VHT Audiospace

User Guide

v1.0

virtualhometheater.com

Table of Contents

Introduction

Audiospace is a companion file for a media file that will specify how the audio channels should be used in virtual space as virtual audio objects for binaural audio rendering output.

This audiospace file will be used by Virtual Home Theater (VHT) VR app when a media file with the same name will be played.

By using an audiospace file you can route any audio channel to any virtual speaker in virtual space either as static audio objects (position is fixed in space) or dynamic audio objects (position change with time).

This can be used to:

- play an audio file using any user defined layout (e.g. play a 16 channels audio file using the 9.1.6 speakers layout)
- play a VR image plus audio with virtual static audio objects simulating the actor voice or object sound location in space
- play a VR video with virtual dynamic audio objects simulating the actor voice or object sound location moving through space

Creating Audiospace File

The audiospace file can be created manually by user or by a software.

Dimensions are in meters, time in seconds and angles in degree.

Channel IDs and track IDs start from zero.

The number of audiospace channels must be equal to the number of audio channels from the media file audio track.

For complete names list of virtual speakers read Acronyms chapter.

By Software

“VHT tools for DAZ” render script can generate audiospace files.

Please visit VHT web page for more info.

When rendering an image it will generate a STATIC_AUDIO_OBJECTS audiospace file if audio objects were added to the scene.

When rendering an image series or a video it will generate a DYNAMIC_AUDIO_OBJECTS audiospace file if audio objects were added to the scene.

By User

This is the case when you want to have a different virtual speakers layout than the standard ones.

First create a new text file on the same folder with the media file.

Rename the text file using this format: “NAME.audiospace” where the NAME is the exact name of the media file (w/o file extension) and “.audiospace” is the file extension.

Open and edit the newly created audiospace companion file.

Add as the first line the following header:

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1
```

Add the audio track ID for which the following settings will apply:

```
track=0
```

Specify the name of the audiospace preset for this audio track by adding the following line:

```
name=type here the name
```

Next you need to specify one of the three audiospace modes that will be used: *custom virtual speakers, static audio objects or dynamic audio objects*.

Custom Virtual Speakers Mode

This is the mode that it is designed to be created by user and will use a left-to-right symmetric layout.

Add the following line:

```
mode=CUSTOM_VIRTUAL_SPEAKERS
```

Next you need to specify one of the three speaker layouts that will be used: *sphere*, *square and vector*.

Sphere Layout

Use this symmetric layout when you want to place all speakers on a sphere.

Add the following lines:

```
layout=SPHERE
sphere_radius=4
front_angle=30
side_angle=90
back_angle=135
top_angle=30
```

Square Layout

Use this symmetric layout when you want to place all speakers on the corners and edges of a box.

Add the following lines:

```
layout=SQUARE
square_size=8
top_height=2
```

In the next and final step common to **sphere and square layouts** you will define the channel-to-speaker routing and audio gain for each virtual speaker like this example:

```
channel[0]*dB(0)=LFE
channel[1]*dB(1.2)=SL
channel[2]*dB(-3.1)=FR
channel[3]*dB(4)=SR
channel[4]*dB(0)=FL
channel[5]*dB(0)=FC
```

This will play channel 0 to LFE virtual speaker with zero dB gain, channel 1 to SL virtual speaker with 1.2 dB gain, channel 2 to FR virtual speaker with -3.1 dB gain and so on.

Vector Layout

Use this asymmetric layout when you want custom and precise Cartesian/polar coordinate position for each virtual speaker. This is the most flexible layout for virtual speakers customization.

Add the following line:

```
layout=VECTOR
```

Optionally you can specify the 3D coordinate system used for the following inputs by adding one of the following flag line:

`LEFT_HANDED` or `RIGHT_HANDED`

Optionally you can add the following translation offset for all speakers:

```
offset=xyz(-1.2,3,4.55)
```

Optionally you can add one of the following two scaling methods for all speakers:

`scale=1.5` or `scale=xyz(0.04,0.01,2)`

1st will scale all dimensions with 1.5

2nd will scale all X components with 0.04, all Y components with 0.01 and all Z components with 2.

Optionally you can add a general signal audio amplification gain:

```
dB=6.2
```

this is an additive parameters to the gain from each channel.

In the next and final step you will define the channel-to-speaker routing, audio gain for each virtual speaker and the Cartesian/polar position of each speaker in virtual space like this example:

```
channel[0]*dB(0)->xyz(0,0,200)=LFE
channel[1]*dB(1.2)->xyz(200,0,200)=SL
channel[2]*dB(-3)->xyz(100,0,200)=FR
channel[3]*dB(0)->xyz(100,-100,200)=SR
channel[4]*dB(0)->xyz(0,0,1.12)=FL
channel[5]*dB(0)→xyz(2.21,0,100)=FC
```

This will play:

- channel 0 to LFE speaker positioned at [(xyz(0,0,200)+offset)*scale] with zero dB gain
- channel 1 to SL speaker positioned at [(xyz(200,0,200)+offset)*scale] with 1.2 dB gain
- channel 2 to FR speaker positioned at [(xyz(100,0,200)+offset)*scale] with -3 dB gain

and so on.

Static Audio Objects Mode

Normally this mode is generated by a software but can also be created or edited by a user.

Use this for a static audio playback (a video file with a single non-changing image) where the sound from audio objects is correlated with the image.

Add the following line:

```
mode=STATIC_AUDIO_OBJECTS
```

Optionally you can also use the following lines as described in the previous Vector Layout chapter like:

```
LEFT_HANDED or RIGHT_HANDED
offset=xyz(-1.2,3,4.55)
scale=1.5 or scale=xyz(0.04,0.01,2)
dB=6.2
```

The following examples lines are four named audio objects positioned in virtual space:

```
channel[0]*dB(0)->xyz(2,0.5,3.5):Henry voice
channel[1]*dB(-2)->xyz(-2,0.4,4):Emma voice
channel[2]*dB(0)->xyz(0,-0.3,4):Guitar
channel[3]*dB(0)->xyz(1,-0.1,5):Drums
```

This will play:

- channel 0 to audio object #0 positioned at [(xyz(2,0.5,3.5)+offset)*scale] with zero dB gain and named as “Henry voice”
- channel 1 to audio object #1 positioned at [(xyz(-2,0.4,4)+offset)*scale] with -2 dB gain and named as “Emma voice”

and so on.

Dynamic Audio Objects Mode

Normally this mode is generated by a software but can also be created or edited by a user.

Use this for a dynamic audio playback (for videos) where objects (car, person, airplane) are moving and the sound from audio objects is correlated with the moving picture.

All audio objects positions and gains are linearly interpolated and updated in virtual space by VHT at the rate of HMD refresh rate (e.g. 90 Hz or more).

Add the following line:

```
mode=DYNAMIC_AUDIO_OBJECTS
```

Optionally you can also use the following lines as described in the previous Vector Layout chapter like:

```
LEFT_HANDED or RIGHT_HANDED
offset=xyz(-1.2,3,4.55)
scale=1.5 or scale=xyz(0.04,0.01,2)
dB=6.2
```

Example 1:

```
channel[0]@T+(0)*dB(0)->xyz(-20,0,3):bird
channel[0]@T+(10)*dB(0)->xyz(20,0,3):bird
```

this is one audio object moving from left to right in 10 seconds.

Example 2:

```
channel[0]@T+(0)*dB(0)->xyz(-20,0,3):bird
channel[0]@T+(10)*dB(0)->xyz(20,0,3):bird
channel[1]@T+(0)*dB(0)->xyz(0,10,3):ball
channel[1]@T+(10)*dB(0)→xyz(0,-1.2,3):ball
```

here are two moving audio objects, one moving from left to right and another from top to floor level in 10 seconds.

Example 3:

```
channel[0]@T+(0)*dB(0)->xyz(-4,0,4):p1
channel[0]@T+(7.5)*dB(0)->xyz(4,0,4):p2
channel[0]@T+(15)*dB(0)->xyz(4,0,-4):p3
channel[0]@T+(22.5)*dB(0)->xyz(-4,0,-4):p4
channel[0]@T+(30)*dB(0)->xyz(-4,0,4):p1
```

this is one audio object moving on a square in 30 seconds.

Where `@T+()` is the playback time in seconds at which this line is valid (presentation timestamp).

Direct Playback

You can route any audio channel directly to the headphone speakers without any 3D spatial binaural audio rendering.

These lines are added only by user when he want a special audio effect:

`->HEADPHONES` will route any channel directly to the headphone speakers.

`->LEFT_HEADPHONE` will route any channel directly to the left headphone speaker.

`->RIGHT_HEADPHONE` will route any channel directly to the right headphone speaker.

This direct channels routing are added once and are valid for the entire audio playback duration and can't be changed.

Direct channel routing examples for custom virtual speakers.

A user can change any channel like these (for binaural playback):

```
channel[0]*dB(0)=FL
channel[1]*dB(0)=FR
channel[2]*dB(0)=LFE
```

into this (for direct playback):

```
channel[0]*dB(0)->LEFT_HEADPHONE=FL
channel[1]*dB(0)->RIGHT_HEADPHONE=FR
channel[2]*dB(0)->HEADPHONES=LFE
```

Direct channel routing examples for static or dynamic audio objects.

For example: a user can add additional audio channels (8, 9 and 10) to the already software generated 0 to 7 channels:

```
channel[0]@T+(0)*dB(0)→xyz(-4,0,4):p1
...
channel[7]@T+(0)*dB(0)->xyz(-0,0,0):p2
channel[7]@T+(30)*dB(0)->xyz(-4,0,4):p2
channel[8]*dB(0)->HEADPHONES:Hall reverb
channel[9]*dB(0)->LEFT_HEADPHONE:Coiled Tube Earpiece
channel[10]*dB(0)->RIGHT_HEADPHONE:phone talk
```

notice the missing presentation timestamp `@T+()` for channels 8, 9 and 10.

Miscellaneous

You can deactivate any line by using “//” at the beginning of a line:

```
//channel[1]*dB(1.2)->xyz(200,0,200)=FL
```

You can add comments like this:

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1
This is a comment.
track=0
...
This is another comment.
track=1
...
```

For imperial input distances (foot) use a scale factor of 0.305 like:

```
scale=0.305
```

If you have multiple audio tracks (e.g English, Japanese, etc.) that use the same settings you can skip writing the same preset again for the other tracks by using:

```
track[1]=track[0]
track[2]=track[0]
```

Instead of Cartesian coordinate input position like `xyz(-4,2,4)` you can use:

Cylindrical coordinate system (polar + height)

`pol(`*`azimuth`*`,`*`radius`*`,`*`height`*`)`

where:

- **azimuth** [degrees] is the horizontal angle between front direction to audio object (clockwise)
- **radius** [meter] is from listener vertical to the audio object
- **height** [meters] is the same like the Z coordinate from right-handed Cartesian system or like the Y coordinate from left-handed Cartesian system

Spherical coordinate system

`sph(`*`azimuth`*`,`*`distance`*`,`*`elevation`*`)`

where:

- **azimuth** [degrees] is the horizontal angle between front direction to audio object (clockwise)
- **distance** [meters] is from listener to the audio object
- **elevation** [degrees] is the vertical angle between horizontal plane to audio object (upward)

The center of input coordinate system is at the head level, so a `xyz(0,0,0)` will position an audio object between your ears not on the floor!.

Limitations

Because the human brain at one time can spatially localize only 16 discrete audio sources the maximum number of virtual audio objects is set to be 16.

Default audiospace reference layouts

You can use a default audiospace layout preset without creating one if you add the following keywords to the media file name.

Using R-REC-BS.2051 standard:

- **5.1+2** for 8 discrete audio channels
- **5.1+4** for 10 discrete audio channels
- **7.2+3** for 12 discrete audio channels
- **7.1+4** for 12 discrete audio channels

Using Dolby recommendation:

- **9.1.6** for 16 discrete audio channels
- **9.1.4** for 14 discrete audio channels
- **9.1.2** for 12 discrete audio channels
- **7.1.6** for 14 discrete audio channels
- **7.1.4** for 12 discrete audio channels
- **7.1.2** for 10 discrete audio channels
- **7.0.2** for 9 discrete audio channels
- **5.1.4** for 10 discrete audio channels
- **5.1.2** for 8 discrete audio channels
- **4.1.4** for 9 discrete audio channels

For example, if a media file name is “movie with 7.1.4 audio.mkv” and the audio track has 12 discrete and available audio channels then VHT will use the reference 7.1.4 default audiospace from the following VHT installation directory:

C:\Program Files (x86)\Steam\steamapps\common\Virtual Home Theater\assets\audiospace\

These reference audiospace files can be used also as a starting point to create your own custom audiospace companion file for a media file.

Demo Samples

You can download the following demo samples from the VHT website and play the audio samples with VHT app then experimenting by editing the audiospace files to create your own virtual speakers layouts.

16ch_9.1.6 v1 cart.audiospace

This will play a 16 channels audio file using a 9.1.6 speakers layout.

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=9.1.6 v1 cart
mode=CUSTOM_VIRTUAL_SPEAKERS
layout=VECTOR
RIGHT_HANDED
offset=xyz(-100,-100,0)
scale=xyz(0.04,0.04,0.01)

channel[0]*dB(0)->xyz(48,200,0)=FL
channel[1]*dB(0)->xyz(152,200,0)=FR
channel[2]*dB(0)->xyz(100,200,0)=FC
channel[3]*dB(0)->xyz(100,200,-100)=LFE
//channel[3]*dB(0)->HEADPHONES=LFE
channel[4]*dB(0)->xyz(0,100,0)=SL
channel[5]*dB(0)->xyz(200,100,0)=SR
channel[6]*dB(0)->xyz(0,0,0)=BL
channel[7]*dB(0)->xyz(200,0,0)=BR
channel[8]*dB(0)->xyz(0,168,0)=WL
channel[9]*dB(0)->xyz(200,168,0)=WR
channel[10]*dB(0)->xyz(48,152,200)=TFL
channel[11]*dB(0)->xyz(152,152,200)=TFR
channel[12]*dB(0)->xyz(48,100,200)=TSL
channel[13]*dB(0)->xyz(152,100,200)=TSR
channel[14]*dB(0)->xyz(48,48,200)=TBL
channel[15]*dB(0)->xyz(152,48,200)=TBR
```

16ch_9.1.6 v1 cart H.audiospace

This is the same like the previous “9.1.6 v1 cart” layout but notice that the LFE audio channel number three is routed directly to the headphones without any spatial audio rendering.

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=9.1.6 v1 cart_H
mode=CUSTOM_VIRTUAL_SPEAKERS
layout=VECTOR
RIGHT_HANDED
offset=xyz(-100,-100,0)
scale=xyz(0.04,0.04,0.01)
channel[0]*dB(0)->xyz(48,200,0)=FL
channel[1]*dB(0)->xyz(152,200,0)=FR
channel[2]*dB(0)->xyz(100,200,0)=FC
//channel[3]*dB(0)->xyz(100,200,-100)=LFE
channel[3]*dB(0)->HEADPHONES=LFE
channel[4]*dB(0)->xyz(0,100,0)=SL
channel[5]*dB(0)->xyz(200,100,0)=SR
channel[6]*dB(0)->xyz(0,0,0)=BL
channel[7]*dB(0)->xyz(200,0,0)=BR
channel[8]*dB(0)->xyz(0,168,0)=WL
channel[9]*dB(0)→xyz(200,168,0)=WR
channel[10]*dB(0)->xyz(48,152,200)=TFL
channel[11]*dB(0)->xyz(152,152,200)=TFR
channel[12]*dB(0)->xyz(48,100,200)=TSL
channel[13]*dB(0)->xyz(152,100,200)=TSR
channel[14]*dB(0)->xyz(48,48,200)=TBL
channel[15]*dB(0)->xyz(152,48,200)=TBR
```

16ch_9.1.6 v1 pol.audiospace

This is the same like the previous “9.1.6 v1 cart” layout but using a mix of polar and cartesian coordinates input method using LEFT_HANDED axis system (default).

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=9.1.6 v1 pol+cart
mode=CUSTOM_VIRTUAL_SPEAKERS
layout=VECTOR
scale=xyz(0.04,0.01,0.04)
channel[0]*dB(0)->pol(-28,113,0)=FL
channel[1]*dB(0)->pol(28,113,0)=FR
channel[2]*dB(0)->pol(0,100,0)=FC
channel[3]*dB(0)->pol(0,100,-50)=LFE
channel[4]*dB(0)->pol(-90,100,0)=SL
channel[5]*dB(0)->pol(90,100,0)=SR
channel[6]*dB(0)->pol(-136,141.4,0)=BL
channel[7]*dB(0)->pol(136,141.4,0)=BR
channel[8]*dB(0)->pol(-56,121,0)=WL
channel[9]*dB(0)->pol(56,121,0)=WR
channel[10]*dB(0)->xyz(-52,200,52)=TFL
channel[11]*dB(0)->xyz(52,200,52)=TFR
channel[12]*dB(0)->xyz(-52,200,0)=TSL
channel[13]*dB(0)->xyz(52,200,0)=TSR
channel[14]*dB(0)->xyz(-52,200,-52)=TBL
channel[15]*dB(0)->xyz(52,200,-52)=TBR
```

16ch_9.1.6 v2 cart.audiospace

This is another version of 9.1.6 speakers layout.

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=9.1.6 v2 cart
mode=CUSTOM_VIRTUAL_SPEAKERS
layout=VECTOR
LEFT_HANDED
offset=xyz(-100,0,-100)
scale=xyz(0.04,0.01,0.04)
channel[0]*dB(0)->xyz(0,0,200)=FL
channel[1]*dB(0)->xyz(200,0,200)=FR
channel[2]*dB(0)->xyz(100,0,200)=FC
channel[3]*dB(0)->xyz(100,-100,200)=LFE
channel[4]*dB(0)->xyz(0,0,100)=SL
channel[5]*dB(0)->xyz(200,0,100)=SR
channel[6]*dB(0)->xyz(0,0,0)=BL
channel[7]*dB(0)->xyz(200,0,0)=BR
channel[8]*dB(0)->xyz(0,0,168)=WL
channel[9]*dB(0)→xyz(200,0,168)=WR
channel[10]*dB(0)->xyz(0,200,200)=TFL
channel[11]*dB(0)->xyz(200,200,200)=TFR
channel[12]*dB(0)->xyz(0,200,100)=TSL
channel[13]*dB(0)->xyz(200,200,100)=TSR
channel[14]*dB(0)->xyz(0,200,0)=TBL
channel[15]*dB(0)->xyz(200,200,0)=TBR
```

16ch_9.1.6 v2 cart2.audiospace

This is the same like the previous “9.1.6 v2 cart” layout but without offset and scale.

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=9.1.6 v2 cart2
mode=CUSTOM_VIRTUAL_SPEAKERS
layout=VECTOR
channel[0]*dB(0)->xyz(-4,0,4)=FL
channel[1]*dB(0)->xyz(4,0,4)=FR
channel[2]*dB(0)->xyz(0,0,4)=FC
channel[3]*dB(0)->xyz(0,-1,4)=LFE
channel[4]*dB(0)->xyz(-4,0,0)=SL
channel[5]*dB(0)->xyz(4,0,0)=SR
channel[6]*dB(0)->xyz(-4,0,-4)=BL
channel[7]*dB(0)->xyz(4,0,-4)=BR
channel[8]*dB(0)->xyz(-4,0,2.72)=WL
channel[9]*dB(0)->xyz(4,0,2.72)=WR
channel[10]*dB(0)->xyz(-4,2,4)=TFL
channel[11]*dB(0)->xyz(4,2,4)=TFR
channel[12]*dB(0)->xyz(-4,2,0)=TSL
channel[13]*dB(0)->xyz(4,2,0)=TSR
channel[14]*dB(0)->xyz(-4,2,-4)=TBL
channel[15]*dB(0)->xyz(4,2,-4)=TBR
```

16ch_9.1.6 v3.audiospace

This is the same like the previous “9.1.6 v2 cart” layout but without offset and scale.

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=9.1.6 v3
mode=CUSTOM_VIRTUAL_SPEAKERS
layout=SPHERE
sphere_radius=4
front_angle=30
side_angle=90
back_angle=150
top_angle=45
channel[0]*dB(0)=FL
channel[1]*dB(0)=FR
channel[2]*dB(0)=FC
channel[3]*dB(0)=LFE
channel[4]*dB(0)=SL
channel[5]*dB(0)=SR
channel[6]*dB(0)=BL
channel[7]*dB(0)=BR
channel[8]*dB(0)=WL
channel[9]*dB(0)=WR
channel[10]*dB(0)=TFL
channel[11]*dB(0)=TFR
channel[12]*dB(0)=TSL
channel[13]*dB(0)=TSR
channel[14]*dB(0)=TBL
channel[15]*dB(0)=TBR
```

10ch_5.1+4.audiospace

This will play a 10 channels audio file using a 5.1.4 speakers layout.

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=5.1+4
mode=CUSTOM_VIRTUAL_SPEAKERS
layout=SPHERE
sphere_radius=4
front_angle=30
side_angle=110
back_angle=135
top_angle=30
channel[0]*dB(0)=FL
channel[1]*dB(0)=FR
channel[2]*dB(0)=FC
channel[3]*dB(0)=LFE
channel[4]*dB(0)=SL
channel[5]*dB(0)=SR
channel[6]*dB(0)=TFL
channel[7]*dB(0)=TFR
channel[8]*dB(0)=TSL
channel[9]*dB(0)=TSR
```

8ch_7.1 H.audiospace

This is the standard 7.1 layout but LFE audio channel number three is routed directly to the headphones without any spatial audio rendering.

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=7.1 H
mode=CUSTOM_VIRTUAL_SPEAKERS
layout=SPHERE
channel[0]*dB(0)=FL
channel[1]*dB(0)=FR
channel[2]*dB(0)=FC
channel[3]*dB(0)->HEADPHONES=LFE
//channel[3]*dB(0)=LFE
channel[4]*dB(0)=SL
channel[5]*dB(0)=SR
channel[6]*dB(0)=BL
channel[7]*dB(0)=BR
```

1ch_moving ao square.audiospace

In this example one audio object is moving on a square path in 30s.

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=moving ao square
mode=DYNAMIC_AUDIO_OBJECTS
channel[0]@T+(0)*dB(0)->xyz(-4,0,4):m1
channel[0]@T+(7.5)*dB(0)->xyz(4,0,4):m2
channel[0]@T+(15)*dB(0)->xyz(4,0,-4):m3
channel[0]@T+(22.5)*dB(0)->xyz(-4,0,-4):m4
channel[0]@T+(30)*dB(0)->xyz(-4,0,4):m5
```

1ch_moving ao hexa.audiospace

In this example one audio object is moving on a hexagonal path in 30s.

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=moving ao hexa
mode=DYNAMIC_AUDIO_OBJECTS
channel[0]@T+(0)*dB(0)->pol(0,5,0):q0
channel[0]@T+(5)*dB(0)->pol(60,5,0):q60
channel[0]@T+(10)*dB(0)->pol(120,5,0):q120
channel[0]@T+(15)*dB(0)->pol(180,5,0):q180
channel[0]@T+(20)*dB(0)->pol(240,5,0):q240
channel[0]@T+(25)*dB(0)->pol(300,5,0):q300
channel[0]@T+(30)*dB(0)->pol(360,5,0):q360
```

1ch_changing ao gain.audiospace

In this example one static audio object is changing the gain overt time.

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=changing gain
mode=DYNAMIC_AUDIO_OBJECTS
channel[0]@T+(0)*dB(0)->pol(0,5,0):g0
channel[0]@T+(4.9)*dB(0)->pol(0,5,0):g0
channel[0]@T+(5)*dB(-30)->pol(0,5,0):g-30
channel[0]@T+(10)*dB(-30)->pol(0,5,0):g-30
channel[0]@T+(10.1)*dB(0)->pol(0,5,0):g0
channel[0]@T+(15)*dB(0)->pol(0,5,0):g0
channel[0]@T+(20)*dB(-20)->pol(0,5,0):g-20
channel[0]@T+(25)*dB(0)->pol(0,5,0):g0
channel[0]@T+(30)*dB(10)->pol(0,5,0):g10
```

8ch_audio bullets.audiospace

In this example eight dynamic audio bullets objects will be fired toward you.

8ch_audio bullets inverted.audiospace is the inverted version.

Be careful not to be hit!

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=audio bullets
mode=DYNAMIC_AUDIO_OBJECTS
channel[0]@T+(0)*dB(-60)->xyz(0,0,5):b1
channel[0]@T+(2.9)*dB(-60)->xyz(0,0,5):b1
channel[0]@T+(3)*dB(0)->xyz(0,0,5):b1
channel[0]@T+(29.9)*dB(0)->xyz(40,-30,-100):b1
channel[0]@T+(30)*dB(-60)->xyz(40,-30,-100):b1
//
channel[1]@T+(0)*dB(-60)->xyz(0,0,5):b2
channel[1]@T+(6.9)*dB(-60)->xyz(0,0,5):b2
channel[1]@T+(7)*dB(0)->xyz(0,0,5):b2
channel[1]@T+(29.9)*dB(0)->xyz(-40,-10,-200):b2
channel[1]@T+(30)*dB(-60)->xyz(-40,-10,-200):b2
//
channel[2]@T+(0)*dB(-60)->xyz(0,0,5):b3
channel[2]@T+(8.9)*dB(-60)->xyz(0,0,5):b3
channel[2]@T+(9)*dB(0)->xyz(0,0,5):b3
channel[2]@T+(29.9)*dB(0)->xyz(-11,21,-100):b3
channel[2]@T+(30)*dB(-60)->xyz(-11,21,-100):b3
//
channel[3]@T+(0)*dB(-60)->xyz(0,0,5):b4
channel[3]@T+(10.9)*dB(-60)->xyz(0,0,5):b4
channel[3]@T+(11)*dB(0)->xyz(0,0,5):b4
channel[3]@T+(29.9)*dB(0)->xyz(4,-30,-100):b4
channel[3]@T+(30)*dB(-60)->xyz(4,-30,-100):b4
//
channel[4]@T+(0)*dB(-60)->xyz(0,0,5):b5
channel[4]@T+(15.9)*dB(-60)->xyz(0,0,5):b5
channel[4]@T+(16)*dB(0)->xyz(0,0,5):b5
channel[4]@T+(29.9)*dB(0)->xyz(11,-20,-500):b5
channel[4]@T+(30)*dB(-60)->xyz(11,-20,-500):b5
//
channel[5]@T+(0)*dB(-60)->xyz(0,0,5):b6
channel[5]@T+(21.9)*dB(-60)->xyz(0,0,5):b6
channel[5]@T+(22)*dB(0)->xyz(0,0,5):b6
channel[5]@T+(29.9)*dB(0)->xyz(18,20,-100):b6
channel[5]@T+(30)*dB(-60)->xyz(18,20,-100):b6
//
channel[6]@T+(0)*dB(-60)->xyz(0,0,5):b7
channel[6]@T+(24.9)*dB(-60)->xyz(0,0,5):b7
channel[6]@T+(25)*dB(0)->xyz(0,0,5):b7
channel[6]@T+(29.9)*dB(0)->xyz(40,-80,-100):b7
channel[6]@T+(30)*dB(-60)->xyz(40,-80,-100):b7
//
channel[7]@T+(0)*dB(-60)->xyz(0,0,5):b8
channel[7]@T+(25.9)*dB(-60)->xyz(0,0,5):b8
channel[7]@T+(26)*dB(0)->xyz(0,0,5):b8
channel[7]@T+(29.9)*dB(0)->xyz(-20,15,-100):b8
channel[7]@T+(30)*dB(-60)->xyz(-20,15,-100):b8
```

1ch_moving ao from left to right.audiospace

In this example one audio object moving from left to right in 10 seconds.

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=left to right
mode=DYNAMIC_AUDIO_OBJECTS
channel[0]@T+(0)*dB(0)->xyz(-20,0,3):bird
channel[0]@T+(10)*dB(0)->xyz(20,0,3):bird
```

Two moving AOs.audiospace

In this example are two moving audio objects, one moving from left to right and another from top to floor level in 10 seconds.

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=2 mov AOs
mode=DYNAMIC_AUDIO_OBJECTS
channel[0]@T+(0)*dB(0)->xyz(-20,0,3):bird
channel[0]@T+(10)*dB(0)->xyz(20,0,3):bird
channel[1]@T+(0)*dB(0)->xyz(0,10,3):ball
channel[1]@T+(10)*dB(0)->xyz(0,-1.2,3):ball
```

Four static AOs.audiospace

Four static AOs representing a music formation on a stage.

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=Four static AOs
mode=STATIC_AUDIO_OBJECTS
channel[0]*dB(0)->xyz(2,0.5,3.5):Henry voice
channel[1]*dB(0)->xyz(-2,0.4,4):Emma voice
channel[2]*dB(0)->xyz(0,-0.3,4):Guitar
channel[3]*dB(0)->xyz(1,-0.1,5):Drums
```

VHT demo dynamic audio object

This is a step-by-step demonstration work flow on how to create a playable stereoscopic VR image and video with static or dynamic audio objects.

For this you will need Daz Studio 3D modeling software, VHT tools for DAZ and FFmpeg.

A "VHT demo dynamic audio object.duf" sample scene is provided.

This scene has one audio object attached to a cube and one EQD180 camera.

The camera and cube are animated.

Read the README.txt for more info.

Play the “VHT demo dynamic audio object R_2023-11-15-18-53-52_EQD180_FF3D_SbS_crosseye_tilt-0_.mkv” for static audio object demonstration.

Play the “VHT demo dynamic audio object R_2023-11-15-18-48-34_EQD180_FF3D_SbS_crosseye_tilt-0_.mkv” for dynamic audio object demonstration.

```
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

track=0
name=n/a
mode=DYNAMIC_AUDIO_OBJECTS
channel[0]@T+(0.000)*dB(0)->xyz(1.286,0.706,4.526):AO cube
channel[0]@T+(0.667)*dB(0)->xyz(1.266,0.706,4.475):AO cube
channel[0]@T+(0.933)*dB(0)->xyz(1.246,0.706,4.428):AO cube
channel[0]@T+(1.167)*dB(0)->xyz(1.222,0.706,4.375):AO cube
channel[0]@T+(1.267)*dB(0)->xyz(1.227,0.706,4.317):AO cube
channel[0]@T+(1.333)*dB(0)->xyz(1.239,0.706,4.258):AO cube
channel[0]@T+(1.400)*dB(0)->xyz(1.258,0.706,4.185):AO cube
channel[0]@T+(1.467)*dB(0)->xyz(1.280,0.706,4.098):AO cube
channel[0]@T+(1.533)*dB(0)->xyz(1.306,0.706,4.000):AO cube
channel[0]@T+(1.567)*dB(0)->xyz(1.320,0.706,3.947):AO cube
channel[0]@T+(1.600)*dB(0)->xyz(1.334,0.706,3.893):AO cube
channel[0]@T+(1.633)*dB(0)->xyz(1.349,0.706,3.836):AO cube
channel[0]@T+(1.667)*dB(0)->xyz(1.363,0.706,3.777):AO cube
channel[0]@T+(1.700)*dB(0)->xyz(1.378,0.706,3.717):AO cube
channel[0]@T+(1.733)*dB(0)->xyz(1.393,0.706,3.656):AO cube
channel[0]@T+(1.767)*dB(0)->xyz(1.407,0.706,3.593):AO cube
channel[0]@T+(1.800)*dB(0)->xyz(1.421,0.706,3.530):AO cube
channel[0]@T+(1.833)*dB(0)->xyz(1.435,0.706,3.466):AO cube
channel[0]@T+(1.867)*dB(0)->xyz(1.449,0.706,3.402):AO cube
channel[0]@T+(1.900)*dB(0)->xyz(1.462,0.706,3.338):AO cube
channel[0]@T+(1.933)*dB(0)->xyz(1.474,0.704,3.274):AO cube
...
```

More demo samples

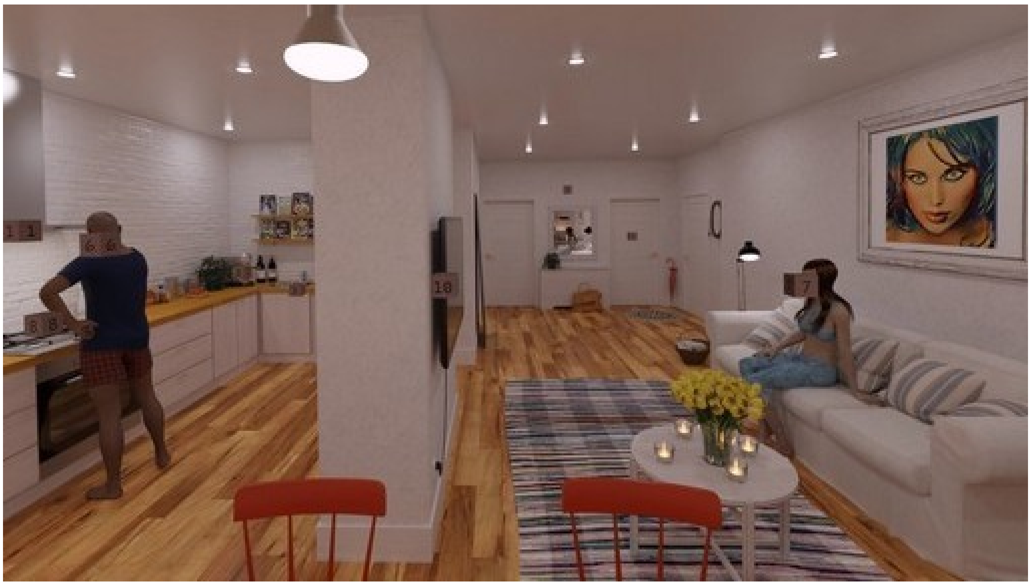
Go to https://virtualhometheater.com/

to download a 3D stereoscopic 360 panoramic demo sample with 11 static audio objects:

```
channel[0]*dB(-13)->xyz(-3.220,-0.661,-0.148):AO radio
channel[1]*dB(3)->xyz(-3.553,-0.059,2.838):AO hood
channel[2]*dB(9)->xyz(-3.629,-0.651,5.386):AO espresso
channel[3]*dB(11)->xyz(-3.062,-0.907,5.665):AO sink
channel[4]*dB(24)->xyz(1.716,-0.425,9.317):AO knock knock
channel[5]*dB(28)->xyz(0.504,0.466,8.992):AO bell
channel[6]*dB(-8)->xyz(-2.814,-0.152,2.728):AO James voice
channel[7]*dB(-5)->xyz(1.827,-0.518,3.620):AO Olivia voice
channel[8]*dB(8)->xyz(-3.603,-0.750,3.006):AO kooking
channel[9]*dB(-12)->xyz(-2.440,0.347,-0.723):AO clock
channel[10]*dB(0)->xyz(-0.614,-0.463,3.140):AO TV
```

made by using the following software applications:

- DAZ Studio 4.22
- VHT tools for DAZ 1.0.0
- Virtual Home Theater 1.18.x
- Audacity
- FFmpeg



Acronyms

- VHT – Virtual Home Theater
- dB – Decibel (audio signal gain amplification)

Head level virtual speakers VHT names:

- **FL** – Front Left
- **FR** – Front Right
- **FC** – Front Center
- **FLC** – Front Left Center
- **FRC** – Front Right Center
- **WL** – Wide Left
- **WR** – Wide Right
- **LFE** – Main Low Frequency Effect (LF)
- **LFE2** – Second Low Frequency Effect (LF2)
- **SL** – Side Left
- **SR** – Side Right
- **BL** – Back Left
- **BC** – Back Center
- **BR** – Back Right

Above head level virtual speakers VHT names:

- **TFL** – Top Front Left
- **TFC** – Top Front Center
- **TFR** – Top Front Right
- **TSL** – Top Side Left
- **TSR** – Top Side Right
- **TBL** – Top Back Left
- **TBC** – Top Back Center
- **TBR** – Top Back Right
- **TC** – Top Center (Above Head)

Bellow head level virtual speakers VHT names:

- **DFL** – Down Front Left
- **DFC** – Down Front Center
- **DFR** – Down Front Right
- **DSL** – Down Side Left
- **DSR** – Down Side Right
- **DBL** – Down Back Left
- **DBC** – Down Back Center
- **DBR** – Down Back Right
- **DC** – Down Center (Bellow Foot)

Links

Virtual Home Theater:

- https://virtualhometheater.com/
- https://store.steampowered.com/app/989060/
- https://youtu.be/uuFLk3BQLus

Sound system speaker layout recommendations:

- https://www.itu.int/rec/R-REC-BS.2051/en
- https://www.dolby.com/siteassets/technologies/dolby-atmos/atmos-installation-guidelines-121318_r3.1.pdf

Revision History

- v1.0 – Feb 17, 2024: First release