INTERNATIONAL STANDARD

ISO/IEC 14496-4

Second edition
2004-12-15

AMENDMENT 22
2008-03-01

Information technology — Coding of audio-visual objects —

Part 4: Conformance testing

AMENDMENT 22: AudioBIFS v3 conformance

Technologies de l'information — Codage des objets audiovisuels —

Partie 4: Essai de conformité

AMENDEMENT 22: Conformité AudioBIFS v3

Reference number
ISO/IEC 14496-4:2004/Amd.22:2008(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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

Amendment 22 to ISO/IEC 14496-4:2004 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Coding of audio-visual objects —

Part 4:
Conformance testing

AMENDMENT 22: AudioBIFS v3 conformance

In subclause 4.4.3.3 Bitstreams, insert the following row into Table 6 after the row for AABper1-76:

Table 6 – Test Sequence Providers and Reason for Existence

ABv3_AAB01-05	Thomson	Tests AudioBIFS v3 node AdvancedAudioBuffer
ABv3_ACC01-05 ABv3_ACC06a-c ABv3_ACC07a-c ABv3_ACC09-13	FhG, Thomson, FT	Tests AudioBIFS v3 node AudioChannelConfig
ABv3_SS01a-b ABv3_SS02a-b ABv3_SS04-05	FhG, FT	Tests AudioBIFS v3 node SurroundingSound
ABv3_T3DA01-08	FhG	Tests AudioBIFS v3 node Transform3Daudio
ABv3_WS0101-03	FhG, Thomson,	Tests AudioBIFS v3 node WideSound
Abv3_Afxp_Ach01 Abv3_Afxp_Aco01 Abv3_Afxp_Aec01 Abv3_Afxp_Aeq01 Abv3_Afxp_Afi01 Abv3_Afxp_Afl01 Abv3_Afxp_Ana01 Abv3_Afxp_Are01 Abv3_Afxp_Asp01 Abv3_Afxp_Ast01 Abv3_Afxp_Avi01	Thomson	Tests AudioBIFS v3 AudioFXProto effects

In subclause 6.5.1 File name conventions, insert the following row into Table 29 in alphabetical order:

Table 29 – File name conventions

A.1 AudioBIFS v3	A.2 ABv3_<nodeAbbrev><coreSetup>	A.3 -- not applicable --

*In subclause 6.5.1 File name conventions, insert the following paragraph after "***_s<speedfac>*** is a number refering to the decoder configuration with regard to the speed factor.":*

<nodeAbbrev> is the abbreviation of one of the AudioBIFS v3 node names (see Table 6).

After subclause 6.11.5.4.4.2 Test scenes, insert the following subclauses:

6.12 AudioBIFS v3 Nodes

6.12.1 Introduction

This clause describes the conformance testing for the rendering and output of AudioBIFS v3 nodes, which are used for adding support for shaped sound sources (**WideSound** node), Ambisonics™ audio streams to two- and three-dimensional BIFS scenes. Furthermore they allow 3D sound in 2D visual scenes (**Transform3DAudio** node) and simplify the use of pre-defined audio effects (**AudioFXProto**s). With AudioBIFS v3 also a labelling mechanism was introduced that transports the channel configuration information through the AudioBIFS sub-graph and can be altered with the **AudioChannelConfig** node.

6.12.2 Composition Unit Inputs

The input audio streams used in the conformance testing of AudioBIFS v3 shall be outputs of an AAC decoder (AOT 2), and they are monophonic, stereophonic, 5.1 (surround) and Ambisonics™ sounds. They are explained below:

ABv3_CU01_2ch_AOT2_L1R2: Composition Unit Input AAC: white noise for one second on left, for one second on right channel. Duration: 8 seconds, sampling rate 44100 Hz, stereo.

ABv3_CU02_2ch_AOT2_L3R4: Composition Unit Input AAC: after 2 seconds silence white noise for one second on left, then for one second on right channel. Duration: 8 seconds, sampling rate 44100 Hz, stereo.

ABv3_CU03_2ch_AOT2_L5R6: Composition Unit Input AAC: after 4 seconds silence white noise for one second on left, then for one second on right channel. Duration: 8 seconds, sampling rate 44100 Hz, stereo.

ABv3_CU04_1ch_AOT2_M5: Composition Unit Input AAC: after 4 seconds silence white noise for one second. Duration: 8 seconds, sampling rate 44100 Hz, mono.

ABv3_CU05_1ch_AOT2_M6: Composition Unit Input AAC: after 5 seconds silence white noise for one second. Duration: 8 seconds, sampling rate 44100 Hz, mono.

ABv3_CU06_1ch_AOT2_M0-3: Composition Unit Input AAC: 3 seconds white noise starting from the beginning. Duration: 3 seconds, sampling rate 44100 Hz, mono.

ABv3_CU07_6ch_AOT2_surround: Composition Unit Input AAC: 0.5 seconds silence, then for 0.2 seconds an 880Hz sinus tone on center channel, 0.3s silence, 0.5s 440Hz sin on front left channel, 0.5s 440Hz sin on front right channel, 0.4s 220Hz sin on left surround channel, 0.1s silence, 0.4s 220Hz sin on right surround channel, 0.1s silence, 1s white noise on LFE channel. Duration: 4 seconds, sampling rate 44100 Hz, 5.1 channels.

ABv3_CU08_1ch_AOT2_one: Composition Unit Input AAC: male voice ‘one’. Duration: 1 second, sampling rate 44100 Hz, mono.

ABv3_CU09_1ch_AOT2_two: Composition Unit Input AAC: male voice ‘two’. Duration: 1 second, sampling rate 44100 Hz, mono.

ABv3_CU10_1ch_AOT2_three: Composition Unit Input AAC: male voice ‘three’. Duration: 1 second, sampling rate 44100 Hz, mono.

ABv3_CU11_1ch_AOT2_four: Composition Unit Input AAC: male voice ‘four’. Duration: 1 second, sampling rate 44100 Hz, mono.

ABv3_CU12_1ch_AOT2_five: Composition Unit Input AAC: male voice ‘five’. Duration: 1 second, sampling rate 44100 Hz, mono.

ABv3_CU13_1ch_AOT2_six: Composition Unit Input AAC: male voice ‘six’. Duration: 1 second, sampling rate 44100 Hz, mono.

ABv3_CU14_2ch_AOT2_LCR_Binaural: Composition Unit Input AAC: male voices ‘left, center, right’, repeated 1 time. Duration: 10 seconds, sampling rate 44100 Hz, 2-channel binaural.

ABv3_CU15_2ch_AOT2_applause: Composition Unit Input AAC: applause. Duration: ~12 seconds, sampling rate 44100 Hz, 2-channel stereo.

ABv3_CU16_1ch_AOT2_applause_mono: Composition Unit Input AAC: applause. Duration: ~12 seconds, sampling rate 44100 Hz, 1-channel mono.

ABv3_CU17_2ch_AOT2_orchestra: Composition Unit Input AAC: orchestra. Duration: ~12 seconds, sampling rate 44100 Hz, 2-channel stereo.

ABv3_CU18_1ch_AOT2_orchestra_mono: Composition Unit Input AAC: orchestra. Duration: ~12 seconds, sampling rate 44100 Hz, 1-channel mono.

ABv3_CU19_1ch_AOT2_W: Composition Unit Input AAC: the omni-directional channel (0th order component W) of a 3D ambisonic recording of an ambient sound field: a complex, mostly static, outdoor scene; birds singing all around, splashes caused by stones thrown in the water first on the left then on the right, a child and adults speaking on the front, slightly on the left. In the test sequences, it has to be decoded and played synchronously with the following audio streams (ABv3_CU20_… to ABv3_CU22_...). Duration: ~14.3 seconds, sampling rate 44100 Hz, 1-channel.

ABv3_CU20_2ch_AOT2_XY: Composition Unit Input AAC: the horizontal, bidirectional (1st order) channels X and Y of the same ambisonic recording as ABv3_CU19_1ch_AOT2_W. Duration: ~14.3 seconds, sampling rate 44100 Hz, 2-channels.

ABv3_CU21_1ch_AOT2_Z: Composition Unit Input AAC: the vertical bidirectional (1st order) channel Z of the same ambisonic recording as ABv3_CU19_1ch_AOT2_W. Duration: ~14.3 seconds, sampling rate 44100 Hz, 1-channel.

ABv3_CU22_2ch_AOT2_UV: Composition Unit Input AAC: the horizontal, 2nd order channels U and V of the same ambisonic recording as ABv3_CU19_1ch_AOT2_W. Duration: ~14.3 seconds, sampling rate 44100 Hz, 2-channels.

ABv3_CU23_1ch_AOT2_ambITU_C: Composition Unit Input AAC: the center channel of a 5.0 playback version of the recording described for ABv3_CU19_1ch_AOT2_W that could be played over a standard ITU loudspeaker arrangement. Duration: ~14.3 seconds, sampling rate 44100 Hz, 1 channel.

ABv3_CU24_2ch_AOT2_ambITU_LR: Composition Unit Input AAC: the front left and front right channels the 5.0 playback version described above. Duration: ~14.3 seconds, sampling rate 44100 Hz, 2 channels.

ABv3_CU25_2ch_AOT2_ambITU_SLSR: Composition Unit Input AAC: the surround left and surround right channels the 5.0 playback version described above. Duration: ~14.3 seconds, sampling rate 44100 Hz, 2 channels.

Abv3_CU26_2ch_AOT2_seawash_cl.media: Composition Unit Input AAC: center channel and left channel ‘seawash’. Duration: ~20 seconds, sampling rate 48000 Hz, 2 channels.

Abv3_CU27_2ch_AOT2_seawash_rs.media: Composition Unit Input AAC: right channel and surround channel ‘seawash’. Duration: ~20 seconds, sampling rate 48000 Hz, 2 channels.

Abv3_AudioNaturalReverb_ImpulseResponse.wav: Original impulse response for binary comparison for test scene Abv3_aFXP_aNa01.

6.12.3 Compositor Output

The output of the audio compositor will be investigated for conformance, and shall be a single output, N channel (depending on the spatialization and reproduction method used) PCM audio stream. The input audio streams are at 16 bit signed integer sample format, and the processing defined by the Advanced Audio BIFS nodes in the scene will be carried out at an accuracy of at least 16 bits.

Because of the non-normative nature of implementing many of the AudioBIFS features, no sample-wise comparison is done to the output sound from the compositor. Some of the features can be evaluated in a static situation (no dynamic changes, such as sound source or viewpoint movements, in the 3D environment) by measuring certain parameters of the compositor's output. Some functionalities, on the other hand, require testing in a dynamic situation where only subjective evaluation can be used (the user is listening to the sound compositor output, and watching the visual compositor output if visual components are present).

In addition to objective and subjective testing a third on one is needed for AudioBIFS v3, the parameter printout. Therefore the printout of certain parameters (like the absolute position of an audio source or the current channel configuration of the audio data) can be compared with a given reference. Note, that the parameter printout does not print the *contents* of a node's field, but the status information like channel configuration that has to be passed through the audio scene graph along with the audio data or through the BIFS scene graph like the transform hierarchy. The parameter printout should be ASCII text with the format

```
parameterName value_1 … value_n
```

whereby value should be in the format of the corresponding field and logged in the top-level sound nodes. In case of a SFFloat/MFFloat field type the 1.16 float format should be used.

6.12.4 Conformance Tests for AudioBIFS v3 Nodes

The following chapters contain the detailed description of the conformance tests for the nodes **AdvancedAudioBuffer**, **AudioChannelConfig**, **SurroundingSound**, **Transform3Daudio**, **WideSound** and **AudioFXProto** effects.

6.12.4.1 Testing of AdvancedAudioBuffer Node

The **AdvancedAudioBuffer** node provides an interface for stored sound. This node has corrected functionality and enhanced reload mechanism compared to the **AudioBuffer** node, e.g to accumulate snippets of sound in the **AdvancedAudioBuffer**. These snippets can be accessed directly or as the full accumulated content.

6.12.4.1.1 BIFS components needed in the conformance testing

For testing the load and playback mechanism of the **AdvancedAudioBuffer** node a minimal set of BIFS nodes is needed: Besides the root node and one grouping node (**Group**, **OrderedGroup**) one top-level AudioBIFS node (**Sound**, **Sound2D**) as well as a node that connects the AudioBIFS sub-graph with the decoder (**AudioSource**) is required.

6.12.4.1.2 Conformance testing procedure

Conformance testing of the **AdvancedAudioBuffer** node requires the player to support the parameter printout of **AdvancedAudioBuffer** node's fields. For subjective testing the listener has to check if playing back the scene has the described effect.

The basic behaviour of this node is determined by the **loadMode** field. For each of these five modes a test scene shall be used to check the functionalities. The read/write access to the different content blocks shall be verified with the corresponding scene by a printout of the index of the content block. Additional subjective listening tests shall be performed.

Test Scenes:

ABv3_AAB01 This scene is used for testing the compatibility mode (**LoadMode**=0) (compatible to **AudioBuffer**, see functionality and semantics in the node definition) of the **AdvancedAudioBuffer** node by loading ABv3_CU8_1ch_AOT2_one with the help of the **AudioSource** node into the internal buffer of the **AdvancedAudioBuffer** node. After 10 seconds (t=10s) the 'one' will be repeated five times, which can be tested subjectively.

ABv3_ AAB02 This scene is used for testing the reload mode (**LoadMode**=1) by loading ABv3_CU08_1ch_AOT2_one with the help of the AudioSource node into the internal buffer of the AdvancedAudioBuffer node. The clip will be repeated five times with loop=enabled from t=2s until t=7s. After 10 seconds the content will be replaced be clip ABv3_CU08_1ch_AOT2_two, which will be repeated five times from t=12 until t=17. Clip ABv3_CU08_1ch_AOT2_three will be loaded after 20 seconds and repeated five times from t=22 until t=27. The playback should be tested subjectively.

ABv3_ AAB03 This scene is used for testing the accumulate mode (**LoadMode**=2) by loading ABv3_CU8_1ch_AOT2_one at t=0 with the help of the **AudioSource** node into the internal buffer of the **AdvancedAudioBuffer** node. At t=3s ABv3_CU08_1ch_AOT2_two will be loaded into the buffer and at t=6s ABv3_CU08_1ch_AOT2_three will be loaded into the buffer. At t=10s all three clips will be played contiuous ('one, two, three') two times while loop is enabled. The loop mechanism will be tested by setting loop to false at t=19s and start replay at t=20s for 6 seconds. Only one block ('one, two, three') should be heard due to the disabled loop mode. The playback should be tested subjectively.

ABv3_ AAB04 This scene is used for testing the continuous accumulate mode (**LoadMode**=3) by loading ABv3_CU08_1ch_AOT2_one, ABv3_CU08_1ch_AOT2_two and ABv3_CU08_1ch_AOT2_three at t=0, t=3s and t=6s into the internal buffer of the **AdvancedAudioBuffer** node with the help of **AudioSource**. At t=10s all three clips will be played contiuous ('one, two, three') two times while loop is enabled. At t=18s a fourth clip (ABv3_CU08_1ch_AOT2_four) will be appended and with the restriction of **length** = 3 seconds, the first clip shall be deleted. At t=20s all three clips will be played contiuously ('two, three, four') two times. This sequence will be repeated with appending ABv3_CU08_1ch_AOT2_five and playing back ('three, four, five') at t=30s and appending ABv3_CU08_1ch_AOT2_six and playing back ('four, five, six') at t=40s. The playback should be tested subjectively.

ABv3_AAB05 This scene is used for testing the accumulate mode with limited number of buffer blocks of the **AdvancedAudioBuffer** node. In this mode the number of accumulaed blocks will be set instead of a maximum length. First a sequence 'one, two, three' will be loaded and will be played back two times at t=10s. Then 'four' will be loaded into the buffer by replacing 'one', played back two times at t=20s. At t=27s the block in the middle ('three') will be deleted and at t=29 'five' will be appended. This block ('two, four, five') will be played back two times at t=31s.

At t=40s the latest block ('five') is individually addressed and shall be played 3 times while loop is still enabled. At t=43s the block in the middle ('four') will be addressed individually and played back 3 times. At t=46s the first block ('two') will be addressed individually and played back 3 times until t=49s. At t=48,5s **loop** will be set to false. The actual active block will be played until its end.

At t=50s, t=55s and t=60s clip 'four', 'five' and 'two' will be played only one time.

The playback should be tested subjectively.

6.12.4.2 Testing of AudioChannelConfig Node

This node is used to label the audio data in the audio subtree to supply the audio presenter with the required information for multi-channel or soundfield signals. The node has the standard audio node interfaces, but no signal processing capability. The samples are passed through and get new channel configuration information.

6.12.4.2.1 BIFS components needed in the conformance testing

For testing the labelling mechanism of the **AudioChannelConfig** node a minimal set of BIFS nodes is needed: Besides the root node and one grouping node (**Group**, **OrderedGroup**) one top-level AudioBIFS node that has a **spatialize** field (**Sound**, **Sound2D**, **DirectiveSound**) as well as a node that connects the AudioBIFS sub-graph with the decoder (**AudioSource**) is required.

6.12.4.2.2 Conformance testing procedure

Conformance testing of the **AudioChannelConfig** node requires the player to support the parameter printout of the channel configuration that reaches the top-level sound nodes. As the way this information is transported through the AudioBIFS sub-graph is implementation-dependent the channel configuration should be printed in the form of the **AudioChannelConfig** node's fields.

6.12.4.2.2.1 Unique generalChannelFormat

In this subclause the testing of the correct composition of one or more audio signal of the same generalChannelFormat is described, i.e. all sources in the scene are channel-oriented presets (or subsets thereof), parametric channel oriented or Ambisonics™ oriented.

6.12.4.2.2.1.1 Pass through

Test scene for parameter printout and subjective testing:

Abv3_ACC01 This scene is used for testing whether the channel configuration of the audio elementary stream is correctly passed through the AudioBIFS tree. For that purpose a 5.1-channel standard multi-channel configuration sound source will be used. The audio compositor shall recognize the channel configurations of the elementary stream and map the channels to the appropriate speaker(s). The 5.1 configuration shall be recognizable in the *parameter printout* of the channel configuration from the top level node.

For *subjective testing* the CU input used for this scene result in a distinct scheme: On every channel white noise is being played for one second, starting with the front left channel, followed by the front right, surround left, surround right, center channel and finally the LFE channel (5s<t<6s).

ABv3_CU07_6ch_AOT2_surround is used as input sounds for this scene.

6.12.4.2.2.1.2 ChannelPreset

Abv3_ACC02 This scene is used for testing whether an alternative channel configuration stream will be passed correctly through the AudioBIFS tree. Therefore **generalChannelFormat** will be set to the ‘ChannelPreset’ mode. A binaural recorded sequence will be used. After start-up the clip will be marked as ‘normal’ stereo. At t=12s the clip will be played again and marked as ‘binaural’ stereo. The configuration shall be recognizable in the *parameter printout* of the channel configuration from the top level node.

For *subjective testing* the properties of the audio presenter must be taken into account. The presenter might have a crosstalk-canceller for loudspeaker playback or a HRTF cross-convolution for headphone playback. In this case the clip will be heard ‘spatialized’ false during the first 12 seconds and for t>=12s the clip will be heard correctly due to the binaural marked content which should lead to the crosstalk-canceller being enabled for loudspeaker listening or disabled HRTF cross-convolution in the headphone listening case.

ABv3_CU08_1ch_AOT2_LCR_Binaural is used as input sounds for this scene.

6.12.4.2.2.1.3 ChannelPresetSubset

Test scene for parameter printout and subjective testing:

ABv3_ACC03 This scene is used for testing the combination of two stereo and two mono sound sources to a 5.1-channel sound source. The audio compositor shall recognize the channel configurations of the sound sources and map the channels to the appropriate speaker(s). It shall be recognizable in the *parameter printout* of the channel configuration that each top-level sound node addresses an individual subset of a 5.1 configuration (the values of **fixedPreset** are 000110b=6 for the front channels, 011000b=24 for the surround channels, 000001b=1 for the center channel and 100000b=32 for the LFE channel).

For *subjective testing* the CU inputs used for this scene result in a distinct scheme: On every channel white noise is being played for one second, starting with the front left channel, followed by the front right, surround left, surround right, center channel and finally the LFE channel (5s<t<6s).

ABv3_CU01_2ch_AOT2_LR12, ABv3_CU02_2ch_AOT2_LR34, ABv3_CU04_1ch_AOT2_M5 and ABv3_CU05_1ch_AOT2_M6 are used as input sounds for this scene.

ABv3_ACC04 This scene is used for testing the combination of two mono sources to one stereo source. The audio compositor shall recognize the channel configurations of the sound sources and map the channels to the appropriate speaker. It shall be recognizable in the *parameter printout* of the channel configuration that each top-level sound node addresses an individual channel (the values of **fixedPreset** are 01b=1 for the left channel and 10b=2 for the right channel).

For *subjective testing* the CU inputs used for this scene result in a distinct scheme: On the left channel white noise is being played for three seconds, followed 1 second silence and then one second white noise on the right channel.

ABv2_CU06_1ch_AOT2_M0-3 and ABv3_CU04_1ch_AOT2_M5 are used as input sounds for this scene.

Abv3_ACC05 This scene is used for testing whether a dynamic changed channel configuration stream (relabelled channels) will be passed correctly through the AudioBIFS subtree. Therefore **generalChannelFormat** will be set to the ‘ChannelPresetSubset’ mode. A stereo signal (‘applause’) will be used. After start-up the clip will be marked as L,R signal. At t=6s the clip will be marked as LS, RS. The configuration shall be recognizable in the *parameter printout* of the channel configuration from the top level node.

For *subjective testing* the clip should be recognized from the left and right loudspeakers after start-up. After 6 seconds the clip should be heard from the left and right surround loudspeakers.

Abv3_Cu15_2ch_Aot2_Applause is used as input sounds for this scene.

6.12.4.2.2.1.4 ParametricChannelOriented

These scenes are used for testing whether an alternative channel configuration stream will be passed correctly through the AudioBIFS tree. Therefore **generalChannelFormat** will be set to the 'ParametricChannelOriented' mode. The audio presenter shall recognize the loudspeaker positions of the sound sources and map the channels to the appropriate speaker(s). Therefore the ITU 5.1 standard configuration will be addressed (r=2m, 0° (center), +/-30° (right/left), +/-110° (right/left surround) and +15°azimuth/-21°elevation (LFE)). At t=12s the configuration will be flipped to (0°, -/+30°, -/+110° and +15°azimuth/-21°elevation). The configuration shall be recognizable in the *parameter printout* of the channel configuration from the top level node.

For *subjective testing* the, the left/right and left/right surround channels will be swapped at t=12s.

ABv3_CU7_6ch_AOT2_surround is used as input sounds for scene Abv3_ACC06a … Abv3_ACC07c and Abv3_ACC09.

Abv3_ACC06a	origin = scene origin, Cartesian corrdinate system
Abv3_ACC06b	origin = scene origin, Polar coordinate system
Abv3_ACC06c	origin = scene origin, Cylindric coordinate system
Abv3_ACC07a	origin = user position, Cartesian corrdinate system
Abv3_ACC07b	origin = user position, Polar coordinate system
Abv3_ACC07c	origin = user position, Cylindric coordinate system
Abv3_ACC09	[C, Left, Right, Left Surround, Right Surround, LFE] = [scene origin: Cartesian, Polar, Cylindric, user position: Cartesian, Polar, Cylindric]

6.12.4.2.2.1.5 ParametricChannelOriented with directional Loudspeaker

Abv3_ACC10 This scenes is used for testing loudspeaker directional parameters. The **generalChannelFormat** will be set to the 'ParametricChannelOriented' mode. The audio presenter shall recognize the loudspeaker positions of the sound sources and map the channels to the appropriate speakers. A 4-channel subset of the ITU 5.1 standard configuration with only one surround channel will be addressed (r=2m, 0° (center), +/-30° (right/left), -180° (dipole surround)), similar to a Dolby ProLogic® configuration. The surround channel will be set to dipole characteristic with **channelDirectionalPattern** = 1. The configuration shall be recognizable in the *parameter printout* of the channel configuration from the top level node.

For *subjective testing*, the surround channel should give a surround feeling, due to the missing lobe from the surround loudspeaker.

origin = user position, Cartesian corrdinate system; Abv3_CU26_2ch_AOT2_sea-wash_cl and Abv3_CU26_2ch_AOT2_seawash_rs are used as input sounds for the scene. The two 2-channel streams are combined to a 4-channel signal, whereby the 4th channel the surround channel is.

6.12.4.2.2.1.6 ParametricAmbisonicsOriented

ABv3_ACC11 This scene is used for testing the gathering of two stereo and two mono streams into a multi-channel audio flow labelled as a Higher Order Ambisonic sound field. In this scene, the input ambisonic channels are in the default order (W, X, Y, Z, U, V) regarding the specification of the spatial resolution (**ambResolution2D** and **ambResolution3D**) and therefore they follow an implicit indexing. The audio compositor shall recognize the channel configuration of the output multi-channel flow. It shall be recognizable in the *parameter printout* of the channel configuration that the top-level **SurroundingSound** sound node addresses an Ambisonic sound field with a 2nd order 2D resolution (*i.e.* regarding the horizontal plane) and a 1st order 3D resolution (*i.e.* regarding the vertical dimension):

```
SurroundingSound: generalChannelFormat 4

SurroundingSound: ambResolution2D 2

SurroundingSound: ambResolution3D 1

SurroundingSound: ambEncodingConvention 2
```

For *subjective testing*: The 3D audio compositor should take into account the channel configuration of the output multi-channel flow in the most appropriate way to process its spatial decoding and render the sound field according to the reproduction setup (a loudspeaker setup or headphones). The listener should perceive the same spatial scene organization as described when presenting the Composition Unit ABv3_CU19_1ch_AOT2_W in 6.12.2

ABv3_CU19_1ch_AOT2_W, ABv3_CU20_2ch_AOT2_XY, ABv3_CU21_1ch_AOT2_Z and ABv3_CU22_2ch_AOT2_UV are used as input sounds for this scene.

ABv3_ACC12 This scene is similar to ABv3_ACC11 and uses the same input sounds except that it changes their order by permuting two children of the **AudioChannelConfig** node. Therefore an explicit indexing of Ambisonic channels is necessary and done by using the **ambComponentIndex** field. The parameter printout should make appear the following lines:

```
SurroundingSound: generalChannelFormat 4

SurroundingSound: ambResolution2D 2

SurroundingSound: ambResolution3D 1

SurroundingSound: ambEncodingConvention 2

SurroundingSound: ambArrangementRule 1

SurroundingSound: ambComponentIndex [0 1 2 4 5 3]
```

For subjective testing: from the listener's point of view, the rendered sound field should sound the same as with scene ABv3_ACC11.

For objective testing: output signals delivered by the audio compositor for restitution over loudspeakers or headphones should be the same as with scene ABv3_ACC11.

ABv3_ACC13 This scene has similarities with ABv3_ACC12 except that the first input flows (representing Ambisonic channels W, X, Y, U and V) are replaced with input streams forming a 5.0 content that can be reconverted into the Ambisonic channels W, X, Y, U and V of scenes ABv3_ACC11 (and ABv3_ACC12). This conversion is done by applying the matrix described by **ambBackwardMatrix** to the input channels. Moreover the **ambComponentIndex** field is used for explicit indexing of resulting Ambisonic channels.

The parameter printout should make appear the following lines:

```
SurroundingSound: ambResolution2D 2

SurroundingSound: ambResolution3D 1

SurroundingSound: ambEncodingConvention 2

SurroundingSound: ambArrangementRule 7

SurroundingSound: ambComponentIndex [0 1 2 4 5 3]

SurroundingSound:  ambBackWardMatrix  [6  6  0.1535  0.9007  0.9007
0.6796  0.6796  0  0.5075  1.4697  1.4697  -1.0610  -1.0610  0  0
0.5580 -0.5580 2.0493 -2.0493 0 8.4222 -4.1298 -4.1298    0.8123
0.8123 0 0 4.0084 -4.0084 -3.5519 3.5519 0 0 0 0         0 0 1]
```

For subjective testing: from the listener's point of view, the rendered sound field should sound substantially the same as with scene ABv3_ACC11 in terms of spatial organization.

ABv3_CU23_1ch_AOT2_ambITU_C, ABv3_CU24_2ch_AOT2_ambITU_LR, ABv3_CU25_2ch_AOT2_ambITU_SLSR and ABv3_CU21_1ch_AOT2_Z are used as input sounds for this scene.

6.12.4.2.2.2 Combination of different formats

To test the ability of the audio compositor to render several multi-channel formats in the same scene and at the same time, use the scene ABv3_SS04, which combines the same ambisonic sound field as used in scene ABv3_ACC11 and the same 5.1 sound field as used in scene ABv3_ACC01.

6.12.4.3 Testing of SurroundingSound Node

The **SurroundingSound** node is used to attach sound to a scene. This causes spatial qualities and makes it related to the visual content of the scene. This includes multi-channel signals that cannot be spatially transformed with the other sound nodes due to their restrictions to the specification of the **phaseGroup** field and the **spatialize** field. By using this node, sound may be attached to a group and spatialized or moved around as appropriate for the spatial transforms above the node.

6.12.4.3.1 BIFS components needed in the conformance testing

Except from basic 3D nodes (**Group**, **Transform**, **ViewPoint**) the **AdvancedAudioBuffer** node is needed for looped playback of the audio data. For subjective testing the scene needs to be animated. The nodes **TimeSensor**, **CoordinateInterpolator** and **OrientationInterpolator** are required as well as the ROUTE mechanism. To ensure the correct channel configuration an **AudioChannelConfig** node is used in the test scenes.

6.12.4.3.2 Conformance testing procedure

Conformance testing of the **SurroundingSound** node requires the player to support the parameter printout of **SurroundingSound** node's fields. For subjective testing the listener has to check if playing back the scene has the described effect. The subjective listening tests shall be performed.

Test Scenes:

ABv3_SS01

This scene is used for subjective testing of the **SurroundingSound** node. This is done by rotating a 5.1 audio track around the listener by modifying the **orientation** field of the node **SurroundingSound** via **TimeSensor** and **OrientationInterpolator** nodes. The listener shall perceive the clockwise rotation of the surround panorama.

ABv3_CU07_6ch_AOT2_surround is used as input sound for this scene.

ABv3_SS01b

Same as ABv3_SS01, except that ABv3_CU23_1ch_AOT2_ambITU_C, ABv3_CU24_2ch_AOT2_ambITU_LR, ABv3_CU25_2ch_AOT2_ambITU_SLSR and ABv3_CU19_1ch_AOT1_W are used as input sounds for this scene (recording of a natural sound scene).

ABv3_SS02

This scene is used for subjective testing of the effect of outer transformations on the **SurroundingSound** node. This is done by rotating a 5.1 audio track around the listener by modifying the **rotation** field of an "outer" **Transform** node via **TimeSensor** and **OrientationInterpolator** nodes. The listener shall perceive the clockwise rotation of the surround panorama (similarly as with scene ABv3_SS01) as long as the **isTransformable** field of the **SurroundSound** node is set to TRUE, *i.e.* during the first 8 seconds. After this delay the **isTransformable** field is set to FALSE so that the **rotation** field of the **Transform** node shall have no longer effect on the **SurroundingSound** node and the listener shall perceive a static surround panorama.

ABv3_CU07_6ch_AOT2_surround is used as input sound for this scene.

ABv3_SS02b

Same as ABv3_SS02, except that ABv3_CU23_1ch_AOT2_ambITU_C, ABv3_CU24_2ch_AOT2_ambITU_LR, ABv3_CU25_2ch_AOT2_ambITU_SLSR and ABv3_CU19_1ch_AOT1_W are used as input sounds for this scene (recording of a natural sound scene).

ABv3_SS04

This scene is used for subjective testing of the distance attenuation, *i.e.* the decreasing (resp. increasing) of the perceived surrounding sound field level as the **ListeningPoint** moves away from (resp. gets closer to) its centre, as defined by the **location** field of the **SurroundingSound** node. In this scene, two sound fields (and **SurroundingSound** instances) are present and centered on distinct locations. The **distance** parameter of the two **SurroundingSound** instances is chosen so that both sound fields are audible with a non negligible level half-way between the two locations. As **ListeningPoint** moves from one location to the other (during the first 10s), the listener shall first perceive the first surrounding sound field (nature ambience) with a significantly higher level than the second one, then a mix between both sound fields and finally mostly the second sound field (test signals). Then **ListeningPoint** moves back to its initial position so that the listener shall perceive the inverse effect (during the following 10 s). The kind of cross-fade effect implied by the distance attenuation should be as smooth as possible. For the next rounds the distance attenuation is deactivated for the first **SurroundingSound** instance (its **distance** parameter is set to 0) so that only the second sound field shall be perceived with a varying level as **ListeningPoint** continues its back-and-forth trajectory.

The sound field already used in scenes ABv3_AAC11 (2nd order ambisonic recording of nature ambience) and ABv3_AAC01 (5.1 test signals) are combined and respectively centered on the first and second location points. Therefore the present scene is also used for subjective testing of the combination of several formats.

ABv3_SS05

This scene is used for subjective testing of the effect the angular distortion of the sound scene as the **ListeningPoint** (or **ViewPoint**) moves in the virtual scene and when the **distortionFactor** is not null. It is also used to test the ability of the 3D audio compositor to appropriately combine several kinds of sound field transformations, namely angular distortion and rotation.

The same sound field as ABv3_AAC11 is used (ABv3_CU19_1ch_AOT2_W, ABv3_CU20_2ch_AOT2_XY, ABv3_CU21_1ch_AOT2_Z and ABv3_CU22_2ch_AOT2_UV are also used as input sounds). The sound field extract is 14.3 second long and played in loop using an **AdvancedAudioBuffer** node. The scene is divided in 5 sequences that correspond to several kinds of **ListeningPoint** movements and induced sound field transformations. The **distortionFactor** is set to 0.2, which corresponds to a reference distance of 5 meters for sounds positioned orthogonally to the **ListeningPoint**'s movement.

1st step (0→14.3 sec.): **ListeningPoint** is static, positioned and oriented according to the default parameter values. As the sound field contains mostly static sources, the listener shall be able to memorize its spatial organization.

2nd step (14.3→28.6 sec.): **ListeningPoint** rotates by half-turn during the first 7.15s. In the first 7.15s the listener should perceive that the sound scene rotates without changing the relative angles between the sounds of the scene, nor their level. In the last 7.15s the listener shall perceive that the front-back and left-right axes have been inverted with respect to the original orientation.

3rd step (28.6→42.9 sec.): **ListeningPoint** moves 5 meters towards its new frontal direction. The listener shall perceive that the front part of the scene progressively enlarges while the lateral sources move backward and the back part of the scene narrows. Changes should be rendered as smoothly as possible.

4th step (42.9→57.2 sec.): **ListeningPoint**'s movement combines half-turn rotation and diagonal trajectory so that its final position is 5m left to the initial position and its final orientation is the original one. The listener should feel both rotation and angular distortion effects. Changes should be rendered as smoothly as possible and without audible click.

5th step (57.2→71.5 sec.): During the first 7.15 s **ListeningPoint** moves to its right (like a crab) to get its initial position. The listener should perceive that the front and back sound sources move to left. He/she should finally retrieve the initial spatial organization.

6.12.4.4 Testing of Transform3DAudio Node

The **Transform3DAudio** node is used to integrate 3D audio in 2D visual scenes. Therefore coordinate transformations can be applied so that movements in the x-y (visual) plane are transformed into movements in the x-z plane for audio (which is the default behaviour of this node). Additionally it allows adding coordinates and vectors to the incoming two-dimensional transform hierarchy to achieve a full 3D addressing.

6.12.4.4.1 BIFS components needed in the conformance testing

For testing the **Transform3DAudio** node a minimal set of BIFS nodes is needed: Besides the root node and a node for 2D grouping (**OrderedGroup**) and transformations (**Transform2D**) one top-level 3D AudioBIFS node having a **spatialize** field (**Sound**, **DirectiveSound**) as well as a node that connects the AudioBIFS sub-graph with the decoder (**AudioSource**) is required.

6.12.4.4.2 Conformance testing procedure

Conformance testing of the **Transform3DAudio** node requires the player to support the parameter printout of the location and orientation of the respective **Sound** node in world coordinates (`globalPosition` and orientation `globalOrientation`). Note that the transformation produced by the **location** field of the **Sound** node also shall be taken into account for parameter printout. With this method the default functionality (flipping coordinate system), the coordinate transformation and the adding of coordinates and vectors to 2D transformations can be tested. For subjective testing the listener has to check if playing back the scene has the described effect.

Test Scenes:

ABv3_T3DA01 This scene is used for testing the default functionality of the **Transform3DAudio** node (mapping the x-y position into the x-z plane) by modifying the **translation** field of an "outer" **Transform2D** node via BIFS updates, which causes the sound source to move in the x-z plane. In the *parameter printout* it shall be visible that the global position in 3D world coordinates of the **Sound** node are initially (-1,0,0), and after each BIFS update (each 500ms) (0.707,0,-0.707), (1,0,0), (0.707,0,0.707) and (0,0,1):

```
Sound: globalPosition -1.0000 0.0000 0.0000
Sound: globalPosition 0.7071 0.0000 -0.7071
Sound: globalPosition 1.0000 0.0000 0.0000
Sound: globalPosition 0.7071 0.0900 0.7071
Sound: globalPosition 0.0000 0.0000 1.0000
```

For *subjective testing* the sound source shall describe a rough semicircle from its initial position right in front of the listener over the right side up to the back of the listener remaining in the x-z plane (y=0).

ABv3_T3DA02 This scene is used for testing the extension of a 2D rotation to a 3D rotation by adding a rotation vector (0,0,-1) (field **rotationVector**) to the rotation angle obtained from the 2D transform hierarchy in the **Transform3DAudio** node. The **rotationAngle** field of an "outer" **Transform2D** node is modified by BIFS updates, which cause the sound source to move in the x-z plane from the front of the listener to his right side (after the default coordinate transformation of the **Transform3DAudio** node is applied). In the parameter printout this movement shall be visible; the absolute position of the sound node changes each 500ms:

```
Sound: globalPosition 0.0000 0.0000 -1.0000
Sound: globalPosition 0.3827 0.0000 -0.9239
Sound: globalPosition 0.7071 0.0000 -0.7071
Sound: globalPosition 0.9239 0.0000 -0.3827
Sound: globalPosition 1.0000 0.0000 -0.0000
```

ABv3_T3DA03 This scene is used for testing the extension of a 2D rotation center to a 3D rotation center by adding a 3rd center coordinate 1 (field **thirdCenterCoordinate**) to the 2D rotation center (1,0) obtained from the 2D transform hierarchy in the **Transform3DAudio** node. The **rotationAngle** field of an "outer" **Transform2D** node is modified by BIFS updates, which cause the sound source to rotate around the transformed rotation center (after the default coordinate transformation of the **Transform3DAudio** node is applied). Additionally the **rotationVector** is (0,0,-1). In the parameter printout this circular movement shall be visible; the absolute position of the sound node changes each 375ms:

```
Sound: globalPosition 0.0000 1.0000 0.0000
Sound: globalPosition 1.0000 1.4142 0.0000
Sound: globalPosition 2.0000 1.0000 0.0000
Sound: globalPosition 2.4142 0.0000 0.0000
Sound: globalPosition 2.0000 -1.0000 0.0000
```

```
Sound: globalPosition 1.0000 -1.4142 0.0000
Sound: globalPosition 0.0000 -1.0000 0.0000
Sound: globalPosition -0.4142 0.0000 0.0000
Sound: globalPosition 0.0000 1.0000 0.0000
```

ABv3_T3DA04 This scene is used for testing the extension of a 2D scaling to a 3D scaling by adding a 3rd scale coordinate 3 (field **thirdScaleCoordinate**) to the 2D scaling (1,0) obtained from the 2D transform hierarchy in the **Transform3DAudio** node. After the default coordinate transformation of the **Transform3DAudio** node is applied the **location** vector of the **Sound** node (1,1,1) shall be (2,3,-2), which can be seen in the parameter printout (no BIFS updates are used, so only one value will be shown):

```
Sound: globalPosition 2.0000 3.0000 -2.0000
```

ABv3_T3DA05 This scene is used for testing the extension of an angular 2D scale orientation to a 3D scale orientation by adding the vector (0,1,0) (field **scaleOrientationVector**) to the 2D scale orientation obtained from the 2D transform hierarchy in the **Transform3DAudio** node. The **scaleOrientation** field of an "outer" **Transform2D** node is modified by BIFS updates, which cause the sound source to move in a plane parallel to the xy-plane (after the default coordinate transformation of the **Transform3DAudio** node is applied). A scaling of (2,2,3) ((2,2) from the 2D transform hierarchy and 3 as **thirdScaleCoordinate**) is applied in order to make the scale orientation effective. In the parameter printout this circular movement shall be visible; the absolute position of the sound node changes each 750ms:

```
Sound: globalPostion 2.0000 3.0000 -2.0000
Sound: globalPostion 3.0000 3.0000 -2.0000
Sound: globalPostion 3.0000 2.0000 -2.0000
Sound: globalPostion 2.0000 2.0000 -2.0000
Sound: globalPostion 2.0000 3.0000 -2.0000
```

ABv3_T3DA06 This scene is used for testing the extension of a 2D translation to a 3D translation by adding a 3rd translation coordinate (field **thirdTranslationCoordinate**) to the translation (1,1) obtained from the 2D transform hierarchy in the **Transform3DAudio** node. The **thirdTranslationCoordinate** field of the **Transform3DAudio** node is modified by BIFS updates, which cause the sound source to move from (1,0,-1) to (1,1,-1) (after the default coordinate transformation of the **Transform3DAudio** node is applied). In the parameter printout this movement shall be visible; the absolute position of the sound node changes each 500ms:

```
Sound: globalPosition 1.0000 0.0000 -1.0000
Sound: globalPosition 1.0000 0.2000 -1.0000
Sound: globalPosition 1.0000 0.4000 -1.0000
Sound: globalPosition 1.0000 0.6000 -1.0000
Sound: globalPosition 1.0000 0.8000 -1.0000
Sound: globalPosition 1.0000 1.0000 -1.0000
```

ABv3_T3DA07 This scene is used for testing the coordinate transform functionality of the **Transform3DAudio** node by modifying the axis of its **coordinateTransform** field via BIFS updates each 500ms, which causes the sound source to move. In the *parameter printout* these position changes shall be visible:

```
Sound: globalPosition 0.0000 0.0000 1.0000
Sound: globalPosition -0.0073 0.0370 0.9993
Sound: globalPosition -0.0451 0.1212 0.9916
Sound: globalPosition -0.1087 0.2093 0.9718
Sound: globalPosition -0.1789 0.2807 0.9430
Sound: globalPosition -0.2440 0.3333 0.9107
```

ABv3_T3DA08 This scene is used for testing the coordinate transform functionality of the **Transform3DAudio** node by modifying the angle of its **coordinateTransform** field via one BIFS update, which causes the sound source to flip from the xz-plane to the xy-plane. In the *parameter printout* this position change shall be visible:

```
Sound: globalPosition 0.0000 0.0000 -1.0000
Sound: globalPosition 0.0000 1.0000 0.0000
```

For all scenes ABv2_CU06_1ch_AOT2_M0-3 is used as input sound.

6.12.4.5 Testing of WideSound Node

The **WideSound** node is used to attach sound to a scene, thereby giving it spatial qualities with a determinable widening for non-phase-related signals from its descendant audio nodes and relating it to the visual content of the scene. Additionally the node has backward compatible functions of the **Sound** and **DirectiveSound** nodes which will not be tested here.

6.12.4.5.1 BIFS components needed in the conformance testing

For testing the **WideSound** node a minimal set of BIFS nodes is needed: Besides the root node and one grouping node (**Group**, **OrderedGroup**) one a node that connects the AudioBIFS sub-graph with the decoder (**AudioSource**) is required.

6.12.4.5.2 Conformance testing procedure

Conformance testing of the **WideSound** node requires the player to support the parameter printout of **WideSound** node's fields. For subjective testing the listener has to check if playing back the scene has the described effect.

Four different shapes of a sound can be selected: shuck, box, ellipsoid and cylinder. The printout of the nodes gives a functional hint. But the real function of the node can be tested subjectively only. Therefore it is a common practice to play back the same sound (here the four shapes) with different behaviour consecutively without disturbing noise. The subjective listening tests shall be performed.

6.12.4.5.2.1 Testing of all shapes

Test Scenes:

ABv3_WS01 This scene is used for testing the different shapes of the **WideSound** node, starting with a stereo version of the clip: ABv3_CU14_2ch_AOT2_applause. At t=15s, t=30s, t=45s and t=60s the clip ABv3_CU15_1ch_AOT2_applause_mono will be used with the four shapes consecutively shuck, box, ellipsoid and cylinder. The size of the shape is {2m,1m,1m} for the box, ellipsoid and cylinder in the location {0, 2, 0} and half/quarter sphere for the shuck.

ABv3_WS02 This scene is used for testing the different shapes of the **WideSound** node, starting with a stereo version of the clip: ABv3_CU17_2ch_AOT2_orchestra. At t=15s, t=30s, t=45s and t=60s the clip ABv3_CU18_1ch_AOT2_orchestra_mono will be used with the four shapes consecutively shuck, box, ellipsoid and cylinder. The size of the shape is {2m,1m,1m} for the box, ellipsoid and cylinder in the location {0, 2, 0} and half/quarter sphere for the shuck.

6.12.4.5.2.2 Testing of signalling plane waves

ABv3_WS03

This scene is used for testing the signalling of plane waves with the **WideSound** node. Therefore a source emanating two-dimensional plane waves is positioned in front of the listener. If the playback system is capable of rendering plane waves the sound source shall be perceived in infinite distance (meaning always from the front, independent from translation in x-z plane, *subjective testing*). After three seconds the source is switched to emanate three-dimensional plane waves. The listener shall perceive the source in infinite distance regardless of any translation.

ABv3_CU18_1ch_AOT2_orchestra_mono is used as sound input for this scene.

6.12.4.6 Testing of AudioFXProto mechanism

The **AudioFxPROTO** node provides an implementation of a tailored subset of functionality available through the **AudioFX** node. The **AudioFX** node normally requires a Structured Audio implementation. The tailored subset allows players without Structured Audio capability to use these standard audio effects.

The intended effect of the **AudioFxPROTO** node is not normative, so that the effect can be tested subjectively only. The **AudioFxPROTO** node interface is normative. The parameters shall be verified with the corresponding scene by a printout of the **audioFXParams** field.

6.12.4.6.1 BIFS components needed in the conformance testing

For testing the **AudioFXProto** with the **AudioFX** node a minimal set of BIFS nodes is needed: Besides the root node and one grouping node (**Group**, **OrderedGroup**) one a node that connects the AudioBIFS sub-graph with the decoder (**AudioSource**) is required. The effect can be implemented in a loadable module, addressed by the PROTO's name instead of a full **AudioFX** implementation.

6.12.4.6.2 Conformance testing procedure

Conformance testing of the **AudioFXProto** with the **AudioFX** node requires the player to support the parameter printout of the **audioFXParams** field. For subjective testing the listener has to check if playing back the scene has the described effect. The effects are tested with Abv3_CU17_1ch_AOT2_orchestra or Abv3_CU18_1ch_AOT2_orchestra_mono, depending on the effect. The effect parameters are changed during the playback of the clip.

6.12.4.6.2.1 Testing of audioChorus

Abv3_aFXP_aCh01

This scene is used for testing the **audioChorus** effect. At start-up the effect is disabled (bypass of the clip). At t=6s the parameters inside the **audioFXParams** field are set to:

`rate=2.0, depth=0.5` and `modulation=1.0` (Triangle modulation)

which addresses a moderate effect strength.

ABv3_CU18_1ch_AOT2_orchestra_mono is used as input for this scene. The output is a 2-channel signal.