /* No configuration element */ break; case ID_EXT_ELE_MPEGS: SpatialSpecificConfig(); break; case ID_EXT_ELE_SAOC: SAOCSpecificConfig(); break; case ID_EXT_ELE_AUDIOPREROLL: /* No configuration element */ break; case ID_EXT_ELE_UNI_DRC: mpeg3daUniDrcConfig(); break; case ID_EXT_ELE_OBJ_METADATA: ObjectMetadataConfig(); break; case ID_EXT_ELE_SAOC_3D: SAOC3DSpecificConfig(); break; case ID_EXT_ELE_HOA: HOAConfig(); break; case ID_EXT_ELE_FMT_CNVTR: /* No configuration element */ break; case ID_EXT_ELE_MCT: MCTConfig(); break; case ID_EXT_ELE_TCC: TccConfig(); break; case ID_EXT_ELE_HOA_ENH_LAYER: HOAEnhConfig(); break; } } } </pre>	1 1	uimsbf uimsbf

Syntax	No. of bits	Mnemonic
<pre> case ID_EXT_ELE_HREP: HREPConfig(current_signal_group); break; case ID_EXT_ELE_ENHANCED_OBJ_METADATA: EnhancedObjectMetadataConfig(); break; case ID_EXT_ELE_PROD_METADATA: prodMetadataConfig(); break; default: length = usacExtElementConfigLength; while (length--) { tmp; } break; } </pre>	a 8	uimbsf
^a The default entry for the usacExtElementType is used for unknown extElementTypes so that legacy decoders can cope with future extensions.		

Table 27 lists the syntax of `mpegh3daConfigExtension()`.

Table 27 — Syntax of mpeg3daConfigExtension()

Syntax	No. of bits	Mnemonic
<pre> mpeg3daConfigExtension() { numConfigExtensions = escapedValue(2,4,8) + 1; for (confExtIdx=0; confExtIdx<numConfigExtensions; confExtIdx++) { usacConfigExtType[confExtIdx] = escapedValue(4,8,16); usacConfigExtLength[confExtIdx] = escapedValue(4,8,16); switch (usacConfigExtType[confExtIdx]) { case ID_CONFIG_EXT_FILL: while (usacConfigExtLength[confExtIdx]--) { fill_byte[i]; /* should be '10100101' */ } break; case ID_CONFIG_EXT_DOWNMIX: downmixConfig(); break; case ID_CONFIG_EXT_LOUDNESS_INFO: mpeg3daLoudnessInfoSet(); break; case ID_CONFIG_EXT_AUDIOSCENE_INFO: mae_AudioSceneInfo(); break; case ID_CONFIG_EXT_HOA_MATRIX: HoaRenderingMatrixSet(); break; </pre>	8	uimsbf