

# SPAT

REVOLUTION



# Flux Spat Revolution

**The Immersive Audio Revolution**

## **A User Guide**

**Written by Cristian Vogel**

with contributions from

G ael Martinet, Hugo Larin, Vincent Carlier, Thibaut Carpentier, Anders Tveit.



## Legal Information

First edition of Flux:: Spat Revolution User Guide published November 2018

All writing is protected under copyright ©2018 Cristian Vogel / Flux:: Sound and Picture Development

Flux Sound and Picture Development is a trademark of Gaël Yvan Son et Développement.

All trademarks and the logos are copyright protected.

Flux:: Spat Revolution is published by

Gaël Yvan & Son et Développement

2, Rue de la Chasse

45000 ORLEANS

R.C.S 431 616 770 ORLEANS

IRCAMTOOLS SPAT Copyright 2018 Flux:: sound and picture development and IRCAM. All rights reserved.

SPAT Revolution Copyright 2018 Flux:: sound and picture development and IRCAM. All rights reserved.

<b>1. Welcome to Spat Revolution</b>	Environment Setup Editor 68
Spat Revolution's Heritage 7	VU Meters 70
About this Guide 8	Inputs 73
<b>2. Installation and Activation 9</b>	Mono Input 74
iLok User Account 10	Two Channel 75
iLok License Manager 11	Four Channel Input 76
Flux Center 13	Multi-Channel Based Input 77
Installation Options 15	Input Transcoder 80
Beta Versions 17	Transcoding Matrix 81
<b>3. Users and Applications 18</b>	When to Transcode Inputs? 82
<b>4. Spat Revolution Project Environment 20</b>	Source 83
Rendering To Speakers 23	Tracking 85
Diffusion 24	Room 86
Routing Matrix 25	Room Graphic Engine 88
Custom Speaker Configuration 27	Room Camera View 89
The Ambisonic Method 30	Room Output Configuration 90
Introduction to Ambisonic 31	Channel Based Room 91
Rendering to Ambisonic Format 34	Nebula Spatial Spectrogram 92
Multichannel Multiformat Audio Data 35	Virtual Room Concepts 95
<b>5. Spatialisation Technology 36</b>	Speakerless Virtual Rooms - Binaural and Ambisonic 100
Binaural 37	Binaural Room 104
HRTF 38	High Order Ambisonic Room 106
HRTF Profiles 39	B-Format Room 110
Binaural Monitoring Module 40	Mid-Side Room 111
Binaural Rooms 42	Barycentric Groups in Rooms 112
Transaural 44	Master Section 115
Channel Based Streams 45	Output 118
Listener Position 48	<b>7. SPAT Plugins 119</b>
Panning Algorithms 51	Single Hardware Workflow 120
Room Panning Types 54	Distributed Hardware Workflow 122
Vector Base Amplitude (VBAP) 55	Local Audio Path Workflow 123
Vector Base Intensity (VBIP) 56	Troubleshooting Local Audio 126
Distance Base Angular (DBAP) 57	<b>8. Artificial Reverberation 128</b>
K Nearest Neighbour (KNN) 58	Reverb Parameters 130
Speaker-Placement Correction Amplitude (SPCAP) 60	Defaults 132
Ambisonic Equivalent (AEP) 61	Preset Memories 133
Angular and PanR 63	Reverb General 134
XYAB 64	Reverb Perceptual Factors 136
<b>6. Spat Environment 66</b>	Room Response Parameters 137
Global Bars 67	Reverb Options 139
	Crossover 141

Reverb Design Presets	142
<b>9. Source Parameters</b>	<b>143</b>
Defaults	144
Preset Memories	145
Multiple Source Selection	146
Smart Property Filter	148
Perceptual Factors	149
Reverb Options	151
Axis Omni Filters	152
Spreading	155
Send LFE	156
Barycentric	157
Options	158
<b>10. Automation</b>	<b>160</b>
DAW Automation - Local Audio Path	162
DAW Automation - Manual setup	163
OSC Connections Matrix	165
Introduction to Generic OSC	167
<b>11. Set Up Examples</b>	<b>170</b>
Parallel Room Mixes	171
Disconnecting and Connecting	173
Quick Start Graph	174
Multi Matrix Routing	175
Channel Based Setup Examples	175
Dolby Atmos 7.1.2 Room 16 stereo sources	176
Mixing Surround Sources	178
Mixing Acoustic Music in Stereo	180
Upmixing Surround Formats	182
Explore Ambisonic Decoding Methods	186
<b>12. Third Party Integration</b>	<b>188</b>
Cockos Reaper DAW	189
Bitwig Studio	204
ProTools	211
Figure53 QLab	219
Avid VENUE S6L	228

<b>Appendix A</b>	
BlackTrax tracking system	236
<b>Appendix B</b>	
Spat Plugins Troubleshooting	243
<b>Appendix C</b>	
Sub and LFE Channels	248
<b>Appendix D Credits</b>	

# 1. Welcome to Spat Revolution

First of all, thank you for purchasing *Spat Revolution*. We hope it will provide you with new levels of productivity, creativity and experience in sound design. Our goal was to deliver the most comprehensive real-time spatial audio renderer ever designed, and to make the whole process of spatialised audio production powerful and intuitive for beginners and professionals alike. Our real-time audio environment provides easy access to some of the most advanced spatialisation algorithms currently available, at the very best sound quality currently possible.

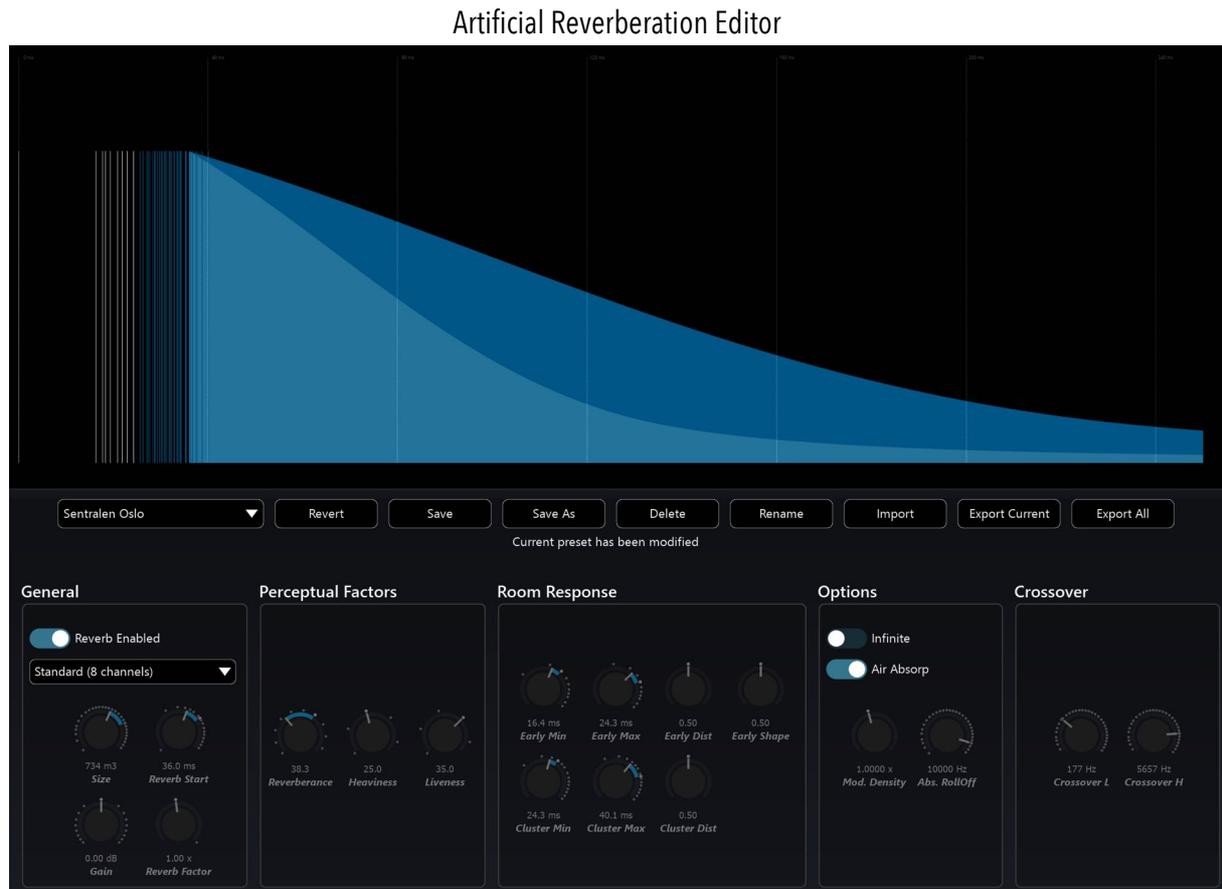
'Spat' is short for *Spatialisateur* in French. It is a real-time audio library that allows composers, sound artists, performers and sound engineers to control the localisation of sound sources in virtual and real 3D auditory spaces.



*Spat Revolution* wraps the 'Spat' processing library in a luxurious and characteristic graphic environment to help visualise many aspects of a spatial audio composition in realtime. This graphic interface allows sound mixes to be composed as interac-

tive spatial models existing in *Virtual Room*. High definition graphic displays like the unique *Nebula* analyser for example, can simulate how sound sources localise their direct sound over different speaker setups.

In addition, Spat contains a powerful multichannel reverberation engine which can be applied to design and add a sense of auditory space in studio mixes and real-time on location.



**Spat Revolution maintains the highest audio quality throughout the entire signal flow**

## 1.1

# Spat Revolution's Heritage

## **Spat Revolution is underpinned by 30 years of technical research and development in acoustics and sound.**

Behind *Spat Revolution* is the partnership of Flux and IRCAM in Paris, France. Founded in 1977, IRCAM is one of the world's leading public research institutes in the fields of musical expression, science of music, sound and acoustics. The first result of this partnership was the plugin suite IRCAM Tools. In that release was the first incarnation of SPAT as a DAW plugin based on over 30 years of research with IRCAM's Acoustic and Cognitive Spaces Team. After decades of development the next step was the full production environment for spatial audio - *Spat Revolution* - through the elegant simplicity of the graphic interface, *Spat Revolution* represents a formidable achievement. It brings together the technical expertise from decades of academic research and development at IRCAM into an easy to use package that is flexible and powerful enough to meet all the demands of spatial audio production, from day to day surround sound post-production to the most challenging real-time installations. As you will discover *Spat Revolution* can handle a virtually unlimited number of input and output audio streams and is prepared for all formats and 3D-audio workflows currently available or imaginable in the audio industry.

Although this user guide cannot cover every technical aspect of the algorithms and expert knowledge contained within *Spat Revolution* we hope you will feel assured that the science behind the sound is absolutely correct.

## 1.2

# About this Guide

This guide has been written for practitioners already working in immersive sound production yet new to the Spat Revolution software environment. It is also intended to be read as a practical introduction to surround and immersive audio production for those who are new to the medium and coming to it through Spat Revolution. Of course, there is plenty more knowledge to be had in the field of immersive audio and the technology behind it, which will strengthen your understanding and decision making. We have included an appendix with some references for further reading about Ambisonics, Channel Based audio systems and acoustics for those who are curious to find out more.

We strongly suggest spending an hour or two to read through this guide before starting on your first major production and keep it on hand during the process. Expert support and advice can also be sought from the Flux Spat Revolution support team and web-based knowledge base.

If you prefer a fast start introduction to the characteristics of the Spat Revolution environment, certainly jump ahead to Section 6, 10 or 11.

But let us start now with a practical guide on getting the software installed and running.

# 2 Installation and Activation

The following section will guide you through the procedure for software license activation. It involves two stages. Firstly, a license activation <sup>1</sup> through the iLok license management system followed by the main download and installation through the Flux:: Center installation software.



---

<sup>1</sup> If you have an activation code from a dealer purchase or from a trial request then please visit our activation page <https://fluxhome.com/license-activation/>

## 2.1

# iLok User Account

*Flux* uses the iLok license management system to deliver software licenses to users. If you don't have an iLok account yet, then please create a free iLok account at [www.ilok.com](http://www.ilok.com) and download the iLok license manager.

A *Flux* Spat Revolution license come with two activations that need to be linked to your user account. Having two activations gives you the possibility of having a fixed license on one particular machine and a portable license on an iLok USB key if you own one, or a Main and Back up Spat Engine running on a live show.

### **To Activate Licenses:**

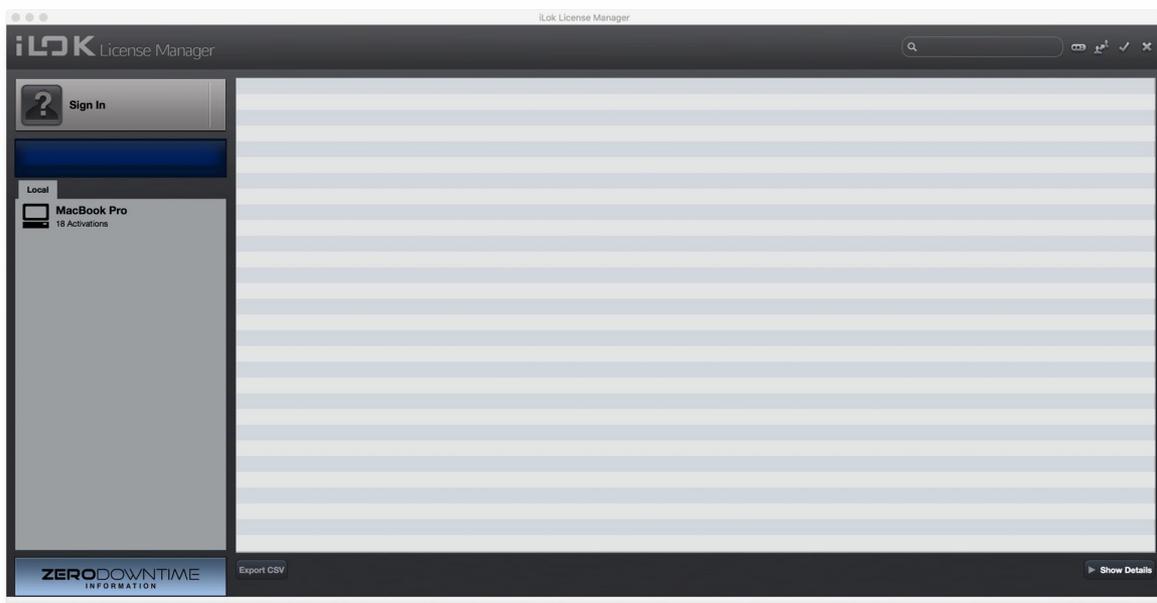
- An iLok user account is required
- An iLok USB key is optional

## 2.2

# iLok License Manager

If you already have an iLok user account, licenses will have been connected directly to your account. Skip ahead to the next section guiding you through the *Flux Center*

For new iLok users, the first step is to download and install the iLok license manager available on the home page of the iLok website. When your user account is successfully activated and the iLok license manager correctly installed, you can start the software and log in to your iLok user account.



Pressing on the Sign In button will allow you to connect to your account. After Logging in, you are now ready to transfer any licenses to a computer or to any iLok USB key if you happen to have one. The process of transferring a license is as simple as dragging the license from the *Available* tab to your Local Computer on the left side.

The screenshot shows the iLok License Manager interface. At the top, there are statistics: All Licenses (430), Available (36), All Activations (369), Unavailable (86), and Hidden (1708). A search bar contains the text 'spat'. On the left sidebar, there is a user profile section with '430 Licenses' and a 'Local' section for 'MacBook Pro' with '18 Activations'. A red arrow points from the 'Local' section to the first row of the license table.

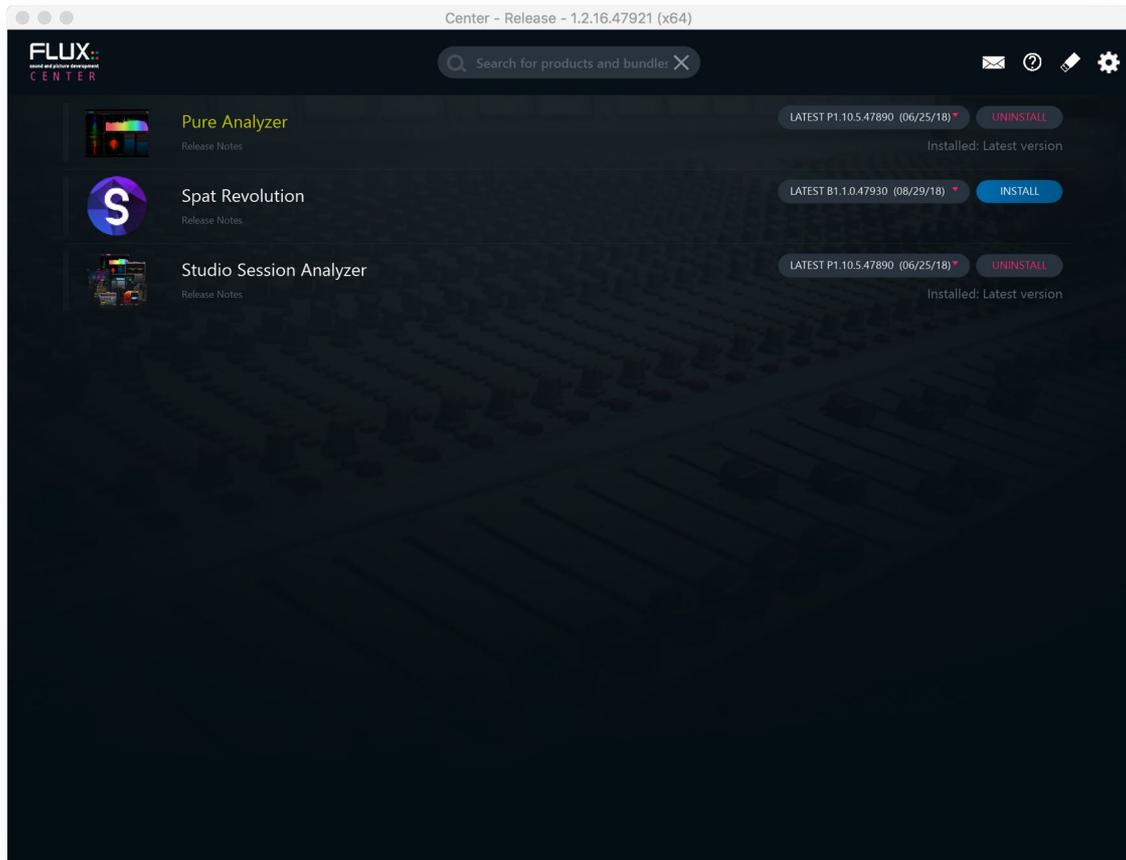
Valid Locations	Product Name	Publisher Name	Subtype	Expiration Date	Deposit Date	Type	License Status	Owner	Activations	Publisher License ID
	Spat Revolution	Flux	Product	Expires 30 Days after activation	2018-05-31 09:35 AM	Trial/Demo	Active	You	0 of 2 activations used	
	Spat Revolution	Flux	Product	2019-05-13 01:20 PM	2018-05-18 01:15 PM	Not for Resale	Active	You	0 of 2 activations used	
	Spat Revolution	Flux	Product	Expires 360 Days after activation	2018-05-09 03:25 PM	Not for Resale	Active	You	0 of 2 activations used	
	Spat Revolution	Flux	Product	2019-01-16 11:33 AM	2017-12-11 11:35 AM	Not for Resale	Active	You	1 of 2 activations used	
	Spat Revolution	Flux	Product		2017-03-08 12:06 PM	Not for Resale	Active	You	1 of 2 activations used	
	Spat Revolution	Flux	Product		2017-03-08 12:05 PM	Not for Resale	Active	You	1 of 2 activations used	
	Spat Revolution	Flux	Product		2017-03-08 12:05 PM	Not for Resale	Active	You	1 of 2 activations used	
	Spat Revolution	Flux	Product		2017-03-08 12:05 PM	Not for Resale	Active	You	1 of 2 activations used	

You are then set with a license on your Local Computer or on an iLok USB key.

If you require further information about iLok and managing licenses please refer to [iLok.com](http://iLok.com) website.

## 2.3

# Flux:: Center

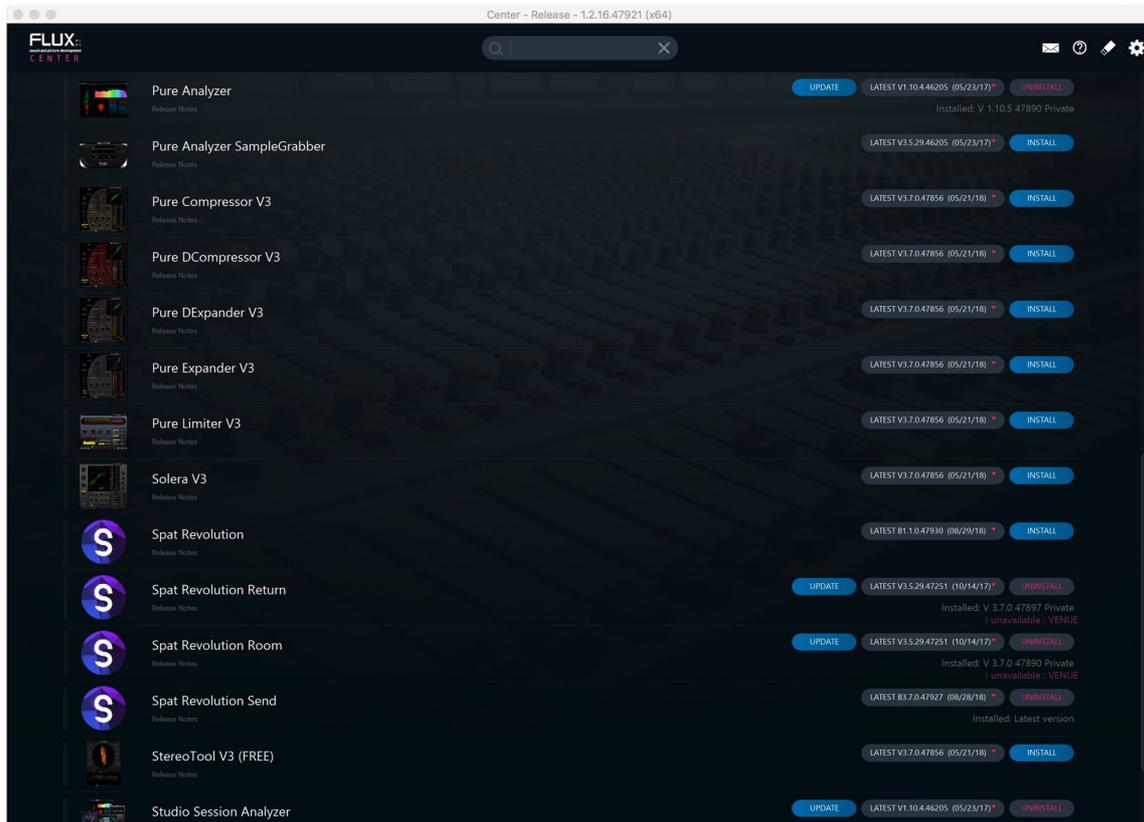


Next step is to get the installers for the *Flux* products you are licensed for. All the software and plugins from *Flux* are available via our custom *Flux:: Center* software. This is a Mac or Windows application we have created to help keep your *Flux* products up to date and to give you a clear overview of what you have installed. Firstly, please visit the download section of the *Flux* Website to get the installer for the *Flux:: Center* application at <https://fluxhome.com/download/>

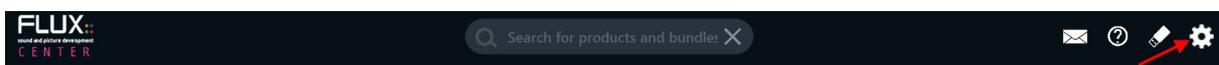
On this page you will find a Mac OS X, Windows 32-bit and Windows 64-bit version of the installer. After downloading and installing, you can open the *Flux:: Center* application to begin the process of installing the *Spat Revolution* software

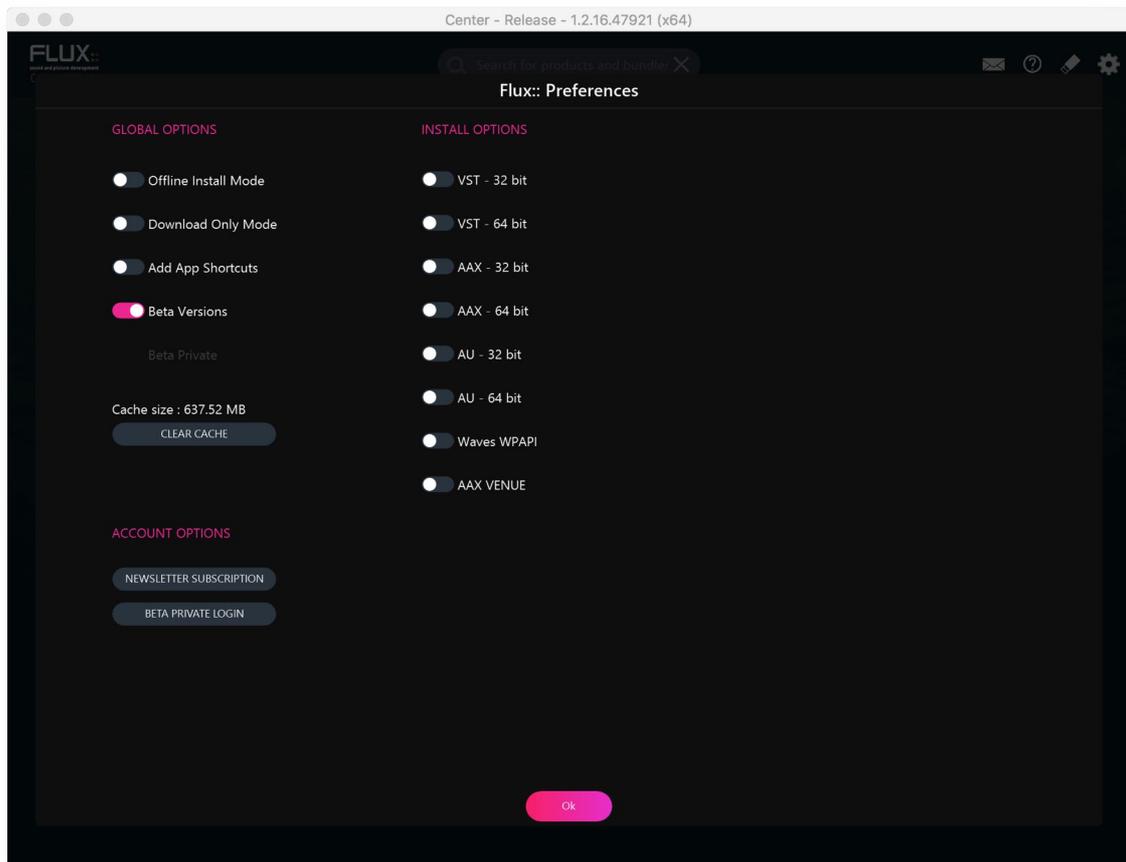
## 2.4

# Installation Options



When you open *Flux:: Center* you will see a page that lists all *Flux* products available for you to install. You will also find information about which version you have currently installed on your system and which new versions might be available for you to update to. You can select versions to install - or uninstall if necessary - using the pull down menus. If you would like to access more installer options such as your preferred plug-in format, please click on the gear icon to the top right of the header area.





This preferences page will allow you to choose various installation options such as preferred plug-in formats for your system. Choosing your format and returning to the main page by pressing the OK button will show all your install options for software and plugins based on the desired formats chosen. When an option is enabled, the corresponding checkbox is pink.

*△ Spat Revolution Send/Room/Return plugins are available in VST-64bit, AU-64bit and AAX-64bit only*

## 2.5

# Beta Versions

If you would like to be closer to the most current development cycles of the software, you can enable the Beta Versions option. This will give you access to a special set of software installers from the pull down menus on the main *Flux Center* page. Beta versions are the new builds that are still under development but may contain useful bug fixes and new features. If you find that a Beta Version is not stable enough for you, then you can always roll back to a stable release version at any time through the *Flux Center* installer.

*⚠ Try a beta on a safety copy of a project - just in case things go wrong.*

# 3. Users and Applications

*Spat Revolution* is aimed at all practitioners working in the medium of spatial audio and 3D sound production - old and new. It is an expert system intended for professional use but its intuitive graphical interface invites a more diverse range of creators to engage in spatial sound production - if it is your first experience working with spatial sound technology, *Spat Revolution* is a great way to start out directly with a professional quality level.

*Spat Revolution* spatialises and renders audio in real-time. It is used in:

- post production for film
- studio composition
- concert diffusion
- virtual reality
- audio for video games
- 360 video soundtracks
- 3D Audio for broadcast
- interactive sound art
- live sound mixing
- scientific research and development
- sound design for film, music and theatre
- environmental sound

In the contexts of live concerts or sound installation, the composer or sound mixer can associate sound events with a room effect or a specific position in space. Virtual objects can be controlled on screen, or by a sequencer, a score-following system, or any other algorithmic approach. *Spat* can easily be linked to any remote control device (tracking systems, tablet, smartphone, joystick, gestural sensors, etc.) through OSC and RTTrPM interfacing protocols.

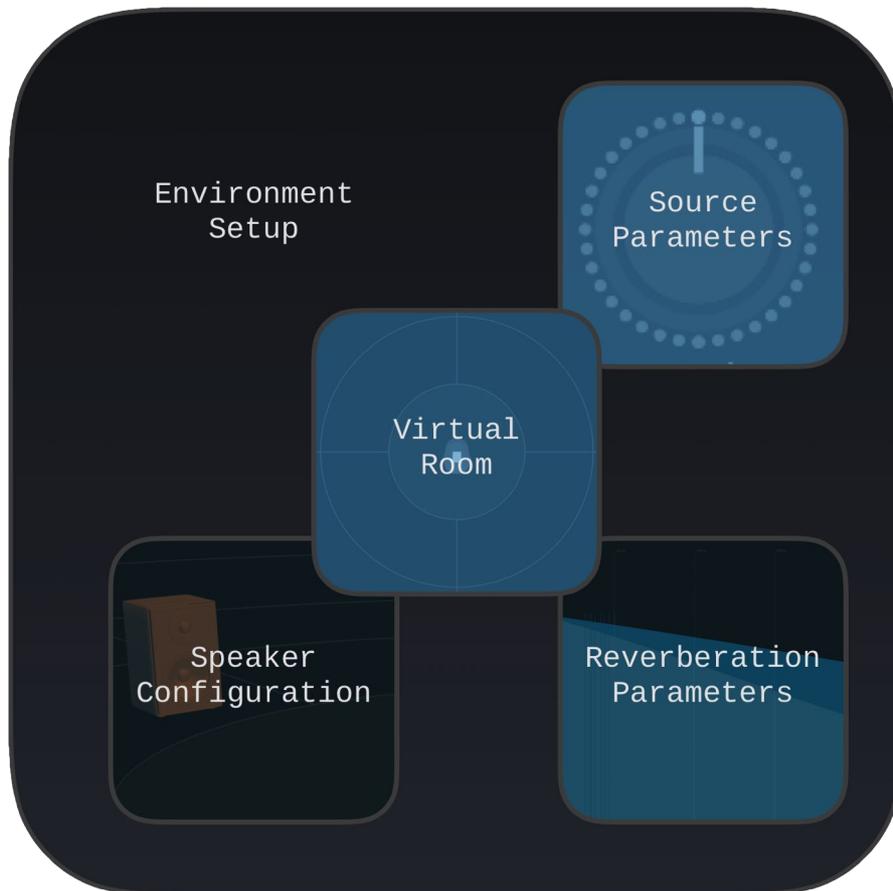
In the contexts of studio mixing and post-production, a virtual source can receive its audio from individual channels on a mixing desk with additional controllers easily set up to allow hands on control of the positions and characteristics of virtual sources and associated room effect. Spat Revolution can re-mix a multichannel mix from one format in a virtual room that could be rendered in a different output format, allowing a novel approach to up- and down-mixing.

In the context of Augmented and Virtual Reality, a spatialised auditory component is essential in creating the sensations of presence and immersion in interactive virtual reality applications. In such scenarios, the B-Format and binaural 3D capabilities of Spat is particularly well suited.

An important thing to keep in mind is the long heritage and technical expertise behind *Spat Revolution*. There are many critical factors to consider when multichannel audio is actually applied in the real world. But rest assured that the algorithms behind your spatial sound project are being implemented correctly at the stage of the authoring and rendering environment. It is good to know that *Spat Revolution* is built with such a high level of technical expertise in such an intuitive package.

**Whatever the scale of your spatialised audio production, *Spat Revolution* makes it simple to craft an impressive and reliable end result.**

# 4. Spat Revolution Project Environment



## 4.1 An Overview

The Spat Revolution production environment is based around five graphical user interfaces that you will focus on at different stages of a spatialisation project.

### Environment Setup

- Signal Routing, Format Transcoding, Panning Algorithms, Mixing

### Speaker Configuration

- Sound system modelling

### Virtual Rooms

- Dynamic control, Spatial positioning, Scene Visualisation

### Source Parameters

- Perceptual sound design

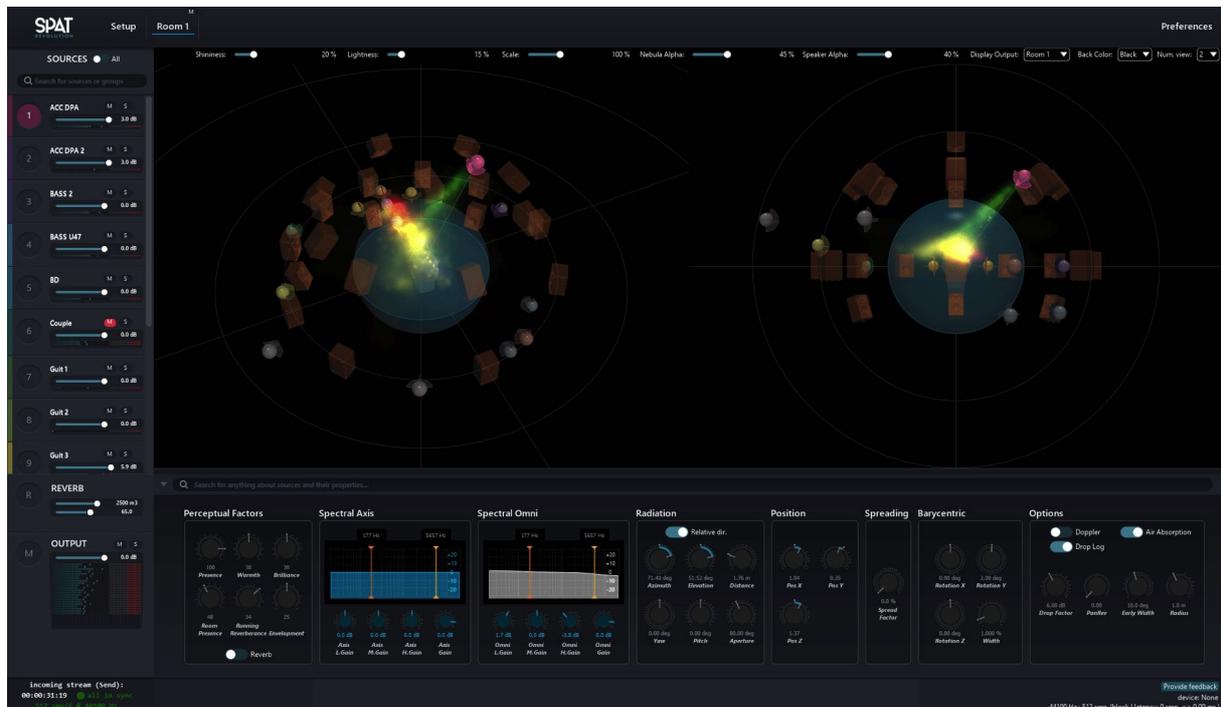
### Reverberation Parameters

- Perceptual sound design for artificial reverberation

We will go through each of these areas in more detail in section 6. But first, let us start with a couple of broad workflow concepts involved in the production of digital audio for immersive sound - **rendering and diffusion**.

# Virtual scenes are rendered as *graphics* on a *monitor screen* as *sounds* in a *physical space* and as *stems* to *disk*.

Spat Revolution gives you an unprecedented level of control over the position and characteristics of virtual sound sources placed in virtual spaces. This is the main function of the **Virtual Rooms** editor where you can adjust and visualise the position and acoustic characteristics of virtual sound sources and compose virtual scenes with them on screen.



## 4.2

# Rendering To Speakers

In order for the virtual scene to translate correctly as an immersive sound experience in a speaker format, SPAT needs to have an accurate model of a *multi-channel speaker arrangement* which will be used to map the multi channel information to the destination speakers and render the sound field correctly in a real space. To this end, you will find a large library of standard and specialised speaker arrangements already built into SPAT which can be used in various places throughout the **Environment Setup** .

Speaker configurations can be used to fit the format of a virtual room to match the actual speaker system being used to diffuse the mix in a real room. Channel based speaker configurations can also be used to transcode the format of a virtual source into a virtual room. More about this later.

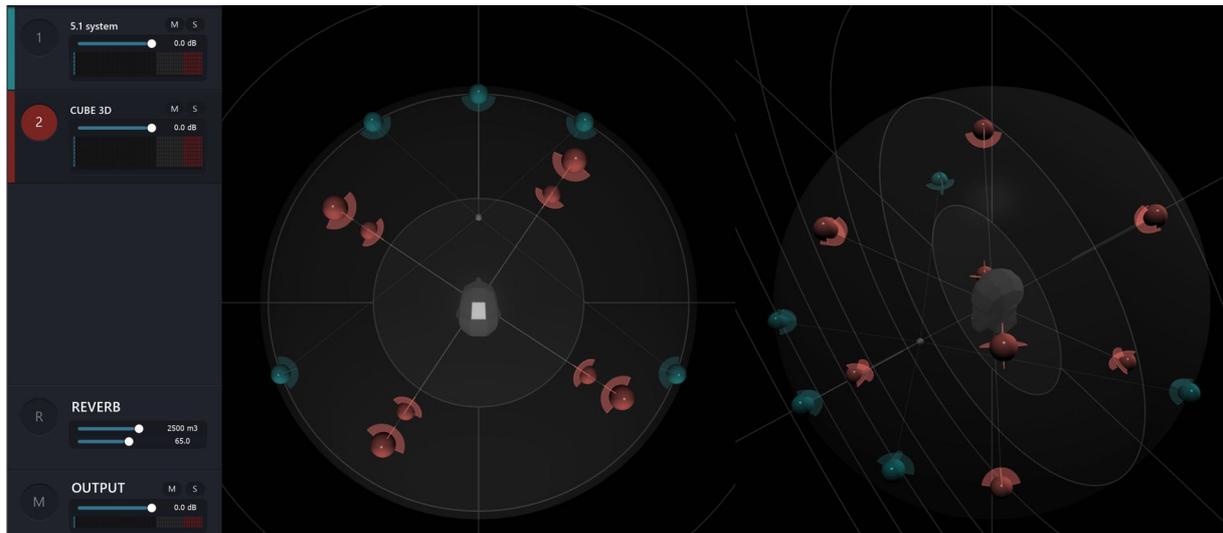
The golden rule when working with multichannel based audio is that you must be sure to choose *exactly* the right formats, speaker arrangements and channel routing throughout, otherwise the virtual space will not map correctly into a physical space.

## 4.3

# Virtual and Real Diffusion

Successful diffusion of a sound field in space relies on every loudspeaker being assigned correctly to each software rendered output channel.

A diffusion system could range from a simple pair of headphones to a 64 speaker array and anything in-between. In some of the more virtualised workflows of Spat Revolution, you may even be thinking about diffusion in a virtual space on configurations of virtual speaker arrangements and channel based formats. The same rule for successful diffusion applies here - the diffusion in the virtual room will be compromised and sound off, if the channel assignments to the virtual systems are incorrect.



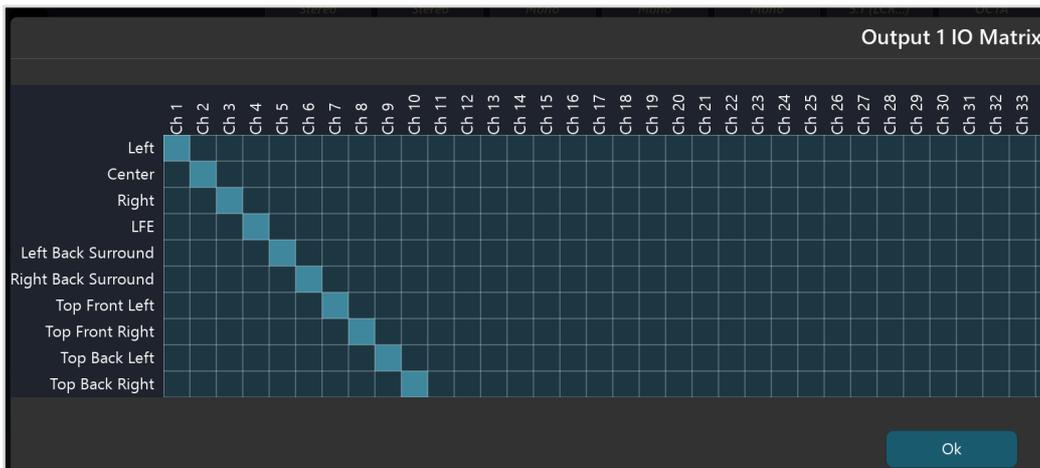
In the above illustration a virtual 5.1 and a virtual Cube arrangement exist together in a High Order Ambisonic Room, which may eventually be rendered to some other channel based end format.

## 4.4

# Routing Matrix

As you can imagine routing and patching high density channel counts can get complicated, especially when setup time is a critical factor. When it comes to that on location, in a virtual space or in the studio, the SPAT routing matrix is there to help. You will find it at many points throughout the **Environment Setup** graph.

In a real setup, the channel based outputs from SPAT can be patched to their corresponding speakers by simply clicking around a Matrix and routing the correct signal to the correct output. In a virtual sound system, or when channels need to be remapped, the Matrix will be an invaluable workflow tool. We have found it to be much less complicated than trying to handle virtual cables on screen.



*⚠ Avoid cable swapping on the loudspeaker setup, use software routing instead*

# Spat Revolution simplifies multichannel configuration in real and virtual spaces.

Speaker configuration: 13.1 Auro 3D

Buttons: Duplicate, New, Delete, Rename, Export, Import, Normalize, Z => 0

Alignment: Reset, Compute

Nur Name	x	y	z	Azimuth	Elevatio	Distance	Delay	Gain
1 Left	-1.00000	1.73205	0.00000	-30.000	0.000	2.000	0.000	0.000
2 Center	0.00000	2.00000	0.00000	0.000	0.000	2.000	0.000	0.000
3 Right	1.00000	1.73205	0.00000	30.000	0.000	2.000	0.000	0.000
4 LFE	0.00000	2.00000	0.00000	0.000	0.000	2.000	0.000	0.000
5 Left Back Surround	-1.87931	-0.68404	0.00000	-110.000	0.000	2.000	0.000	0.000
6 Right Back Surround	1.87939	-0.68404	0.00000	110.000	0.000	2.000	0.000	0.000
7 Top Front Left	-0.86603	1.50000	1.00000	-30.000	30.000	2.000	0.000	0.000
8 Top Front Right	0.86603	1.50000	1.00000	30.000	30.000	2.000	0.000	0.000
9 Top Front Center	0.00000	1.73205	1.00000	0.000	30.000	2.000	0.000	0.000
10 Top Back Left	-1.62760	-0.59241	1.00000	-110.000	30.000	2.000	0.000	0.000
11 Top Back Right	1.62760	-0.59241	1.00000	110.000	30.000	2.000	0.000	0.000
12 Top Back Center	-0.00000	-1.73205	1.00000	180.000	30.000	2.000	0.000	0.000
13 Back Center	-0.00000	-2.00000	0.00000	180.000	0.000	2.000	0.000	0.000
14 VOG	0.00000	0.00000	2.00000	0.000	90.000	2.000	0.000	0.000

Buttons: Close, Apply

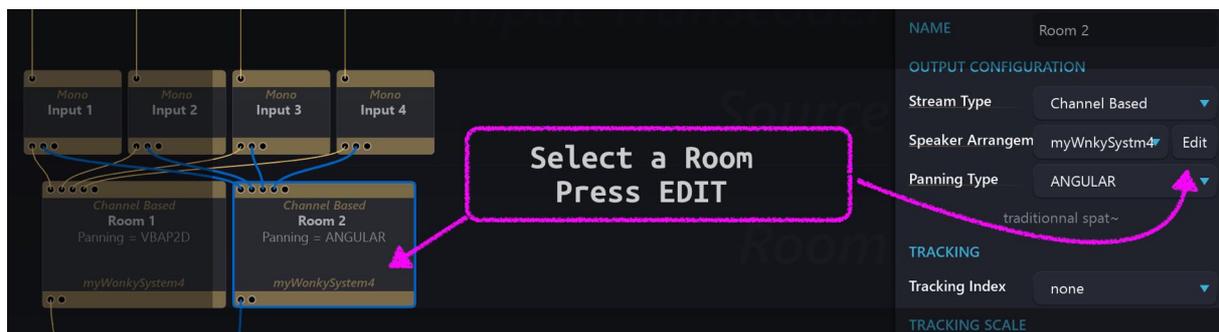
The speaker configuration editor, clear channel labelling and the built-in routing matrix system all help to make the process of signal routing, checking and debugging more straight-forward on location, in the virtual mix and in the studio.

## 4.5

# Custom Speaker Configuration

When you find yourself working on a multi-speaker installation which is not covered in the SPAT speaker arrangement presets, you can still get great results. This is because Spat Revolution has an intelligent **Custom Speaker Configuration** editor where a model of the sound diffusion system you are actually using can be defined and stored into the list of presets.

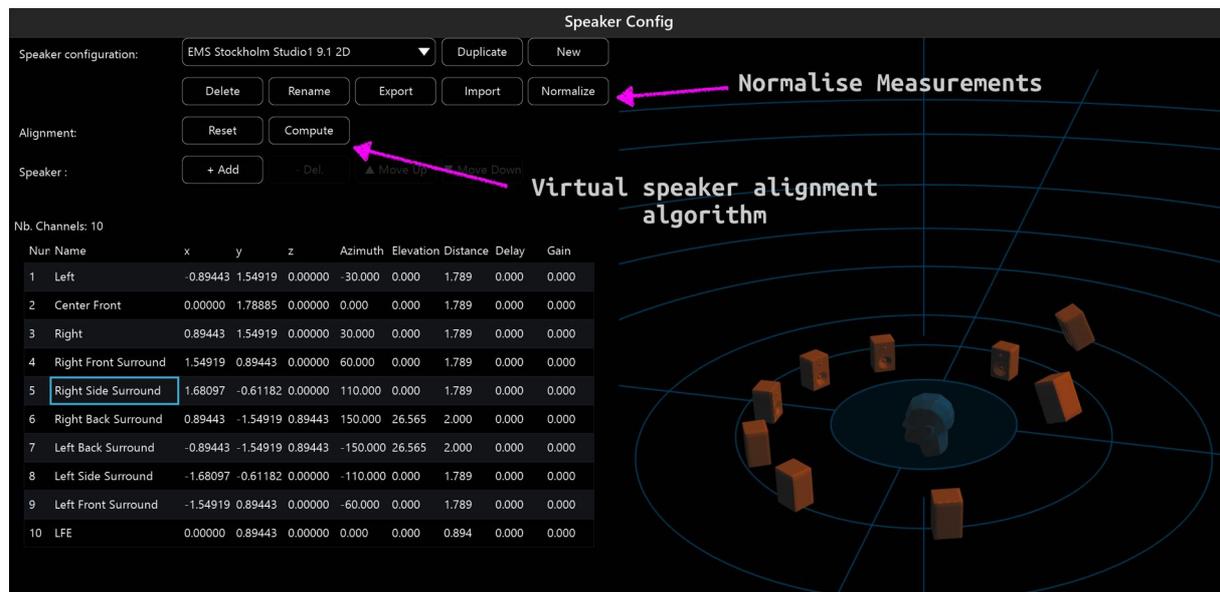
Open the **Custom Speaker Configuration** editor by clicking on the EDIT button of a Channel Based room, by the *Speaker Arrangement* pull down menu. All Channel Based speaker layout presets are stored, duplicatable and editable in the **Custom Speaker Configuration** editor.



Spat Revolution can accept real world measurements which you have entered manually, and can save this system as a speaker arrangement preset to be used in all Channel Based contexts, such as simulating that exact physical system in a virtual room, binauralising or transcoding onto that channel based system from an Ambisonic source.

To create a custom preset, first of all, make a NEW or DUPLICATE an existing preset from the menu. *Add, Delete* and *Move* speakers. Importantly, **LABEL** speakers. You can enter Angles, to map the model in Polar space, and Spat will calculate Cartesian distances. If you have X,Y,Z Cartesian co-ordinates measured from the origin of

(0,0,0) at the Listener Position, then Spat Revolution will take care of the polar co-ordinates.



In addition, it can also use the measurements to compute the optimum delays and gains for better spatial results on a custom speaker configuration. This is an advanced speaker management technique made easily accessible by a single press of the **Compute Speaker Alignment** button.

When you have defined a custom speaker arrangement and press this button, Spat revolution will create virtual speakers for the ones that are not at equal distances to listener. This becomes important when using speakers arrangements with sweep spot centric panning methods as the algorithm will try to 'fill in the gaps' and work out the best speaker weighting internally that will normalise and balance the sound field towards an optimum decoding for the speaker and channel based format being actually used.

The **Normalise** button is also useful, as it will remap real measured distances into a normalised 'absolute' distance measure that should improve the perception of virtual sources' *Distance* variable which is calculated as a simulated distance measure between listener position and virtual speakers in a channel based configuration.

## Before Alignment Compute

Speaker configuration: myWonkySystem (copy) Duplicate New

Delete Rename Export Import Normalize

Alignment: Reset Compute

Speaker: + Add - Del. ▲ Move Up ▼ Move Down

Nb. Channels: 6

Nuri	Name	x	y	z	Azimuth	Elevation	Distance	Delay	Gain
1	Ch 1	2.00000	-3.00000	0.00000	146.310	0.000	3.606	0.000	0.000
2	Ch 2	3.00000	0.00000	0.00000	90.000	0.000	3.000	0.000	0.000
3	Ch 3	0.00000	2.50000	0.00000	0.000	0.000	2.500	0.000	0.000
4	ch4	-1.50000	1.50000	0.00000	-45.000	0.000	2.121	0.000	0.000
5	ch5	1.50000	1.50000	0.00000	45.000	0.000	2.121	0.000	0.000
6	ch6	-2.00000	-1.50000	0.00000	-126.870	0.000	2.500	0.000	0.000

## After Alignment Compute

Speaker configuration: myWonkySystem (copy) Duplicate New

Delete Rename Export Import Normalize

Alignment: Reset Compute

Speaker: + Add - Del. ▲ Move Up ▼ Move Down

Nb. Channels: 6

Nuri	Name	x	y	z	Azimuth	Elevation	Distance	Delay	Gain
1	Ch 1	2.00000	-3.00000	0.00000	146.310	0.000	3.606	0.000	0.000
2	Ch 2	3.00000	0.00000	0.00000	90.000	0.000	3.000	1.764	-1.597
3	Ch 3	0.00000	2.50000	0.00000	0.000	0.000	2.500	3.221	-3.181
4	ch4	-1.50000	1.50000	0.00000	-45.000	0.000	2.121	4.325	-4.607
5	ch5	1.50000	1.50000	0.00000	45.000	0.000	2.121	4.325	-4.607
6	ch6	-2.00000	-1.50000	0.00000	-126.870	0.000	2.500	3.221	-3.181

A detailed tutorial on advanced scripting of Custom Speaker Configurations using the Python language is available on line at the Flux:: support website.

## 4.6

# The Ambisonic Method

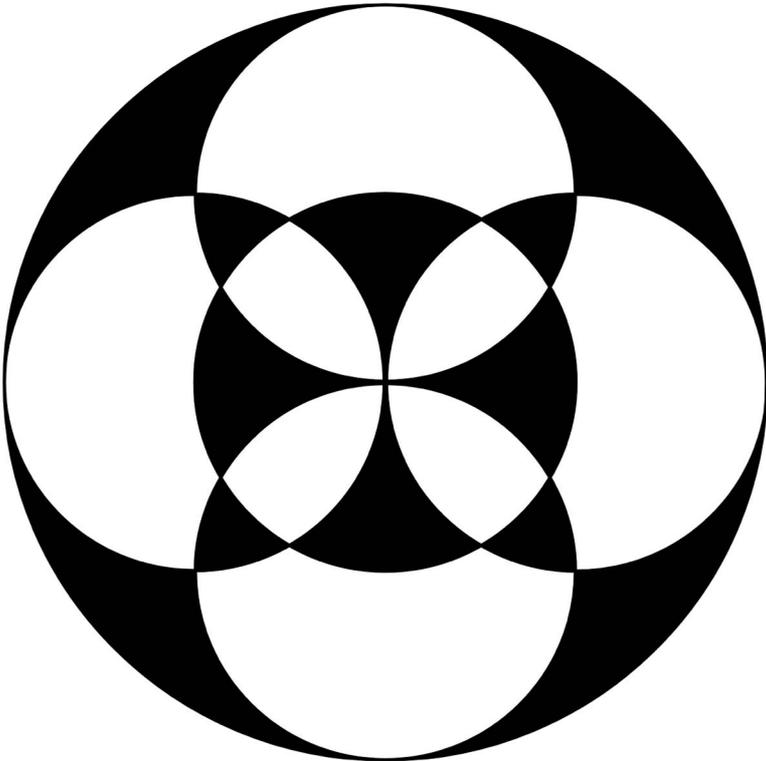
One of the challenges facing spatial audio designers is how to spatialise and mix when the full installation is not available to mix on. The other related problem is how to re-spatialise a virtual scene mixed for one type of speaker layout, onto a different one. Spat Revolution can provide solutions to these problems.

In order to get a rough idea of how a spatialised scene might sound on a particular speaker configuration - including the gaps between the speakers, and the characteristic of a chosen panning type - then the binaural 3D headphone monitor module can be used. It can give a headphone representation of a channel based virtual scene. It has its limitations though so the binaural monitor module should not be relied upon for all mix decisions. It is there as a quick way to 'dip' into the virtual scene but only has the spatial resolution of the number of speakers in the channel based stream.

A better way to get a high resolution 3D audio image would be to create a binaural room in parallel with a channel based room. This will *binauralise* the sources and their virtual acoustics in a room with no speakers - this is the best binaural experience Spat Revolution provides - although of course it is still limited to headphones only (see section 5.1).

Perhaps the most practical solution, comes from the ambisonic technology which is at the heart of Spat Revolution, and in fact behind the binaural rendering just described. Ambisonics, as we are about to discover, is an **encoded** audio format that gives us the possibility to **decode** consistent 3D spatial information onto any channel based / speaker compatible format. Because Spat Revolution is fundamentally a High Order Ambisonics engine, it gives a designer the possibility to keep working on the spatial parameters of a complex virtual audio scene and monitor on whatever speaker or headphone setups she might have access to - such as binaural, 5.1 surround, simple Quad, stereo and so on. The spatialisation should translate

more or less intact after decoding. Of course, there's still tuning work to do in practice, but it is much better than imagining how things are going to sound, and further more - is a very interesting application of an audio technology with quite a legacy behind it. Time for a quick introduction to Ambisonic technology.



## 4.7

# Introduction to Ambisonic

Simply put: Ambisonic is a specific method for creating, capturing and playing back spatial audio. It is radically different from other surround techniques as the technology is capable of reproducing a spherical representation of sound where the directional information of a source is located in a 3D soundfield.

Ambisonic is also both a recording and a spatial synthesis technique, where one can capture the full environment in 3D sound through the use of so called A or B-format microphones such as the: Soundfield SPS200, Røde NT-SF1, Sennheiser Ambeo, Coresound TetraMic and more. Alternatively, a sound field can be synthesised from any mono, stereo and multichannel sources to Ambisonic, constructing a virtual 3D sound environment by placing the sources at locations in a virtual 3 dimensional field.

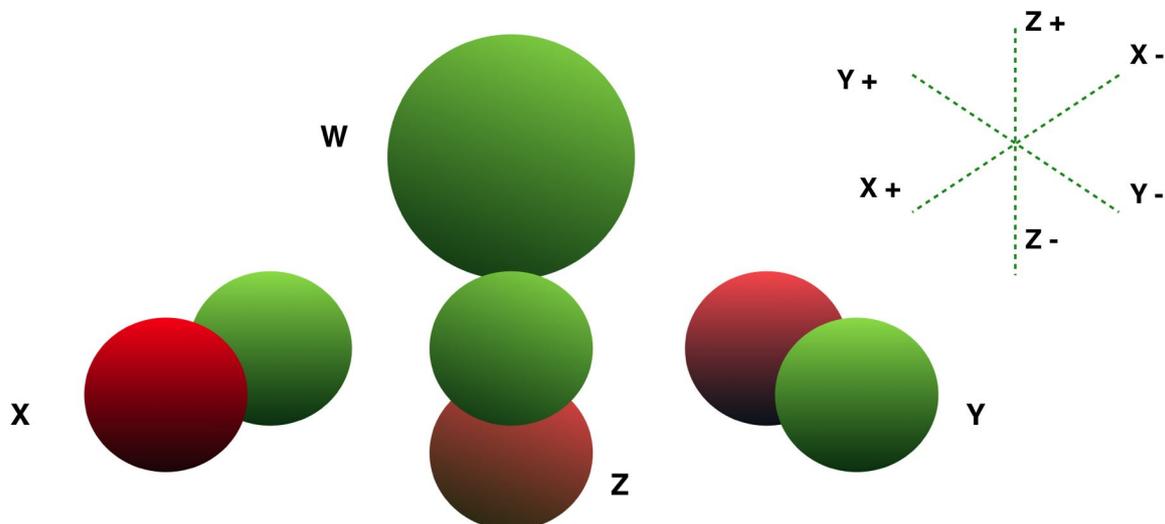


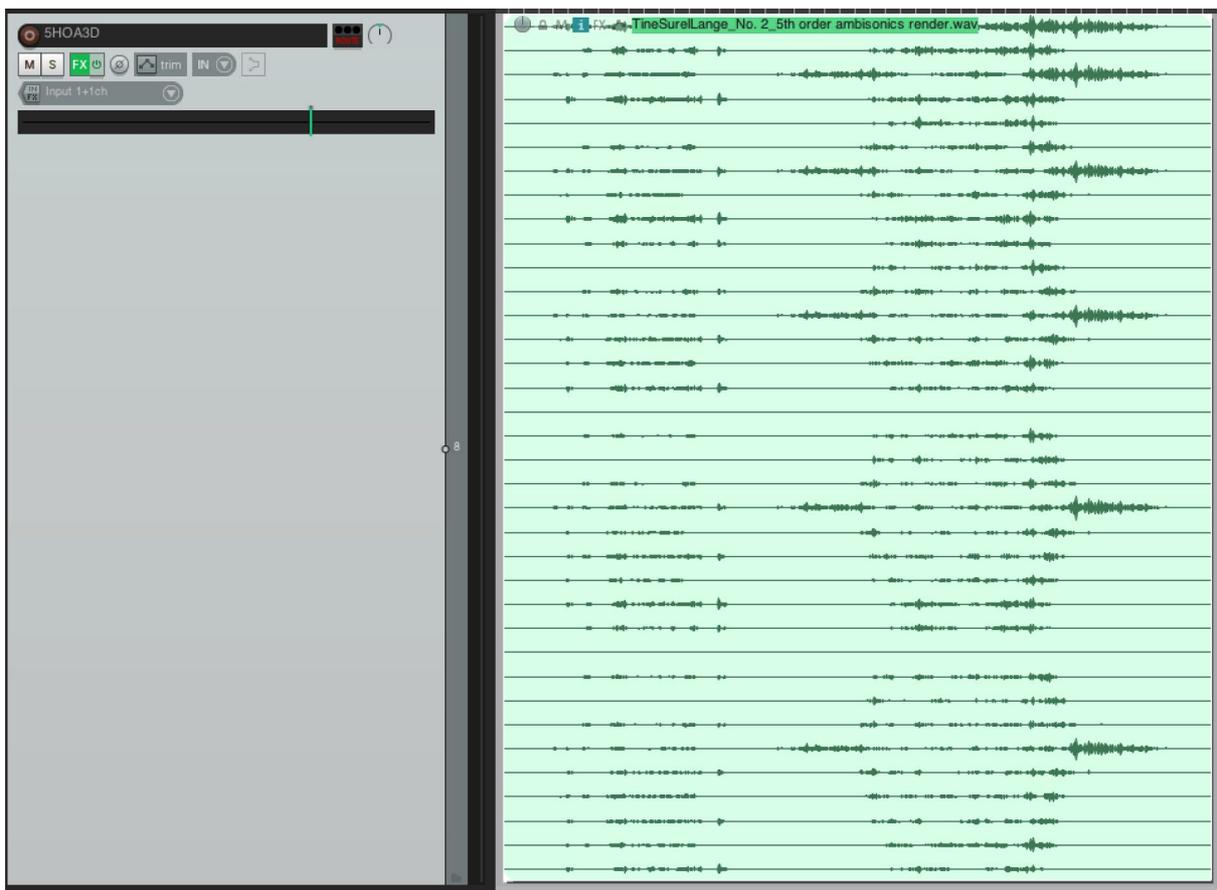
Fig.1. Ambisonic 1st. order WXYZ spherical components. Pressure and gradient polar patterns

In the simplest form of Ambisonic - the 1st order ( also called **B-format**) - only 4 channels is needed to represent a full 3D sound. The 4 channels or spherical components W, X, Y and Z resemble the pressure patterns found in an omni microphone (W) and three figure-of-8 microphones for left/right (Y), front/back (X) and

up/down (Z) as depicted in Fig.1.

Ambisonic as opposed to other surround and spatial techniques and methods does not carry a speaker signal. It is an **encoded** audio signal that has to be **de-coded** to the speaker signals. This encoding / decoding scheme has the advantage of being very portable and flexible since one is not bound to a specific speaker setup. i.e you can have your ambisonic mix played on a number of speaker setups, for instance Quad, headphones (binaural), 5.1, 6, 8, 7 speaker etc. based on the chosen decoder.

When Ambisonic is played back on speakers all the speakers contribute to the directional content, what one is hearing is not the sound coming from a specific speaker but from a specific direction.



**Overview of a 5 HOA 3D Ambisonic File created by Tine Surell Lange**

Ambisonic was originally developed by the late British mathematician and sound engineer Michæl Gerzon and others in the 1970s. Although it was a commercial failure at the time, this very powerful spatial technique has since been advanced greatly by a number of composers, sound designers and researchers. With the introduction of Virtual Reality, fast decoders and related technology, Ambisonic is getting a new renaissance being a perfect format for such applications.

If you want to learn more about Ambisonic and its mathematical foundation here are some good starting points:

[https://www.researchgate.net/publication/280010078\\_Introduction\\_to\\_Ambisonics](https://www.researchgate.net/publication/280010078_Introduction_to_Ambisonics)  
<http://flo.mur.at/writings/HOA-intro.pdf>

## 4.8

# Rendering to Ambisonic Format

As you gain a better understanding of how Spat Revolution works and the spatialisation technologies that underpin it, you will see that one advantage it has over other surround audio software is that the end format of your spatial sound composition, does not necessarily need to be fixed. In fact, when working with *Spat Virtual Rooms* which can output Ambisonic formats you may decide to encode a virtual scene to disk as interleaved Ambisonic audio, which is independent of any fixed speaker setup. You are then free to decode the Ambisonic information into any channel based format at any given time with accurate and consistent results.

For example, an Ambisonic composition could be decoded live to any given speaker system using Spat Revolution for the live control and spatialisation of sources in realtime - or a pre-composed Ambisonic composition could be decoded to disk instead. In the latter case, it is possible to render versions of the Ambisonic encoded spatialisation decoded for different channel based output formats in Spat Revolution. The decoded result will then be normal, multi-channel audio files suitable for playback on a specific speaker configuration, such as 5.1 or 7.1 or whatever deliverable format was requested. Such decoded multi channel files can be played independently of Spat Revolution on location using a normal multichannel playback system.

**★ If you are delivering multichannel files to another engineer or client always label the stems as clearly as possible!**

## 4.9

# Multichannel Multiformat Audio Data

SPAT Revolution mixes, reverberates and spatializes audio from all types of microphones, arrays, high order Ambisonic captures, and from any type of pre-produced stems, supporting a wide range of input and output audio stream formats.

When mixing in Spat Revolution - whether live on the system or off location - it is possible to listen, decode, transcode or record a virtual scene for a specific Channel based speaker arrangement or a high definition binaural headphone mix or to various Ambisonic formats.

A designer could render to all these output types simultaneously if the computer is powerful enough. As we go on, we will learn more about these Spat Revolution workflows, the technology behind them and how to use them in different contexts.

# 5. Spatialisation Technology

In SPAT Revolution multiple 3D-spatialisation technologies can be combined and mixed in different formats and rendered simultaneously. Let's start with an introduction to the immersive audio technology known as Binaural.



## 5.1

# Binaural

The expression “binaural technology” covers various methods of sound recording, synthesis and reproduction which can render 3D spatial audio content over headphones. For instance, binaural field recordings can be made by placing miniature microphones in the ear canals of a listener or of a dummy head (aka '**Kemar**') and when played back over headphones such recordings can produce an authentic immersive auditory experience with enhanced spatial aspects.

Recent advances in signal processing technology has made it possible to synthesise binaural signals without the need of microphones. Using binaural synthesis, a sound can be arbitrarily positioned around a listener synthesising the sensory experience of extended spatialisation. Like some other two channel formats such as Mid-Side Stereo, binaurally encoded audio recordings are not compatible with stereo speakers. If a binaural encoded audio file is played on a normal stereo setup, audio will be heard, but it won't sound good.

*△ It is important to point out to a client who might be new to binaural monitoring, that binaural files should only be listened to on a good pair of headphones.*

## 5.11

# HRTF

HRTF is an abbreviation for Head Related Transfer Function. The function is a mathematical model of the filtering effect caused by a listener's own head, external ear and torso. This filtering plays a significant role in the way we localise sounds around us and is unique to every individual.

When synthesising binaural monitoring, a perfect result could be attained by rendering through the exact HRTF that matches the body filtering effect of an individual. In practice this is not easily done, so Spat Revolution offers many different choices of pre-analysed HRTF profiles which you can apply for monitoring and encoding binaural audio. You can manage the selection of HRTF profiles in the Spat Revolution Preferences where you will find a number of different profiles including the option to load your own HRTF. The default HRTF is the Kemar dummy head model, which is often used as an all round generic head and shoulder filter.

## 5.12

# HRTF Profiles

The included HRTF profiles in Spat Revolution are taken from a number of large scale laboratory research projects where measurements were taken on many individuals<sup>1</sup>. The chances are that one person's ears may sound more natural to you than others. For a quick way to monitor binaural, you should try to find a profile that you feel most comfortable with when monitoring your virtual scene on headphones. If you are providing a 3D in-ear monitor mix for a performer or a visitor to an installation, try to find an HRTF that suits them best. This can be fun. If you are not comfortable with listening through someone else ears - which is understandable - you could look into creating a personalised HRTF from your own head and upper torso measurements. There already exist a number of services that can create HRTF profiles taken from laboratory measurements. If you decide to do this, for yourself or someone else, then you can add the personalised profile to the list in the HRTF Manager. In fact you can import any HRTF in [SOFA](#) format to the Spat Revolution binaural encoding list, making Spat Revolution a very flexible solution for binaural monitoring and rendering.

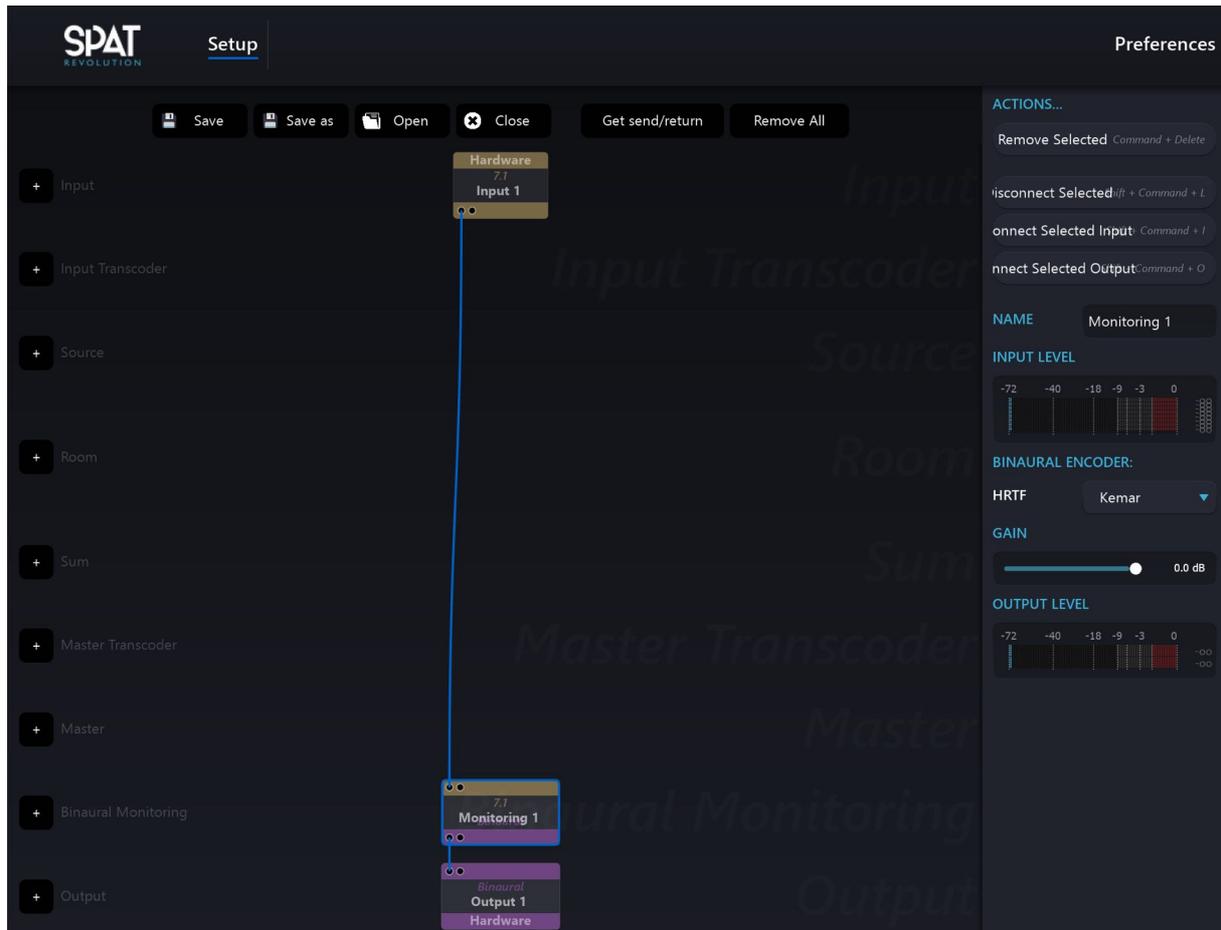
*△ An imported HRTF profile should be in SOFA format and should match the sample rate for your project. It is preferable to use a "SimpleFreeFieldSOS" IIR type of HRTF*

---

<sup>1</sup> The profiles come from the "LISTEN" , "CROSSMOD" and "BiLi" databases.

## 5.13

# Binaural Monitoring Module



In the *Environment Setup* of Spat Revolution, you will find a module dedicated to Binaural Monitoring. Its purpose is to monitor any kind of speaker setup using headphones and binaural encoding. This can give you an impression of how your spatialisation might sound on a particular channel based system when you are off location.

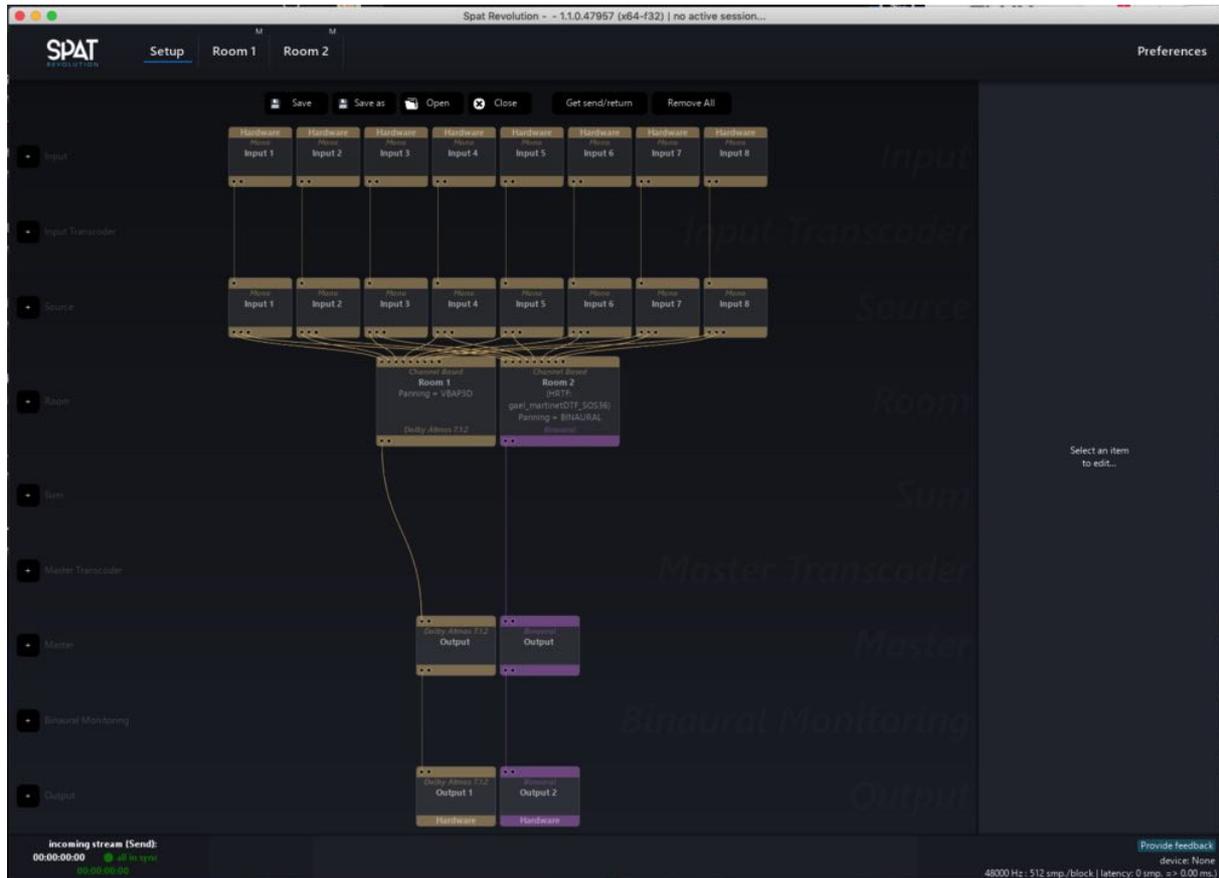
You can add a Binaural Monitoring Module by clicking on the + icon of the Binaural Monitoring row towards the bottom of the **Environment Setup** graph. The module is very simple to use. It will automatically detect the type of channel based audio stream you connect into it.

The Binaural Monitoring module works by virtualising each speaker, not each source, so any real world speaker phenomena will be reflected in the binaural rendering. For example, a virtual source positioned in the centre between two virtual speakers will be rendered with the same 'phantom speaker' in the binaural monitoring as in the physical world, because there is no virtual speaker at the centre point either.

To listen to the binaural stream on headphones, you should select the HRTF you would like to use for the encoding, and connect the output from the module to a dedicated output module at the bottom of the graph. The output should be routed to the headphone monitor outputs of your audio hardware.

## 5.14

# Binaural Rooms



★ For the best Binaural Monitoring set up two rooms, one all Channel Based going to speakers and one Binaural for a 3D headphone mix.

The Binaural module is great for easy channel based system monitoring. But it is not the best binaural experience we can provide with Spat Revolution. The best way to work with binaural encoding in Spat Revolution is to use a *Virtual Room* which is binaurally encoding each virtual source at its exact position in a virtual space, rather than virtual speakers. In fact, in a Binaural Room there are no virtual speakers being modelled. Instead each source's direct sound and the reverberation it creates are modelled and synthesised binaurally for *each individual source*.

This advanced processing can result in an excellent binaural experience; more precise and natural sounding than the Binaural Monitoring module. It is the preferred method to use when rendering Binaural content to disk. To do that, you simply need to connect the binaural stream from the room directly to a (stereo) SPAT Return path back to your DAW. You could record and listen to a Binaural Room by using two output modules, one to the recording route, and one to a headphone output.

## 5.2

# Transaural

Although we said earlier that Binaurally encoded files are not compatible with stereo loud speaker systems - well, it turns out that in *Spat Revolution* there is a way to play binaurally encoded signals on loud speakers if necessary. You can do this by transcoding a binaural stream into a Transaural stream either at the *Input Transcoder* stage of the **Environment Setup** graph or at the *Master Transcoder* stage.

The term *Transaural* refers only to the transcoding of binaural signals into loud-speaker-compatible signals. The reproduction mode is also known as "cross-talk cancellation" as it uses this very process to establish the loudspeaker-compatible signals.

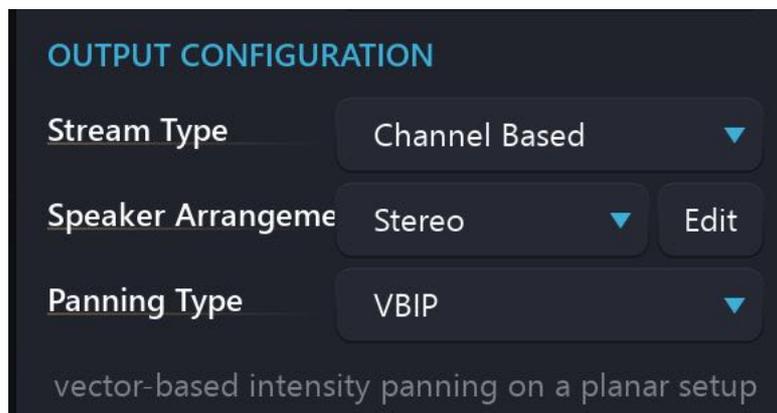
△ *To successfully achieve cross-talk cancellation, it is assumed that the listener will be placed at a particular position with respect to the loudspeaker pair, or the spatial information will not be properly perceived.*

An optimum listening position such as that needed for Transaural decoding, is also known as '**the sweet spot**'. It is a fundamental concept to be aware of as we move into the next section; *Channel Based* streams and panning algorithms.

## 5.3

# Channel Based Streams

Spat Revolution is setup to work with a number of different Input and Output formats. The different IO configurations are represented throughout the *Environment Setup* as *Stream Types* .



We have already covered the two channel Binaural stream type for monitoring and final encoding into Binaural format. One of the other important Stream Types is referred to as *Channel Based* . This stream can range from a single channel mono to a multichannel audio stream of up to 64 signals which flow as a perfectly synchronised group through the signal graph defined in the *Environment Setup* . The channel count of the stream is set by the choice in the *Speaker Arrangement* pull down menu of a module. A change to the speaker configuration here, alters the channel count in and out of a module, depending on its context in the signal graph. When you connect Channel Based modules together in SPAT they automatically inherit the Speaker Arrangement and Channel Count from the stream type at their connected input.

Channel Based audio streams - when connected to a hardware Output module - will render the spatial composition on speakers connected to the physical outputs of your audio hardware. SPAT is expecting the loud speaker system specified in the Speaker Arrangement of the Output module. If the real loudspeaker arrangement

does not correctly match the speaker arrangement model or there is a mistake in your routing somewhere, then the spatial sound image will be compromised.

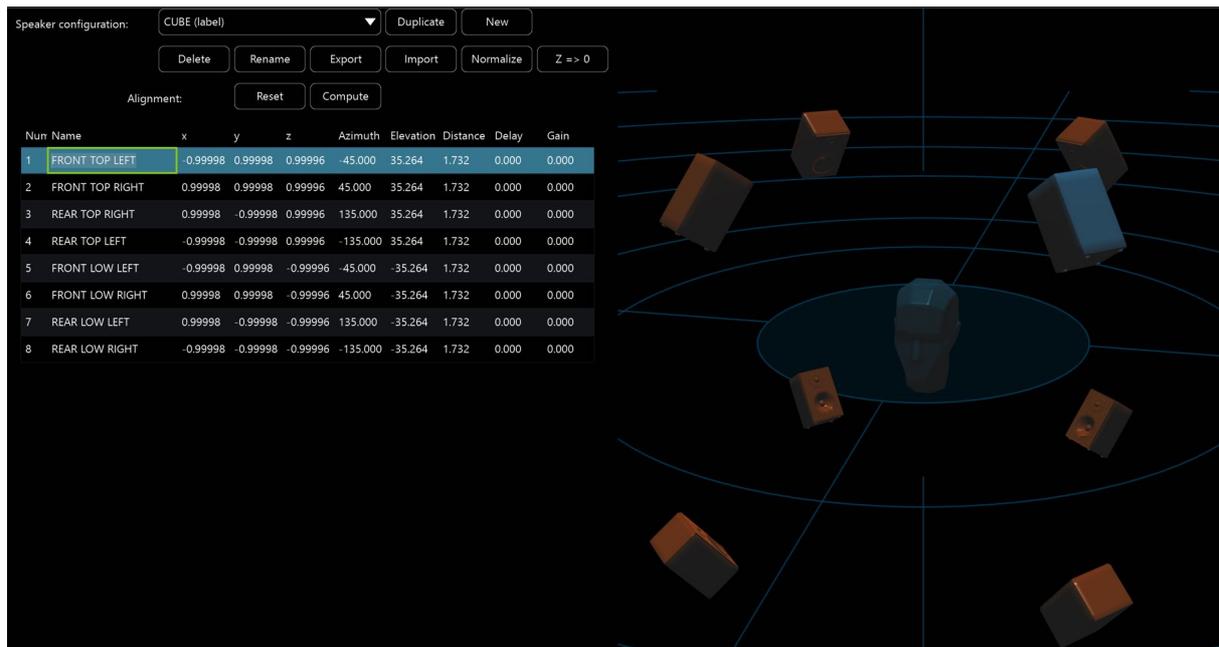
△ *Each output channel must be routed to the correct speaker*

The golden rule for spatialisation using Channel Based audio is that each rendered channel must be connected only to its corresponding sound emitter in the destination system. An exact correlation from the model to the physical system is assumed by the calculations inside each of the panning algorithms. If the installation is not right for some reason, a listener may experience something with spatial aspects, but not with the intended quality. To correctly install and tune a multichannel sound system is one of the more challenging aspects of working with spatial composition and performance systems. So many things can compromise the spatial image, and sometimes it's hard to tell by ear.

In order to try and simplify this process in practice, certain labelling conventions are used by engineers and designers to identify which speaker belongs to which channel. When channel counts are high, angular positioning and distance measurements are used to identify the correct speaker routing. In Spat Revolution's *Speaker Configuration* editor, you additionally get a 3D graphical rendering of the sound system, where you can select speakers which will be visually identified so you can label them clearly. These labels will be saved into a speaker arrangement profile so you can get some more consistent reference points when routing.

Left	Centre	Right
Left Surround	LFE	Right Surround
Left Back Surround	Center Back Surround	Right Back Surround
Top Back Left	Top Front Center	Top Back Right
Left Surround Rear	VOG	Right Surround Rear
Top Front Left	Back Center	Top Front Left

Some common Speaker Channel naming conventions



The process of matching the correct channel to the correct speaker is absolutely vital to the successful rendering of the spatial composition from a SPAT *Virtual Room* into a physical space. Further critical points to a successful project are the choice of panning type in a *Virtual Room* and consideration of *the sweet spot* for listener positioning. As it is such an important and often misunderstood idea, let's take a look at that topic before going further.

## 5.4

# Listener Position

The impression of positioned audio which is rendered on a speaker array is generally successful when a listener is situated in the optimum position to perceive it - the so-called *Sweet Spot*. Thanks to the popularity of stereo sound, people tend to know that if you want to hear a good a stereo image, you should place yourself in front of the two speakers and stand somewhere in the middle. That's a good way to describe the *Sweet Spot* to an audience or client. In more complex speaker arrangements the *Sweet Spot* is usually considered to be the area that all the speakers (or sound emitters in a virtual space) are pointing to. It has the cartesian co-ordinates of  $(0, 0, 0)$  in relation to the speaker positions. In Spat Revolution the optimum listening area is represented by the dummy head and the inner circle that surrounds it.



The dummy head indicates the *Listener Position*. In SPAT the *Listener Position* is actually a bit more useful than just an indication of the Sweet Spot. In general it is useful for getting some bearings in the spatial composition. Knowing which direction the listener position is facing, helps you understand the spatial image and place sources correctly. For example, a train audibly approaches from the left in an ambisonic field recording you are working with as a source, but it visually approaches from the right on the video footage you are editing to. You can use the listener position as a reference point to transform the field recording correctly so that it correlates to what's happening on screen. Also, when working in 'stage' oriented spaces, the listener position in a SPAT Room will help you compose a scene with the correct relationship to the front, back and sides of the space. It also provides a method for giving a sense of distance to a sound source by placing virtual sound objects closer or farther away from the listening position. The internal distance perceptual cues, such as air absorption, doppler and gain drop are all calculated in relation to the *Listener Position*.

**★ Some Panning and Stream Types have wider Sweet Spots than others and some do not have a Sweet Spot at all.**

It is worth noting that there are some panning algorithms in SPAT that are **not** *Sweet Spot* dependent. These are more suitable for use on arbitrarily positioned speaker distributions. We will get to them shortly.

△ *The dummy head is only visible for sweet spot dependent panning types*

★ The Listener Position can be moved in a Spat Revolution room.

The screenshot displays the Spat Revolution software interface. At the top, there are controls for 'Nebula Alpha' (set to 0%), 'Speaker Alpha' (set to 100%), 'Display Output' (Room 1), 'Back Color' (Black), and 'View' (Top/Front). The main 3D view shows a room with a central listener (blue head) and several speakers (orange boxes) arranged in a circular pattern. The bottom panel is divided into three sections: 'Listener', 'Tracking', and 'Output list'.

**Listener** section contains six knobs for: Listener X (0.00000), Listener Y (0.00000), Listener Z (0.00000), Listener Yaw (0.00), Listener Pitch (0.00), and Listener Roll (0.00).

**Tracking** section contains a '1.00 m Scale' knob.

**Output list** section includes an 'Edit output config.' button, 'Alignment' options (Reset, Compute), and a table of speaker configurations.

Nurr	Name	x	y	z	Azimuth	Elevation	Distance	Delay	Gain
1	Left	-1.00000	1.73205	0.00000	-30.000	0.000	2.000	0.000	0.000
2	Center	0.00000	2.00000	0.00000	0.000	0.000	2.000	0.000	0.000
3	Right	1.00000	1.73205	0.00000	30.000	0.000	2.000	0.000	0.000
4	LFE	0.00000	2.00000	0.00000	0.000	0.000	2.000	0.000	0.000
5	Left Back Surround	-1.87939	-0.68404	0.00000	-110.000	0.000	2.000	0.000	0.000
6	Right Back Surround	1.87939	-0.68404	0.00000	110.000	0.000	2.000	0.000	0.000
7	Top Front Left	-0.86603	1.50000	1.00000	-30.000	30.000	2.000	0.000	0.000

In certain advanced situations which might combine position tracking systems with realtime binaural audio it is even possible to transform the *Listener Position* in SPAT. One application of this might be to give the sensation of getting closer to a sound emitter inside a virtual scene for a headset wearing participant at an interactive VR installation (see section 6.61)

## 5.5

# Panning Algorithms

There has been a lot of scientific and artistic work to discover new methods for the electronic reproduction of audio signals to achieve the sensory impression that sounds are positioned in the space between speakers, rather than only coming from a single speaker. One of the underlying mechanisms that makes it possible to position sounds or make it seem like sounds are moving around the listener, is conventionally referred to as *panning* .

Broadly speaking, panning is the result of an algorithm - sometimes called 'the panning law' - that is used to calculate the amplitudes of signals sent to two or more identical loudspeakers (or virtual sources) arranged in a particular spatial configuration. In stereo - which we could say is the lowest resolution of speaker array - a simple panning algorithm can make the sound of a voice, for example, seem to be placed in the air between two speakers - as if there were an invisible speaker there. This illusion works up to a point, and breaks down if the speakers are too far apart. With only two speakers in the system, the panning law is a simple graph which has become quite standardised over the years. Things get more complicated when panning over 2D or 3D speaker setups and methods for spatialising on multichannel systems are not really standardised yet. Many algorithms and standards have been advanced by the audio engineering industry, composers and academics. Essentially, when we start working with multiple speakers in different configurations, things get more complex for the panning algorithms - goals have also expanded over time, with spatial sound design able to reach beyond simple point source panning into fully immersive audio experiences that can be hyper-realistic or totally synthesised. Many practitioners will have their preferred methods, but really it is up to the spatial sound designer to choose methods that seems to give the best experience for the listener on any given type of setup. There's never been just one way to spatialise sound. And with a powerful system like Spat Revolution, panning methods can even be combined in novel ways to achieve high quality immersive audio experiences.

**Spat Revolution lets you explore some of the most advanced panning algorithms for surround, immersive 3D or ad-hoc sound systems.**

In SPAT you will be able to explore some of the best panning algorithms for multi-speaker setups. You can apply them in realtime and identify their characteristic differences by ear. Trying them out in realtime on a setup will help you select the panning algorithm that is best suited for your particular project and material.

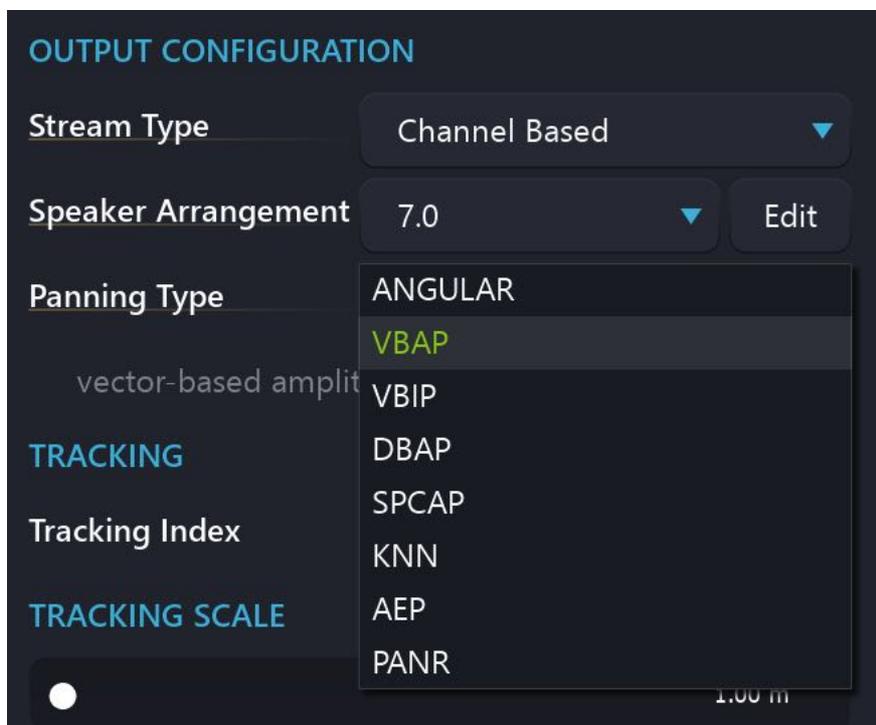
Although there are technical aspects to be interested in and aware of, you are still invited to be creative and use your ears when deciding which are the right panning types for your project and intended audience.

**★ Try using more than one Spat Room to use different panning algorithms for sound material that has different sonic qualities**

## 5.6

# Room Panning Types

In the descriptions that follow, 2D refers to a horizontal speaker configuration on a plane, with no elevation speakers. 3D refers to any type of speaker configuration that features elevation. These panning types are set in the output configuration pull down menu of a Spat Room module.



## 5.61

# Vector Base Amplitude (VBAP)

Vector Base Amplitude Panning has become one of the more standardised methods for multichannel spatialisation. It can reproduce on a 2D or 3D configuration. Its sound is characterised by clearly localisable virtual sound sources. Multiple moving or stationary sounds can be positioned in any direction over the speaker array using this method. In theory VBAP can be used on an unlimited number of loudspeakers and can even be reliable on relatively asymmetric setups.

### How does it work?

Three important dependencies to consider when using VBAP;

1. Speakers must be placed *around* the listening position.
2. Speakers ideally should be equidistant from the listening position<sup>2</sup>
3. 2D Speakers should be on the same horizontal plane as the ears
4. VBAP works best when the listening room is not very reverberant

Traditional VBAP works by manipulating the gain of the signals being routed to the two (in 2D) or three (in 3D) closest speakers to a virtual sound source. VBAP relies heavily on an accurate speaker arrangement model to do this. It triangulates gain vectors mathematically in order to render a virtual object in the physical space and achieves its characteristic 'sharp' focus by using only a few speakers closest to the virtual source location. Additionally it is possible to uniformly extend the traditional VBAP pair-wise (or triplet-wise) speaker picking and activate more of the sound system, effectively diffusing the relationship between individual speakers and sounds using *spread* .

★ **Widen a VBAP point source by increasing the *Spread* source parameter.**

---

<sup>2</sup> See section 4.5 the speaker Alignment feature can give the impression that the actual speakers are equidistant even when they are not

## 5.62

# Vector Base Intensity (VBIP)

Vector Base Intensity Panning is a similar variation to the VBAP technique. It can also reproduce a 2D or 3D immersive sound field with sharply localised virtual sound sources.

### How does it work?

VBIP was designed to improve on VBAP when calculating the high-frequency (above 700 Hz) localisation criteria. The selection of which speakers to use to render a virtual sound source is very similar to VBAP, only the gain calculations differ.

The same 4 dependencies mentioned for VBAP, also apply to VBIP. You will need to listen for quite a nuanced difference between these two panning algorithms. Try to compare how each panning type handles the higher frequency content of your material.

## 5.63

# Distance Base Angular (DBAP)

As we mentioned earlier, there are a few panning algorithms that are **not** *Sweet Spot* dependent. Distance-base amplitude panning is one of them. DBAP is useful in a number of practical situations such as concerts, stage productions, installations and museum sound design where the predefined geometric speaker layouts which immersive sound fields rely on, are not possible to establish.

### How does it work?

DBAP localises sounds towards arbitrarily positioned speakers in a space using a matrix based technique. It calculates signal amplitudes according to the actual positions of the speakers in a space, while making no assumptions as to where the listeners are situated. Speaker tuning and interesting acoustic features in a room should be utilised more when working with DBAP.

**★ Speakers can be freely positioned when using DBAP - look for reflections and reverberations in a room to enhance spatial aspects.**

## 5.64

# K Nearest Neighbour (KNN)

KNN is another panning type that does not depend on a sweet spot to be perceived correctly. It is a version of a 'Nearest-neighbour' interpolation algorithm. These family of algorithms are also used in the fields of complex systems, 3D graphics and network science to name a few. In Spat Revolution you can sonically explore a network of loudspeakers using this panning type and some virtual sound sources.

### How does it work?

An interesting difference between DBAP and KNN is that the user gets manual control over one of the main coefficients in the underlying algorithm. The parameter is called *Nearest Neighbour Spreading* . It sets a maximum limit to the number of speakers that the algorithm can use as neighbours - the parameter becomes available as a continuously variable percentage *for each virtual source* in a Spat room.



What makes this particularly interesting is that different sources can activate less or more of the sound system dynamically and in a very smooth way. For example, one virtual sound source might seem to pop in and out of individual speakers because its *Nearest Neighbours Spread* parameter is set a low percentage. For example, on a 10 speaker arrangement :1-10% will use 1 speaker, 11% to 20% 2 and so on. Another sound source could seem diffuse over the entire sound system, because its spread variable is set to 100%.

**★ Try automating the *Nearest Neighbours Spread* in a relationship with another source property of the same sound source such as room presence.**

## 5.65

# Speaker-Placement Correction Amplitude (SPCAP)

SPCAP is a 3D panning algorithm which takes its inspiration from VBAP. SPCAP selects not just 2 or 3, but any number of speakers to render a virtual source and weights signal gains according to how much each selected speaker is actually contributing to the overall power output of the speaker configuration. Using this method SPCAP guarantees conservation of loudspeaker power output across any speaker arrangement. Its strengths lie in the down-mixing and up-mixing of virtual scenes from very different channel based speaker arrangements, and of being able to render wider sound sources by using more speakers in a smart way.

### How does it work?

The result will still be sweet-spot dependent although it will be a wider listening area. SPCAP inherits some of the dependencies of VBAP to get successful spatial imaging.

1. Speakers must be placed *around* the listening position.
2. 2D speakers should be on the same horizontal plane as the ears

**★ SPCAP panning can do a good job of translating surround audio mixes from one speaker configuration to another.**

## 5.66

# Ambisonic Equivalent (AEP)

In common with the channel based panning types we have covered so far, Ambisonics is a technology that also distributes virtual sound sources in space yet it achieves this in a fundamentally different way. Ambisonics relies on a two step process.

### 1. **Encoding**

Audio sources along with their positional information are wrapped up together using signal mathematics to create encoded Ambisonic audio. Ambisonic scenes are always carried on at least 3 channels of audio. They are not intended to be *listened to directly* they are intended to be *decoded*.

### 2. **Decoding**

Ambisonic audio signals are unwrapped and the positional information contained within them is decoded *specifically* for one type of speaker configuration. What we get is an immersive sound field that should accurately render the original spatial composition in 2D or 3D on the specified speaker configuration.

Keeping these two steps separate has a number of advantages. Primarily, that of being able to record the encoded Ambisonic audio signals independently of any fixed speaker arrangement. On the other hand, it is possible to "fuse" the two stages of the process together resulting in what appears to be the output of a generalised channel based type of panning. That is the AEP panning type in a nutshell.

## **How does it work?**

$\mathcal{A}EP$  has certain computational and ambisonic mixing advantages and exhibits very different behavior from the VBAP/VBIP pairwise approaches. It is up to you to decide whether to work with purely Ambisonic rooms (more about that in the later section) or to use  $\mathcal{A}EP$  as a channel based panning law. Both approaches are valid and could be useful. As we have mentioned a few times already, the choice of panning type depends on what sounds best in the context of your material, your compositional goals and the acoustics of the system you are working with.

## 5.67

# Angular and PanR

These are legacy 2D pan pot laws from the original IRCAM Spat library. They only become available when using 2D channel based streams and are primarily included for backwards compatibility.

### **How does it work?**

Angular and PanR are pairwise amplitude panning essentially the same as VBAP 2D described on the next page. There is a subtle difference however, in the way the panning law changes when moving the source from one speaker to another.

## 5.68

### XY AB

These two Panning Types will only become available when a *Virtual Room* is set to be virtualising a stereo speaker arrangement (see section 6.52) - they are pan laws that are derived from widely used dual microphone techniques for rendering stereo imaging from an omnidirectional scene.

**AB Panning** simulates the recording of the sound scene by a pair of spaced cardioid microphones, pointing laterally at azimuths  $\pm 55$  deg. (elevation 0), with a distance of 17 cm between the two capsules. Also known as **ORTF**.

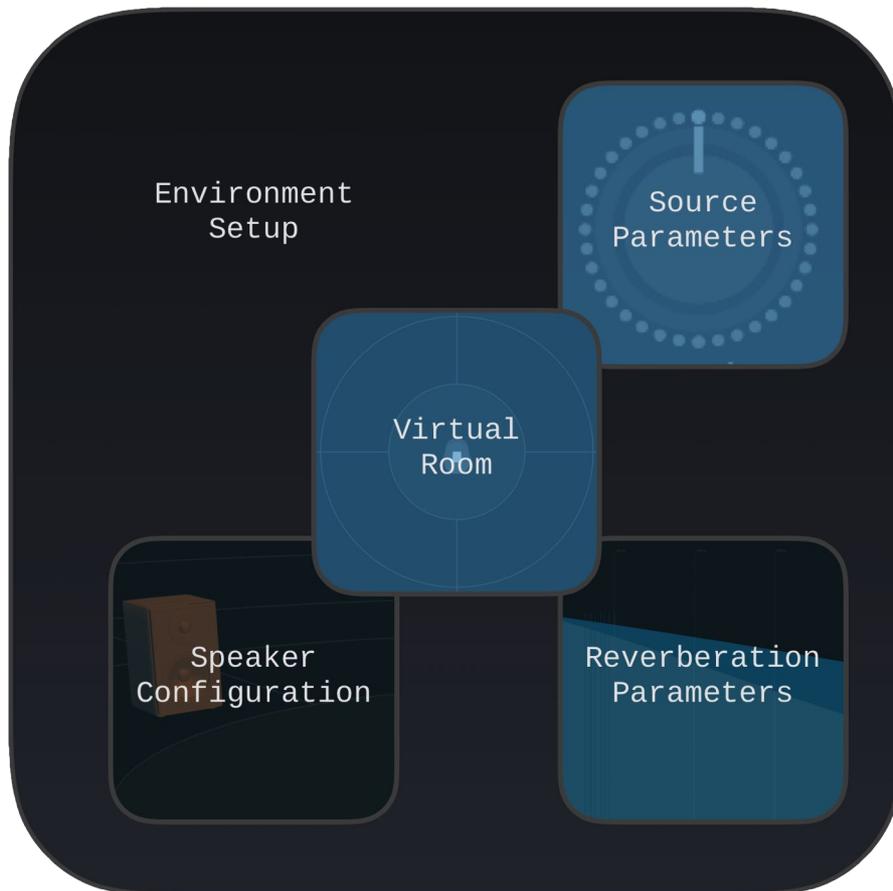
**XY Panning** simulates the recording of the sound scene by a pair of microphones in an XY coincident configuration.

★ The aim is to get the same stereo flavour as these dual microphone tracking techniques. Try them on close miked sources or any mono source, to get a realistic stereo image.

The screenshot displays a DAW interface with a central processing block and a right-hand control panel. The central block is a 'Channel Based Room 1' processor with 'Panning = AB' and 'Stereo' output. It receives four mono inputs: 'Mono Violin', 'Mono Viola', 'Mono Cello', and 'Mono Violin 2'. Each input is connected to a 'Hardware' block. The 'Channel Based Room 1' block is connected to a 'Stereo Output' block, which is then connected to 'Stereo Output 1 Hardware'. The right-hand panel shows settings for 'Room 1', including 'Stream Type' (Channel Based), 'Speaker Arranger' (Stereo), 'Panning Type' (AB), 'Tracking Index' (none), 'Tracking Scale' (1.00 m), 'Gain' (0.0 dB), and 'Output Levels'.

# 6. Spat Environment

This section will go through the Spat Revolution software environment in more detail. We will step through each of the main Graphic Editor views.



## 6.1

# Global Bars



The Navigation bar appears at the top of all views. As well as links to different editor views and the Preferences, it also offers you the possibility to Mute and UnMute Rooms.

The Status and Help bar appears at the bottom of all views. It gives information about the status of audio connections, sample rate, block size and through latency. Some inline help also appears here when the mouse moves over elements of the SPAT graphical user interface. The *Provide feedback* button sends a message directly to Flux support which automatically includes your system information for our support team.

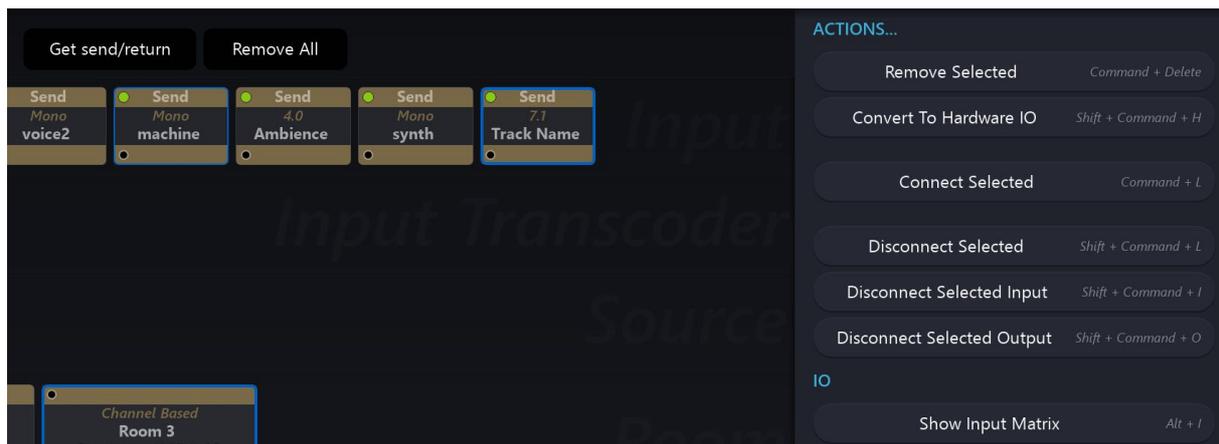


## 6.2

# Environment Setup Editor

This is where you will generally start a project by designing the signal flow graph that you will be working with. *Setup* is also where you manage the loading and saving of projects to disk.

The *Environment Setup* editor is a relatively simple modular environment. The signal flow starts from the inputs at the top of the graph and concludes with the outputs at the bottom. You add modules to rows using the small (+) icon to the left of the window.



Connect or disconnect modules by using *command+click* to select some followed by any of the Actions available in the options panel on the right of the window. Various keyboard shortcuts are also available for each Action. As you connect modules and build up a signal graph, you will see some 'wires' appear which connect modules together. In Spat Revolution these 'wires' represent connections in the signal graph diagram, you do not directly interact with them. It is not a 'patching' type of interface.

## ★ Select multiple modules using drag/lasso selection before an Action

There is no UNDO/REDO paradigm in the signal graph editor at this point. Instead it is advisable to use **Connect/Disconnect Selected** actions to re-structure the signal graph. Try to avoid deleting modules until you are certain that is the correct action.

## ★ Modules can be connected to multiple destinations

It is possible to connect some modules, such as Input or Source, to multiple destinations, as a way of making doubles in a single or doubling a single source into different virtual rooms.



## 6.3

# VU Meters

Throughout Spat Revolution's different editors, you will see a complete set of accurate decibel meters giving you a visual display of all channel activity in an audio stream, whether Ambisonic or Channel Based. They are very useful to see when clipping might be occurring in any of the channels, and to debug signal flow routing in general.

Also, notice how the 'wire' that graphically connects Modules in the Set Up signal graph does not visually change even though it is handling a load of channels.

NAME

Agnete5HOA3D

INPUT LEVEL



INPUT CONFIGURATION

- Stream Type: High Order Ambisonic (
- Dimension: 3D
- Order: 5
- Normalization: N3D
- Sorting: ACN
- Yaw Angle: 0 °

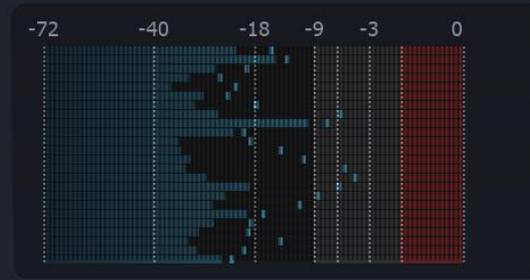
HOA DECODING OPTIONS

- Method: Energy-Preserving
- Type: Basic

OUTPUT CONFIGURATION

- Stream Type: Channel Based
- Speaker Arrangement: Half Sphere | 24. Edit

OUTPUT LEVEL



## 6.4

# Inputs

It might be self-evident, but it's worth pointing out that Spat Revolution itself does not play audio files. It handles the spatialisation and rendering of signal sources routed through it in realtime like some kind of vast spatial sound mixing console.



The top row of the signal graph represents the input sources that will provide the source content for the current project. You add an input module using the (+) icon, and you can remove ones you don't need with the 'Remove Selected' action. The input streams can be of many different formats, the choice of assignable formats will depend on the number of channels each input module represents.

### ★ One Input module can represent any number of audio channels

The second important distinction between inputs, is whether or not it is a **hardware** input receiving an audio stream from a hardware device or a virtual **send** receiving its audio stream from another program currently running on the same machine as SPAT. The latter is done via a Spat Revolution *SEND* plug-in - but before we go into Spat's powerful software signal routing integration, let's focus on the different input formats as these will remain consistent whether the input stream is coming from hardware or from a *SEND* plug-in.

## 6.41

# Mono Input

A one channel audio stream is always treated as a Mono signal. It will appear in a *Virtual Room* as one positionable virtual source with its own directivity and parameters. In many ways, Mono signals are the most straight forward format to work with in a spatial composition. This is because a one channel signal discretely contains all its acoustic and spectral properties without inter-channel dependencies, such as those found in a wide stereo image for example. In practice, such point sources are easier to localise and balance spatially with others.

★ **Mono sources are simple to work with when balancing a spatial mix**

## 6.42

# Two Channel

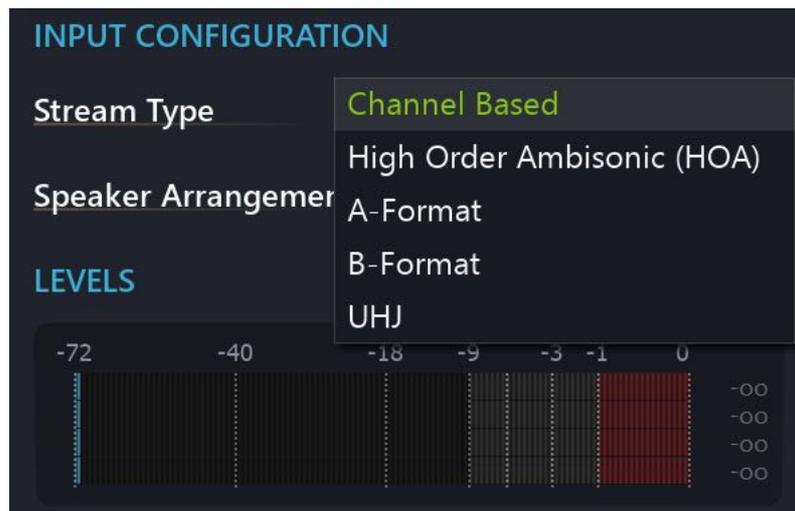
A two channel audio stream will appear in the *Virtual Room* as two mono sources linked together as a group. A two channel audio input will already open a few more choices for disambiguating the configuration. Spat needs to know what format the two channels are in, so it knows how to correctly handle the audio stream later in the signal flow.

- **Channel Based**  
Treated as Normal stereo
- **Mid-Side (MS)**  
Treated as Mid Side encoded stereo
- **Binaural / Transaural**  
Treated as encoded 3D stereo

## 6.43

# Four Channel Input

The next significant channel count that needs disambiguation from the user, is a four channel stream.



A four channel stream could contain the format of a four speaker Channel Based formats ( QUAD , 4.0 , LCRS ) but could also contain different formats of interleaved four channel Ambisonic audio (A-Format, B-Format). You can read more about A-Format and B-Format in the **WORKING WITH AMBISONICS** section of this user guide. The important thing to remember here is that confusing Ambisonic audio and Channel Based audio is a significant mistake, even though you might hear something 'wide sounding'.

*⚠ Do not confuse multi-channel based audio formats with multi-channel Ambisonic audio formats. They may have the same channel counts but are completely different!*

## 6.44

# Multi-Channel Based Input

Any input module configured to represent a stream of multi channel audio can be configured as a Speaker Arrangement format which would require that amount of channels, as a minimum. For example, *DTU 7.1* needs 8 channels, and *DTU 5.1* needs 6. *Auro3D 13.1* needs 14 channels. Unfortunately things can get complicated in practice, as there are a few variations of standardised speaker layouts which have the same number of channels and seem very similar - but need disambiguation. This is important to get right, and will depend a lot on the context of your project and on changing standards in the audio industry. For example, at least four different 7.1 routing standards are to be found 'in the wild' and its important to know which one you are actually dealing with. Often, for example, the so-called 'Low Frequency Effects' channel in cinema surround formats, is not always on the same channel.

**★Try to stick to industry standard channel naming conventions throughout a cinematic surround sound project**

Some common Speaker Channel naming abbreviations

L	C	R
sL		sR
surround Left		surround Left
	LFE	
	Low Frequency Effects	
sbL		sbR
surround back Left		surround back Right

Please refer to Appendix C for special information about handling LFE and Mono Sub Woofer routing in Channel Based applications.

## 6.45 Multi-Channel Ambisonic Inputs

As we have mentioned a few times already, High Order Ambisonics is a technology that encodes sound sources along with full sphere positional information, as complex interleaved audio files that need decoding before they can be listened to on speakers. The lowest order of 3D Ambisonics requires 4 channels, conventionally named as;

- **W** (mono sum)
- **X** ( X axis information)
- **Y** ( Y axis information)
- **Z** (Elevation information)

This is the conventional *B-Format* also known as *First Order Ambisonics*. Some Ambisonic microphones record into a very similar four-channel format called A-Format, which ideally needs transcoding into B-Format. You can use the Input Transcoder Module to do that (see next section). There are a number of options that some Ambisonic encoders set differently to others, and these are sometimes concerned with the actual order of the WXYZ - for example a format called **Ambix** has a different order of channels that carry the spherical components, compared to **FuMa**. It is possible to configure channel sorting and normalisation options in the Input and Input Transcoder modules, and at any point where Ambisonic streams are decoded to a channel based stream. Please contact Flux:: support if you need expert guidance in the area of Ambisonic decoding options, as the topic is very large, and is beyond the scope of this guide.

High Order Ambisonics ( HOA ) needs more channels to contain the complex interleaved Ambisonic domain information. High Orders encode and decode into sharper and more accurate spatial information as the Order gets higher - but the number of channels needed to hold all the 'spherical harmonics' along with the serious computation involved, becomes very complicated very quickly. Fortunately....

## ★*Spat Revolution* makes it easy to work with High Order Ambisonics

Just dial up the Higher Orders using the pulldown menu of an Ambisonic Input module, Transcoder or Ambisonic Room. The output stream will then expand its channels internally to accommodate the higher channel count needed.

3D full sphere HOA channel counts are defined by the function  $(n+1)^2$  (where  $n$  = Ambisonic Order)

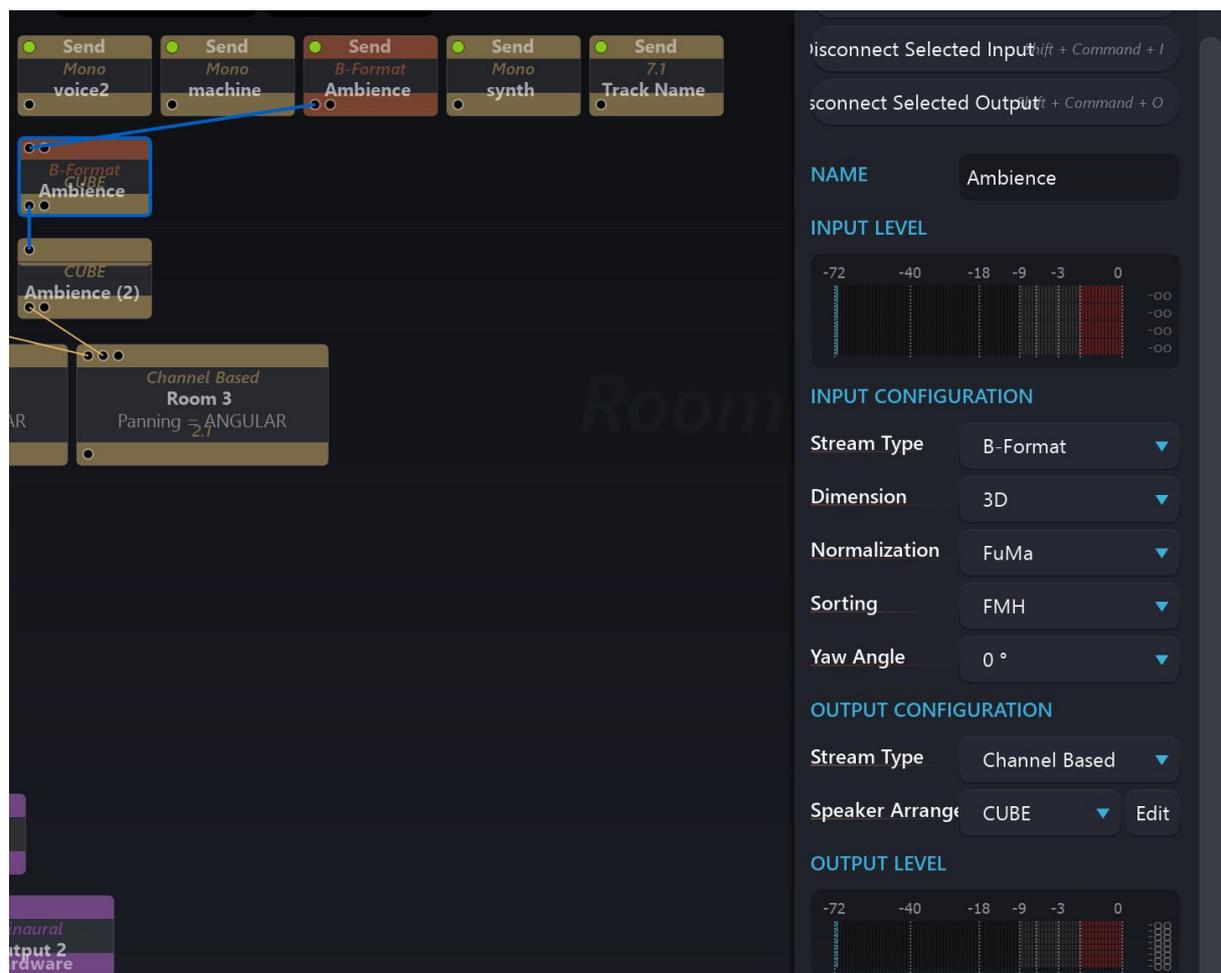
- 1st Order 3D -> 4 Channels
- 2nd Order 3D -> 9 Channels
- 3rd Order 3D -> 16 Channels
- 4th Order 3D -> 25 Channels
- 5th Order 3D -> 36 Channels
- 6th Order 3D -> 49 Channels
- 7th Order 3D -> 64 Channels

*B-Format* can also be encoded without elevation - this is called 2D horizontal and is quite suitable for decoding to configurations that do not have speakers on elevated planes. The 2 dimensional Ambisonic data is encoded in multichannel files with a channel count defined by the function  $2n+1$

- 1st Order 2D -> 4 Channels
- 2nd Order 2D -> 5 Channels
- 3rd Order 2D -> 7 Channels
- 4th Order 2D -> 9 Channels
- 5th Order 2D -> 11 Channels
- 6th Order 2D -> 13 Channels
- 7th Order 2D -> 15 Channels

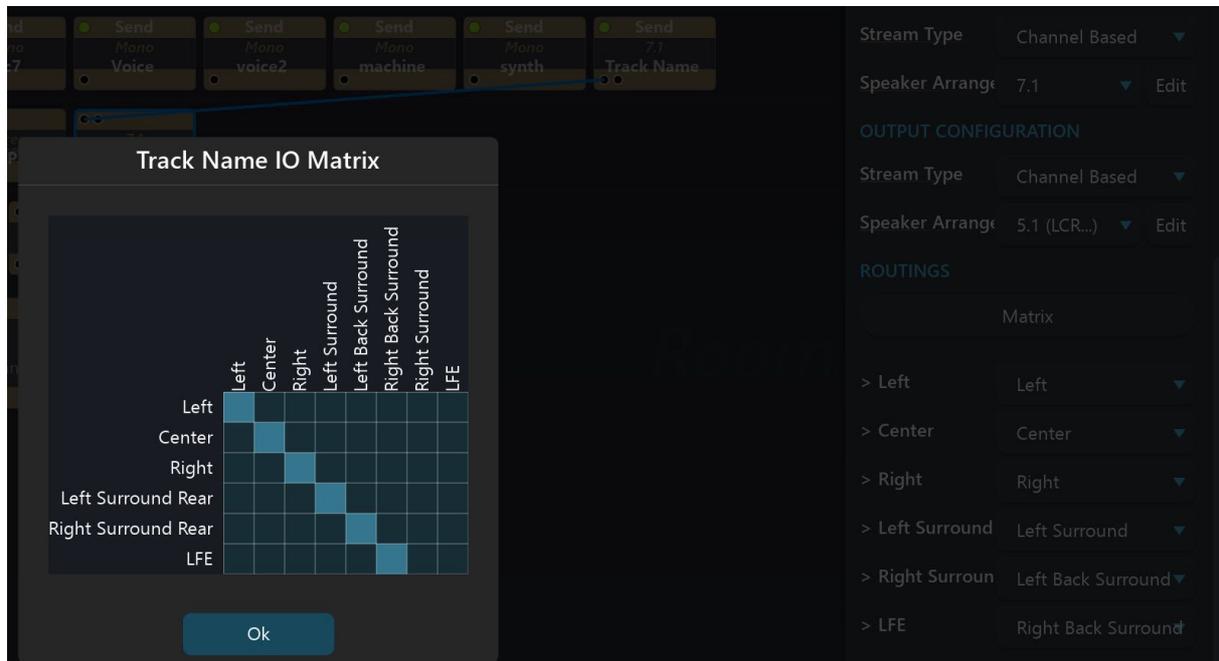
# 6.5 Input Transcoder

Spat Revolution can handle many different formats of multichannel audio through out the signal flow, as we have been pointing out. As we approach the actual virtualisation of inputs into object based audio sources in the *Spat Virtual Rooms* it may be necessary to change from the original input format to another. You use the Input Transcoder module and its parameters to do this.



Transcoder modules may modify the channel count of the stream passing through it, depending on the format transfer being requested. For example, transcoding from Ambisonic B-Format into a Channel Based 3D Cube involves a four channel Ambisonic stream getting transcoded into an eight channel stream grouped and treated as a specific speaker configuration.

# 6.51 Transcoding Matrix

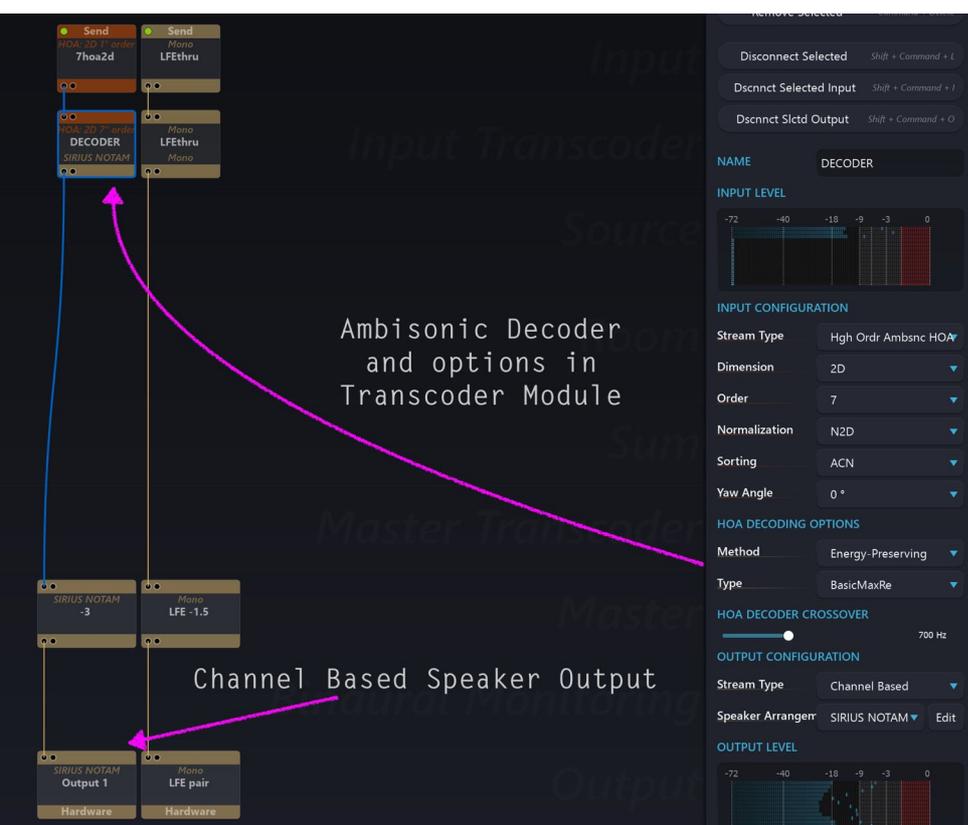


In the case where an incoming Channel Based stream needs transcoding into an outgoing Channel Based stream which has less channels, the IO Matrix is used to re-map the output format by dropping some of the input channels. Similarly, if the transcode results in a format with more channels than the input format, the IO Matrix is used to remap inputs to outputs. This is not strictly *Transcoding* or *decoding* but is a useful tool to have in certain format changing scenarios. To properly up mix or down mix, it is advisable to use a room to take the virtual source of one format, and output with the desired end format.

## 6.52

# When to Transcode Inputs?

The main reason you will need to transcode inputs is when you are mixing and spatialising inputs in a Spat *Virtual Room*. This is because the *Virtual Room* module requires incoming sources to be in a Channel Based format. Internally, the Room may well be panning in Channel based, Ambisonic or binaural format, but it always needs Channel Based streams as inputs. More about this in the *Virtual Room* section. Format transcoding may not always need re-spatialising in a Room. There are some contexts where you will not use a Virtual Room in the signal flow,



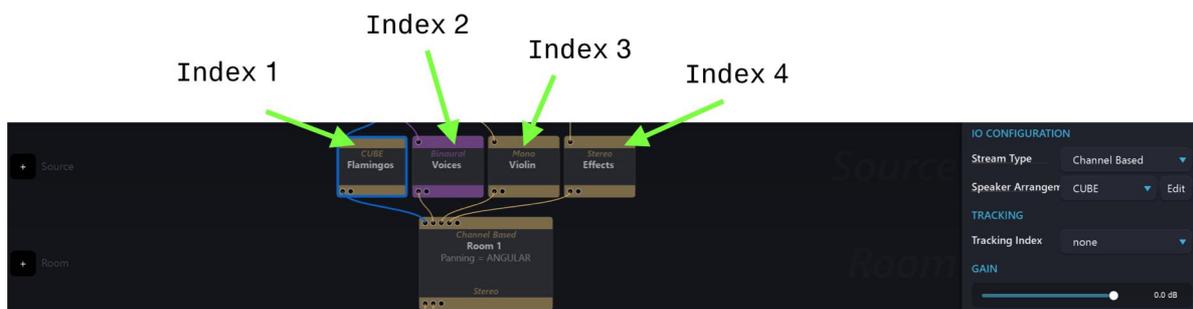
Here is an example of **de-coding** an Ambisonic signal using an Input Transcoder module. *This could also be done in the Master Transcoder section of the graph.*

As mentioned in 'Introduction to Ambisonic' (section 4.7) a decoding stage is absolutely necessary to render Ambisonic encoded audio to speakers. It can be done like this without the need of a room.

## 6.6 Source

The next row transforms the inputs into virtual objects according to their configuration. This is what needs to happen so that a virtual Source appears in a virtual room. If you try to connect an input directly into a Room, Spat will always put a Source module in-between.

The Source modules are where you set a descriptive Name of each virtual object. It's a good idea to do that, as things can get busy inside the Rooms. You can also adjust the overall gain of a Source here no matter how many channels it might have. The Source modules are also important for parameter automation using the *Spat Send* plug-in, and also in the case of external OSC control.



External software needs to know how to identify virtual objects, and that will be done using an Index number rather than a name. This Index number refers to one of these Source modules, numbered from left to right which in turn, becomes a Virtual sound emitting object in the *Virtual Room*.

When working in a Channel Based Room outputting to an *n.1 surround* speaker arrangement, each source will additionally acquire an individual *LFE Send*. This dial will be available in the Source Parameters inside the Virtual Room editor, and also mirrored here as a fader directly in the Source modules. *LFE Send* controls how much of this source's signal is mixed into the LFE channel at the output.

**★ When using a BlackTrax positional tracking system a Tracking index can be assigned to a virtual source directly from the Source module**

## 6.61

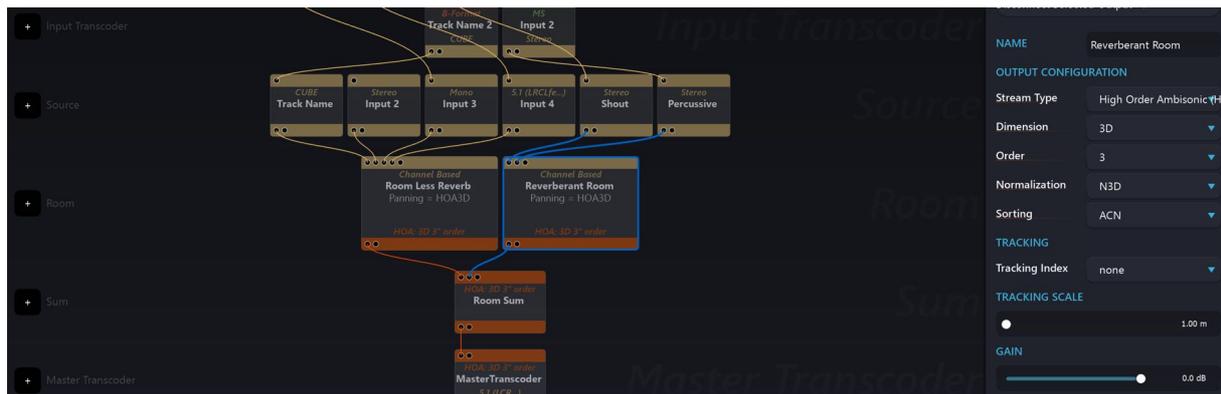
# Tracking

Spat Revolution is able to receive data from RTTrPM open protocol tracking systems. Currently, BlackTrax™ is supported natively. BlackTrax™ is a vision-based system that connects to different third party applications, such as robotic lights, media servers and Spat Revolution.

When you have correctly setup the BlackTrax protocol (see Appendix A) then you can directly assign Tracking Index to virtual sources, and also to listener position (see section 5.4) for advanced virtual reality interactive audio projects.

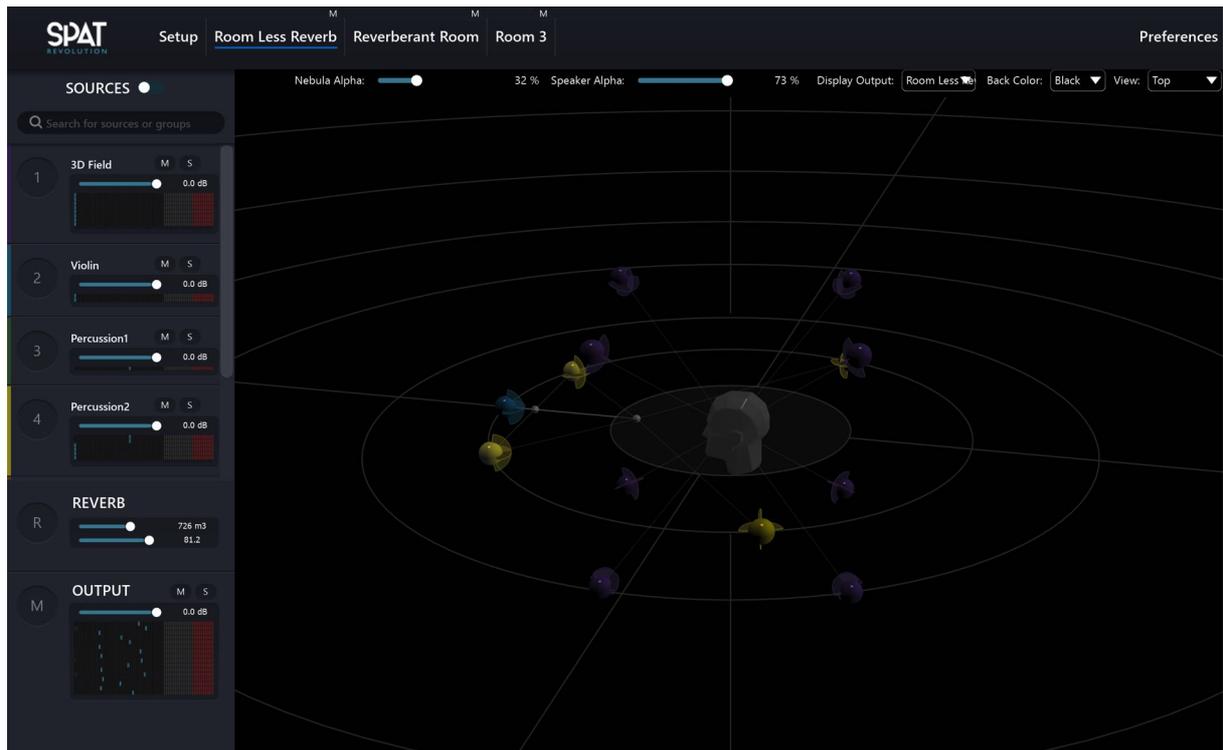
# 6.7 Room

In Spat Revolution, spatialisation of virtual sources takes place inside *Virtual Rooms*. To enter a Room and open its graphic editor environment, you double click on a Virtual Room module in the Setup graph, or select a room tab from the Navigation bar.



The first thing to notice, is that you can add any number of Rooms. In the screenshot above, two HOA 3D rooms are being used, each with differently designed acoustics. There is plenty of flexibility built into Spat Revolution, in order to encompass different workflow ideas or experimental approaches. For example, the same virtual sources may be assigned into multiple rooms, with multiple end destinations. Or as in the screen shot above, virtual sources might exist in different spaces, that get summed together.

When you enter a Virtual Room you will see the 3D positionable editor. All connected virtual sources and their emitters will appear on the 3D scene render. On the left side panel of the room editor, you get a list representation of each source with its Index identification number. You can click on the Index number of each source, and the *Source Parameter* editor for that virtual source will appear (see section 9).



★ If a source consists of a configuration of linked emitters there will be only one index and one set of parameters for the whole cluster

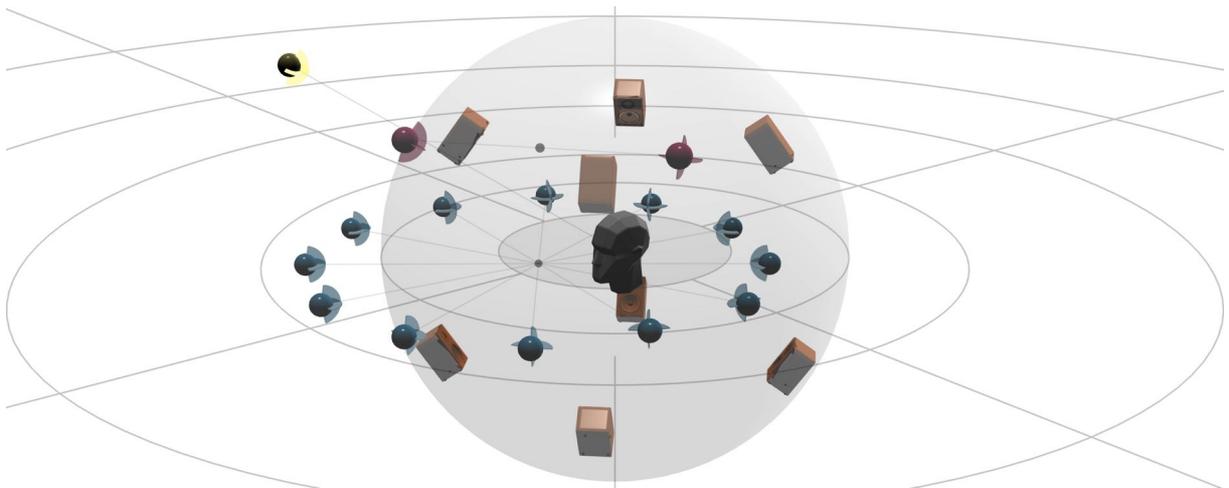
Two special index items labelled as **(R) REVERB** and **(M) OUTPUT** appear fixed at the bottom of the left panel. By clicking on these you then enter two more parameter editors, one relating to the *Artificial Reverberation* (see section 8) and one relating to the Room output configuration and listener position editor (see section 5.4).

All sources and the entire room output can be put into solo or mute mode directly from this index list. When you have more than one Virtual Room in your project, then the SOURCES switch at the top left of a Room editor can be handy, as it will show all Sources from all Virtual Rooms in the same editor - allowing you to edit, mix, solo and mute them from one Room view.

## 6.7.1

# Room Graphic Engine

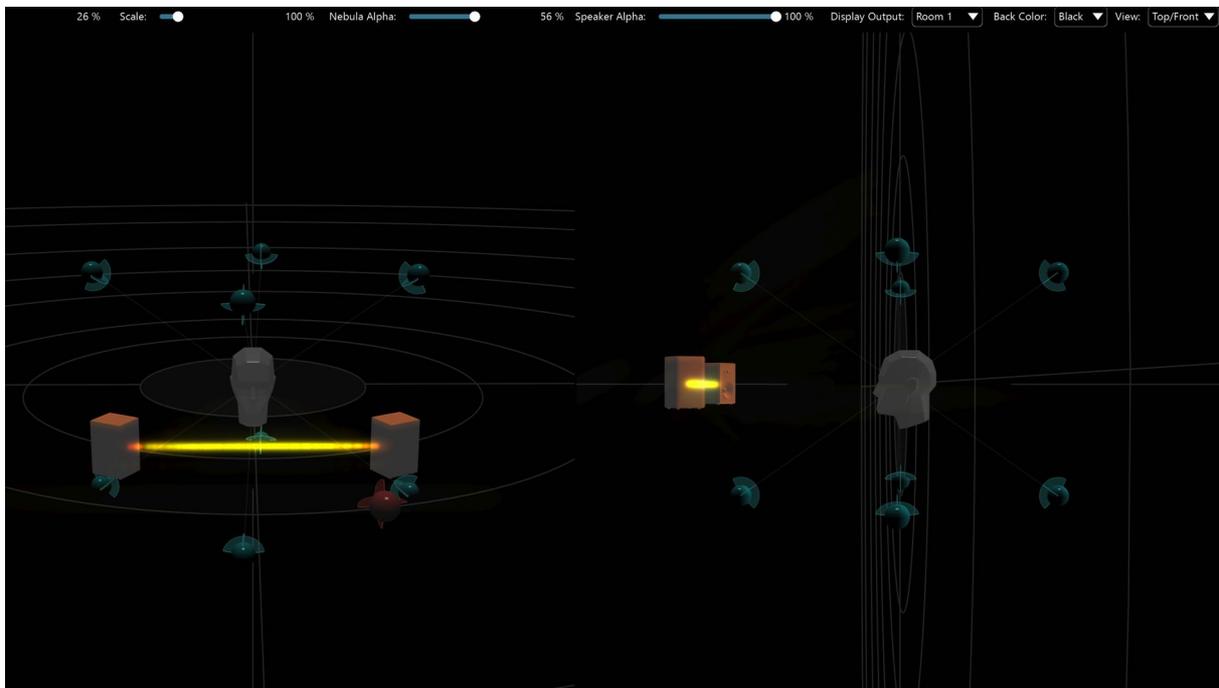
Along with the audio modelling engine, one of Spat's key features is its ability to model a high definition graphical representation of the virtual space inside each Room. You can interact and move sources and 'camera view' directly with your mouse in an intuitive way. Move a source by grabbing its 'emitter' object or in the case of a grouped source grab any one of the emitters that belong to the group. Alternatively, sources can be positioned by manipulating their co-ordinate related source parameter controls (see section 9).



## 6.7.2

# Room Camera View

The camera angle of the 3D scene can be moved continuously using a **control-drag** mouse or trackpad gesture applied directly onto the Room view. This will reposition the camera. A *forwards and backwards scroll* on the mouse or trackpad will zoom the camera in and out of the scene. Furthermore a change (or resetting) of view orientation can be selected from the *View* pull down menu, to show 'top-down' view of the scene.



A split screen Top-Front view is also available

## 6.7.3

# Room Output Configuration

The *Virtual Room* module requires all sources to be in some type of Channel Based format at its inputs. Internally however, the Room may calculate spatial positioning and panning using different methods according to the setting of the *Output Configuration* pull down menu.

A Room will compute internal virtual panning, reverberation and output in five possible formats:

- **Channel Based** (see 6.7.4)
  - Virtualised Sources
  - Virtualised Speakers
  - Virtualised Panning Behaviour
- **Binaural** (see 6.7.7)
  - Virtualised Sources
  - Virtualised HRTF
- **High Order Ambisonic** (see 6.7.8)
  - Virtualised Sources
  - Virtualised Ambisonic Encoding
- **B-Format** (see 6.7.9)
  - Virtualised Sources
  - Virtualised Ambisonic Encoding
- **Mid Side** (see 6.7.10)
  - Virtualised MS microphone arrangement

★ **Different Virtual Room types can be used in parallel**

## 6.7.4

# Channel Based Room

If the output configuration is set to be **Channel Based**, then the user must decide on the appropriate *Panning Type* to work with (see section 5.6). The loudspeaker array selected in the *Speaker Arrangement* pulldown menu will be 'virtualised' in the Room and all the speakers will appear graphically.

★ ***Speaker Arrangement* also defines the number of output channels**

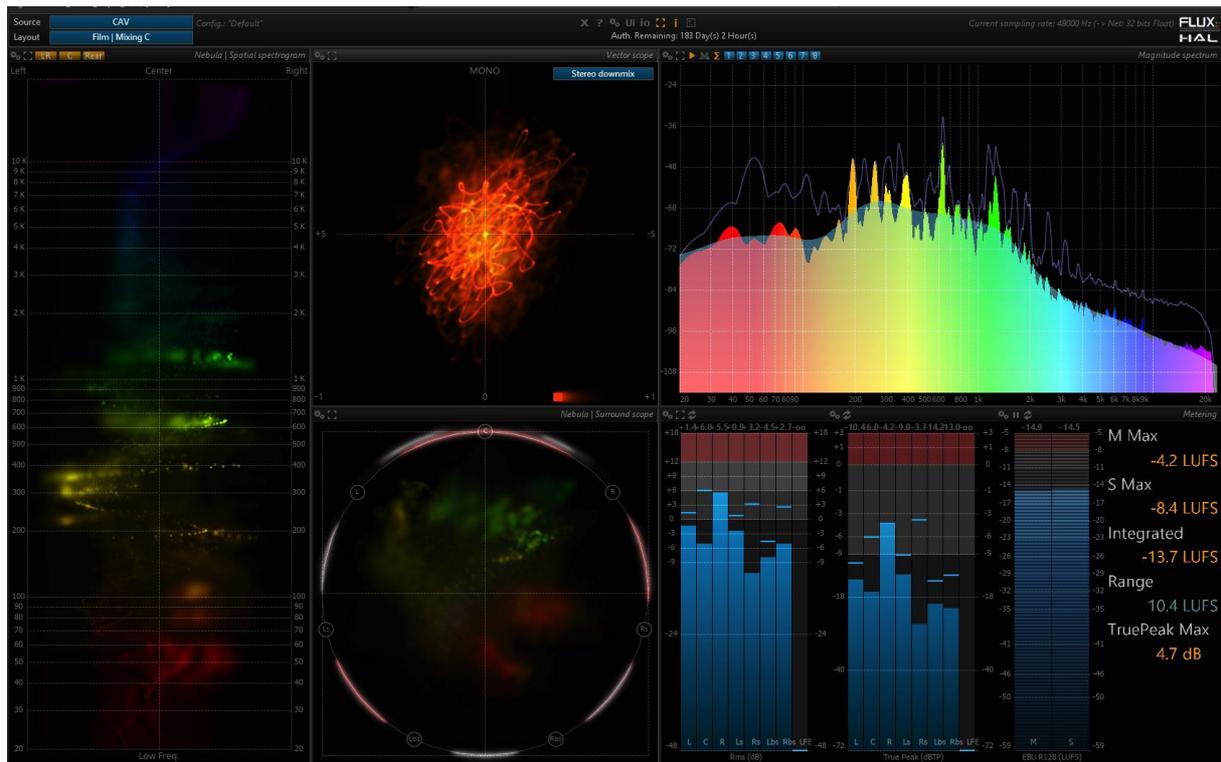
The spatial composition of Virtual Sources is governed in the Virtual Room by the characteristics of the currently selected **Panning Type** and speaker positions. In other words, the *panning type* is simulated inside the Room on the **Speaker Arrangement** model. **Artificial Reverberation** (see section 8) is also modelled in relation to the speaker arrangement in a Channel Based Room.

Additionally, the powerful *Nebula Spatial Spectrogram* is highly dependent on the virtual speakers model, panning type and simulated acoustics of a Channel Based Room. Let's take a quick a look at *Nebula* before continuing with the Room types.

## 6.7.5

# Nebula Spatial Spectrogram

*Nebula* is a technology adapted from our flagship **Flux PURE Analyzer System** a suite of highly regarded professional mastering and mixing visualisation tools.



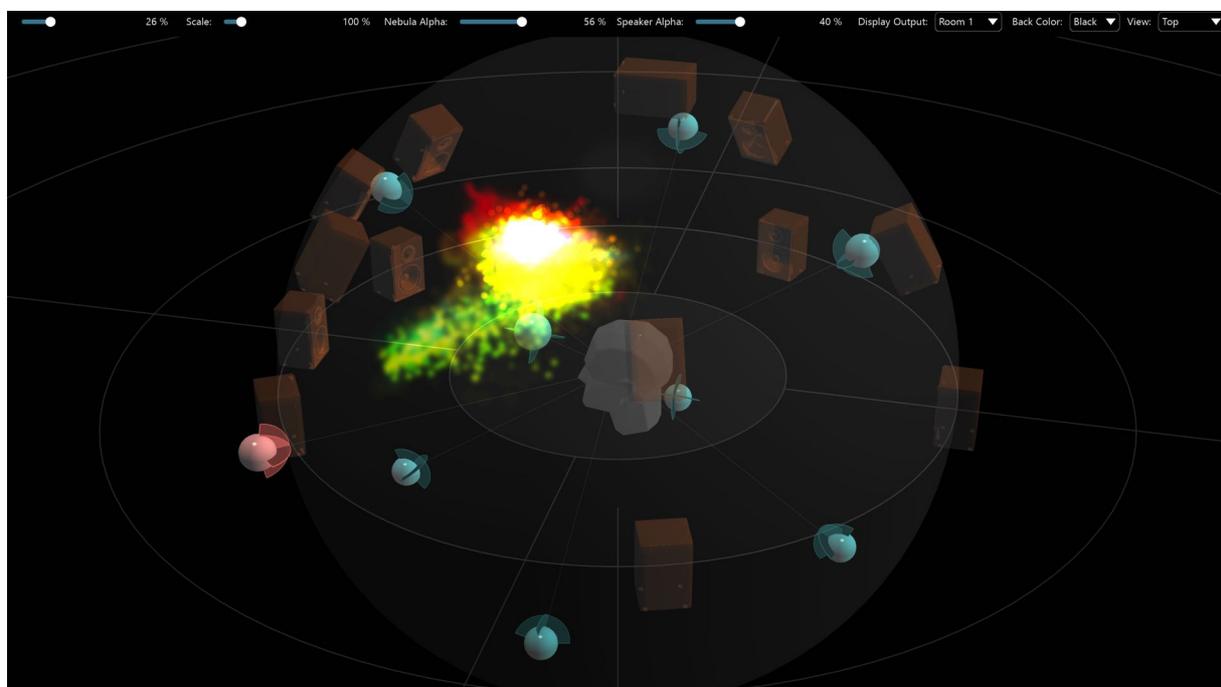
Screenshot from a Flux Pure Analyzer session

**Nebula** in Spat Revolution provides a unique representation of the sound field in terms of spectral content and localisation rendered directly inside the 3D speaker simulation and virtual room display. It combines the functionality of a spectrum analyser and a vector scope in a novel real-time display. It is a useful tool to get a realtime overview of your spatial mix in terms of spectral-spatial diffusion, and can give quite accurate representations of 'where' and 'how' sounds will manifest over a real world sound system. A lot of work has gone into optimising the real-time

rendering of the display, not solely for æsthetic reasons, but because we wanted the display to react instantly to all the details in the incoming multichannel audio. The idea is literally for you to be able to see what the listener will hear and feel.

## How does it work?

The overall principles behind *Nebula* are quite straightforward. At any given time, and for every frequency, the engine computes the position of a frequency in space (2D in stereo and 3D for multi channel surround). This position is taken as the center of gravity of the various channels, weighted by the relative amplitude of the signal in their corresponding channel. A colour-intensity mapped projection is computed for the multi-speaker plane, giving a spectrum-space frame constrained to the surround sound field radius or sphere. Past analysis frames are progressively “forgotten”, using blur and dimming, in order to make place for new information, which gives the graphic display increased legibility and its characteristic 'nebulous' quality.





## 6.7.6

# Virtual Room Concepts



## Simulated Speaker Diffusion

When a Virtual Room is Channel Based, the speaker configuration layout is modelled as fixed speakers which appear in the Room graphically according to their programmed positions (see section 4.5 *Custom Speaker Configuration*) - you will not be able to move those speakers around with the mouse. These fixed speaker positions are virtualising a sound system configuration which is often the same one that will be used for diffusion. These virtual speakers interact with the virtual acoustics of each Room as designed by the *Artificial Reverberation* editor (see section 8).

**★ Each room can have a DIFFERENT Virtual Acoustic design and different sources.**

The idea is that by simulating speaker positions and acoustic interactions, the process of mixing for a particular speaker layout becomes more predictable. For example, by monitoring binaurally a scene from a channel based room, it is possible to get an impression as to how the mix might sound diffused by a particular speaker arrangement (including the space between speakers and gain characteristics belonging to selected panning types). As we have mentioned earlier in section 4.6 on Ambisonics, there are also options for mixing and monitoring a particular Channel Based Room simulation transcoded to a different setup than that being simulated, which may also be useful in certain contexts.

This is really all about mixing. Mixing in stereo is already a significant challenge, some have devoted their entire careers to it. Mixing for multiple speakers, is even more complex.

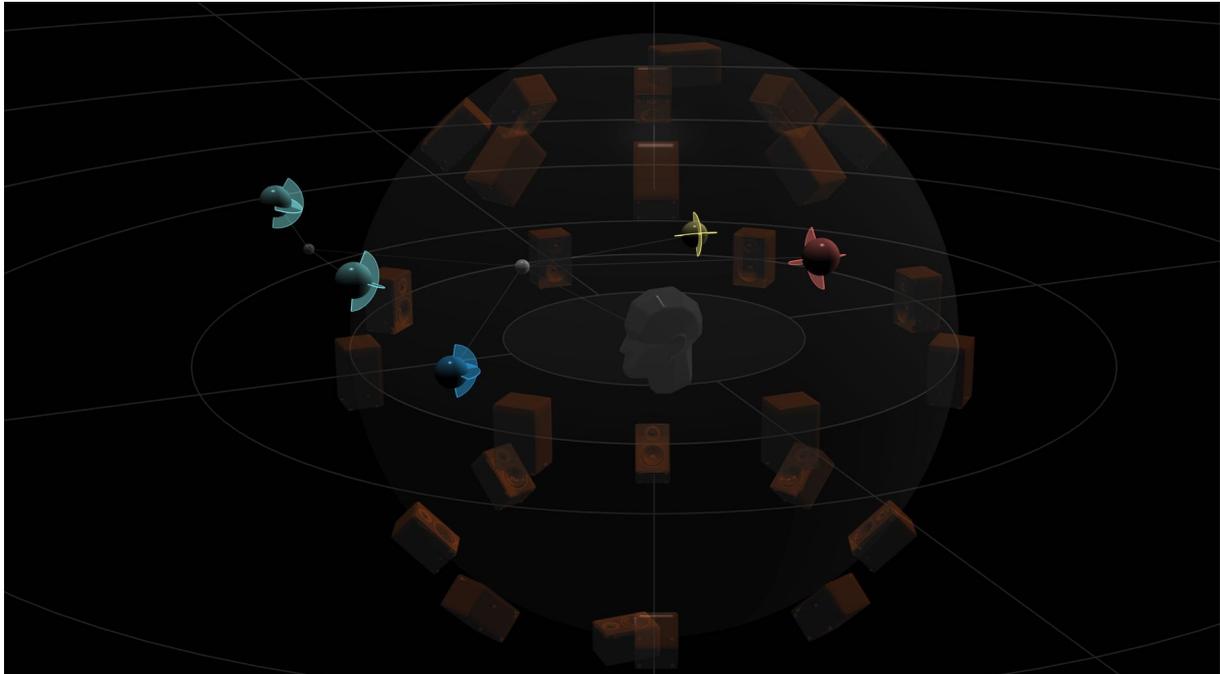
As mentioned in section 4.6 and other places in this guide, it is advisable to think in parallel as far as room mixing goes. You could think of SPAT Rooms as different kinds of spatial busses with integrated acoustic emulation units. Use them in parallel, send some or all of the sources to multiple Rooms at the same time and sum the outputs together into one output format. One room might have a huge reverb, and one might be a much smaller space. The fact of the matter is that different

Channel Based Rooms will have different sounds, but they can all be summed into the same channel based format.

It is only in this kind of Channel Based room workflow that the *Nebula Spatial Spectrogram* will appear, as it works by computing a diffusion map from the combined outputs of statically positioned sound sources - i.e. *speakers* .

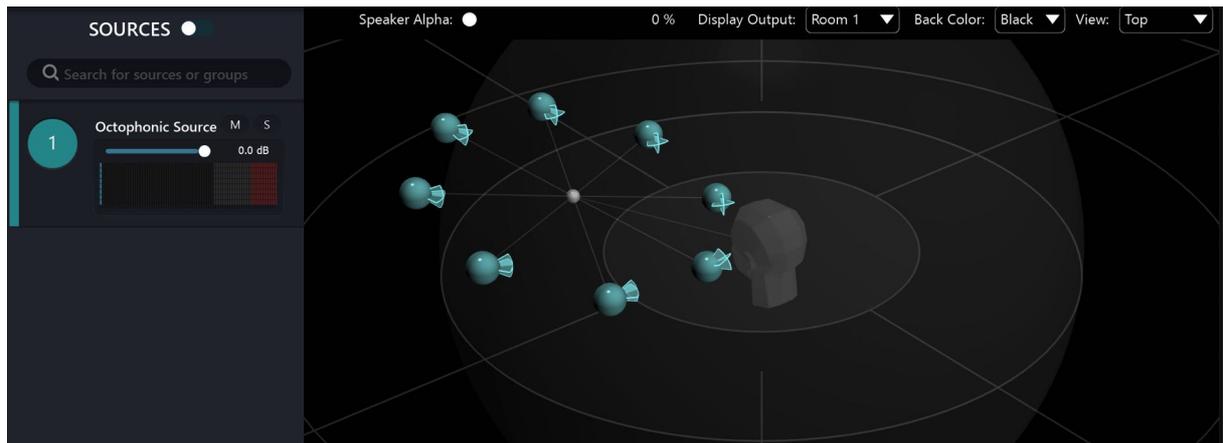
In practice, the best mixes are attained by mixing on the same sound system in the studio or venue as you have represented in the Channel Based Rooms. But there is also the possibility of monitoring Channel Based mixing on headphones using the binaural monitor or a dedicated, parallel, binaural room as mentioned in section 5.1. This is workable enough to allow you to keep on arranging a mix and spatial composition 'off-location' without too many surprises when it comes to the final result.

★ **Channel Based simulations render localisable virtual sources by controlling the virtual speaker array in a similar way that panning-laws do on real world physical systems. They do not model the positions of virtual sound sources, but instead control the gains of the virtual speakers in the model to render perceived positions.**



## **Multichannel Virtual Sources**

All inputs to a Room must be in some kind of Channel Based format, even when the Room is an Ambisonic or Binaural Room. Even a Mono source can be considered as a channel based signal of one channel (see section 6.41). For any source that has more than one channel, it will be structured as a Channel Based configuration - and that configuration will be virtualised inside the Room as one single source in an inter-connected cluster formation - this is designed to maintain inter-channel spatial image relationships as much possible, keeping the multichannel source in a discrete self-contained system. The sound, size and positioning of that configuration can now be manipulated as a unified group.



# Speakerless Virtual Rooms - Binaural and Ambisonic

The better alternative to Channel Based simulations is to use a *Binaural* or *High Order Ambisonic* room to mix. In these Room simulations, speakers and panning are not simulated. Instead all the inputs into the room become *Virtual Sources* - they become sound emitters in the virtual space. To get a grasp on how mind-bending this is, we need to go through it a little.

In a fully virtualised Ambisonic room, the *Nebula Spatial Spectrogram* will no longer be available, as it is too complicated to compute the spatial-spectra of moving sources. No visualisations of speakers will appear either, as they are not being modelled.

So far, we have established how the audio technology at the heart of SPAT is capable of simulating precisely the emissions of virtual speakers at fixed positions, and the way they interact with virtual acoustic space.

But the real big achievement of SPAT and High Order Ambisonic is the modelling of virtual sound sources that are changing orientation, size and position. SPAT can model these *and* simulate the changing response of the acoustic space to these complex, dynamic and *imaginary* sound diffusors.

This is where we touch on what 'virtualisation' of sound sources really means.

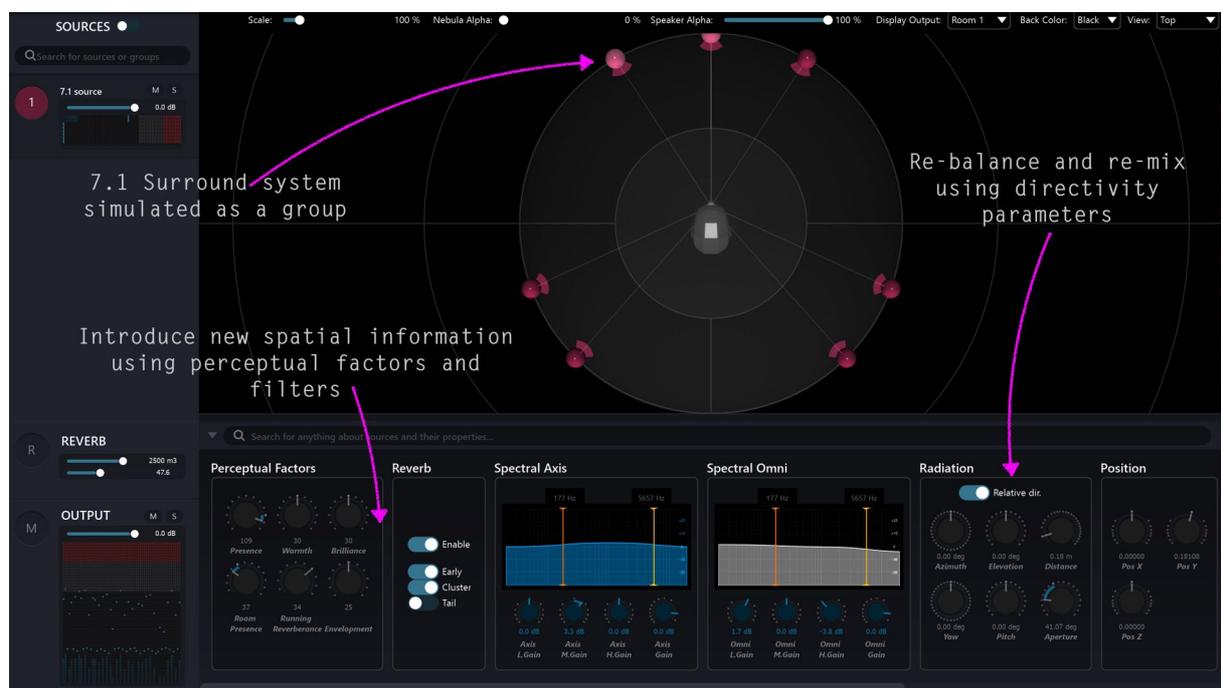
The technology of Virtual Sound Systems and Virtual Spaces invites us to stretch our imaginations beyond what is possible on a physical installation and into a place where sound emitters can continuously alter their structural dimensions, orientation and acoustic characteristics - the idea of speakers as fixed objects no longer applies in the virtual acoustic space. Instead, sound emitters grow wings and become complex musical elements to be organised and composed in time and space, contributing even more to the cultural experience of music.

# Upmixing and Downmixing

One workflow that is often required, is that of re-mixing a pre-rendered channel based mix of a particular format, to get that mix expressed into a different channel based format. For example, the need to down mix a 5.1 surround into stereo is quite common, but also the other way - perhaps a 5.1 surround mix needs to be up-mixed to a 7.1.

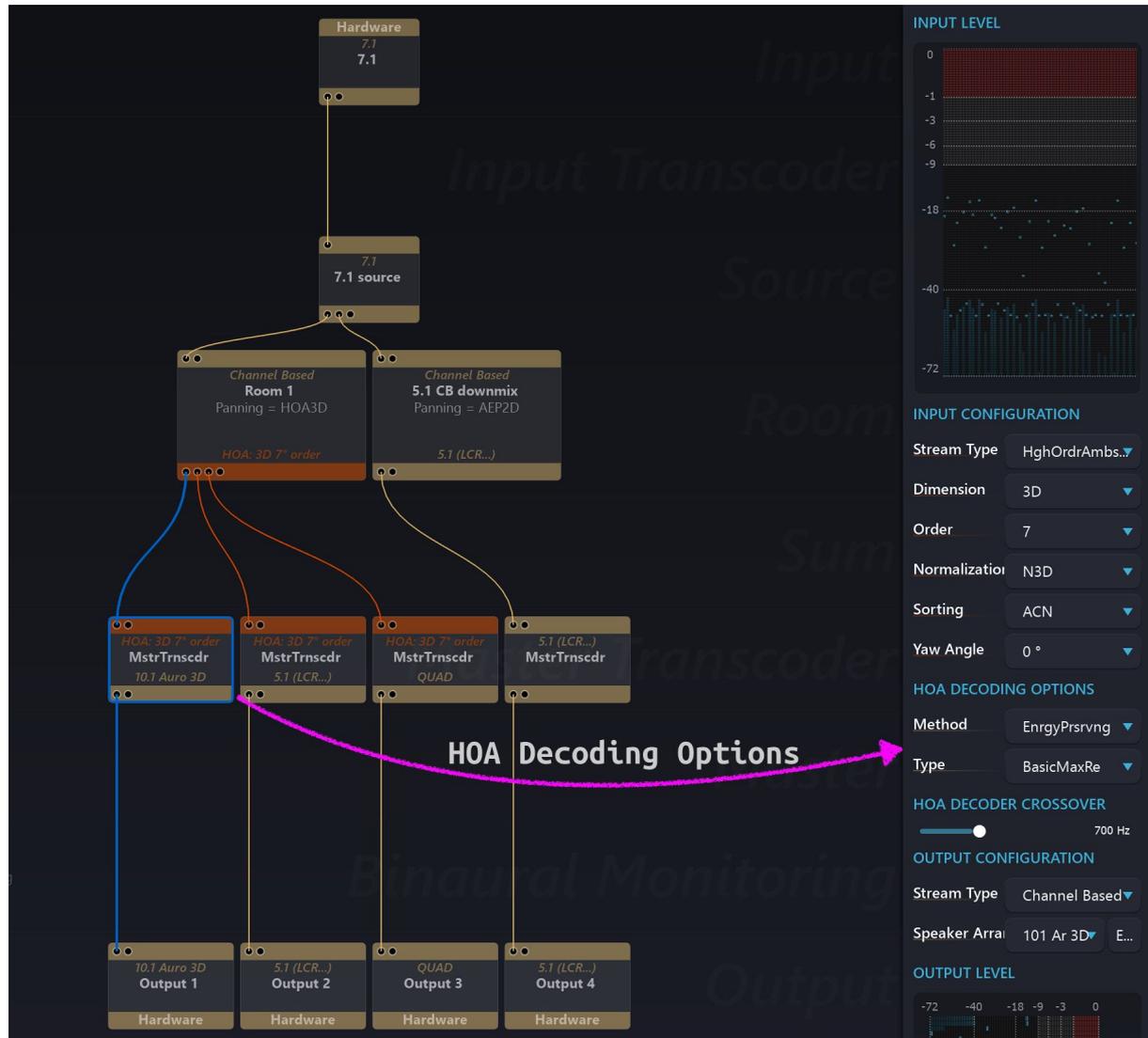
One way to do this, is to use an Ambisonic Virtual Room as a way to simulate the source configuration as if it were being diffused in a space, in its correct speaker format and with simulated, full sphere acoustics and then transcode the output to the desired format.

Here is an example of that in the Room view. Notice how it is possible to re-balance and re-mix the original using the perceptual factors or positional parameters. A room simulation reverb is also playing a significant role in the cross-format mixing process, as the reverb simulation adds new spatial information to the mix, which might help get a more immersive result in the output format.



This is an example of a multiple format changing signal graph, which changes format in two ways, to provide an alternative result. Firstly, by transcoding the output of an HOA Room into various other Channel Based formats, but also a room that simulates how the target speaker configuration might render the source speaker

configuration. Both methods will give different results and the choice is up to the designer.

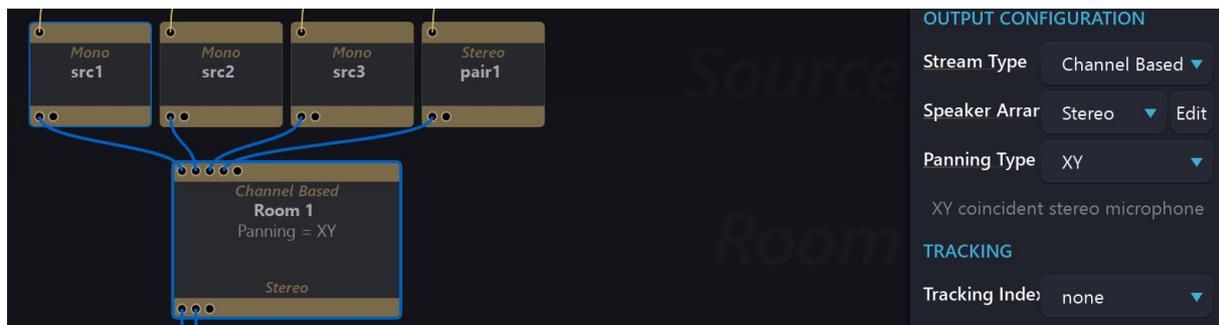


Options are available when *transcoding* (aka *decoding*) from a High Order Ambisonic room into a Channel Based stream. These decoding options will affect the sound in quite different ways - luckily it is possible to listen to the results of different Ambisonic decoding options in realtime. More about this in section 6.7.8 which details more on the HOA Room concept.

## Virtual Stereo Diffusion and Microphones

When a Virtual Room is set to be Channel Based / Stereo it will model a stereo speaker system in the virtual room acoustics *and* virtualise stereo microphones in real-world stereophonic recording configurations. This is a hybrid approach, where

you 'record' the sources positioned in the Room through virtual matched pair microphones at the listener position, as if you were miking up a concert for stereo broadcast or playback. The microphone modes were described in section 5.68 and become available under the Panning Type menu of the Room when it is set to Channel Based Stereo.



You will notice how the Spat Reverberation handles the stereo image naturally, because it is modelling the entire mix scene in the acoustic context of a Virtual Room, rather than the conventional 'voltage controlled' left-right pan pot that all audio mixers offer for stereo mixing.

This method of working in a stereo Room invites a different approach to stereo panorama source mixing in the studio or on twin stack PA systems. It can also be a way to down mix from a surround format to stereo by simulating the speaker configuration and 'recording' it in stereo.

One final thing to note when working in an Ambisonic room, rather than a channel based room, is that the feeling of **Artificial Reverberation** (see section 8) that gets encoded into the High Order Ambisonic will have a much reduced strength than what is heard during a channel based room mix. This needs some trial and error to get a feel for

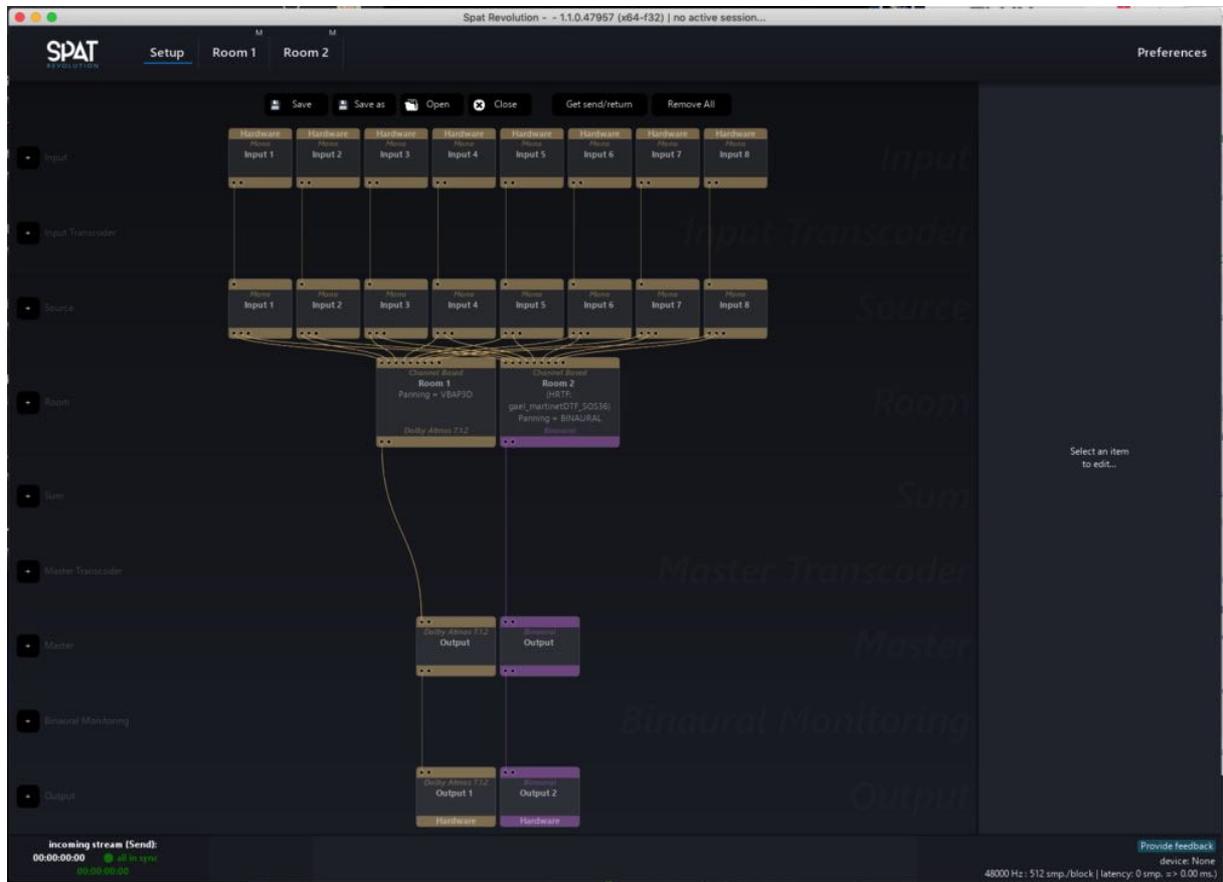
## 6.7.7

# Binaural Room

The Binaural Monitoring module is great for easy channel based system monitoring. But it is not the best binaural experience we can provide with Spat Revolution. The best way to work with binaural encoding in Spat Revolution is to use a *Virtual Room* which is binaurally encoding each virtual source at its exact position in a virtual space, with *Artificial Reverberation* but without modelling any virtual speakers. In fact, in a Binaural Room the only diffusion factor in the modelling is the *HRTF* (see section 5.1) .

Each source's direct sound plus the reverberation it creates are modelled and synthesised binaurally for *each individual source*. This advanced processing can result in an excellent binaural experience; more precise and natural sounding than using the Binaural Monitoring module to listen to a virtualised Channel Based rendering. It is the preferred method to use when rendering Binaural content to disk. To do that, you simply need to connect the binaural stream from the room directly to a (stereo) SPAT Return path back to your DAW. You could record and listen to a Binaural Room by using two output modules, one to the recording route, and one to a headphone output.

★ For the best Binaural Monitoring try setting up two rooms, one all Channel Based going to speakers and one Binaural for your 3D headphone mix.



## 6.7.8

# High Order Ambisonic Room

One of the most straightforward methods to start working with HOA spatialisation in a signal flow is to connect input sources directly into an HOA Room. To convert a Room to be HOA, you select High Order Ambisonic as the *Output Configuration Stream Type*.

All inputs to a Room must be in some kind of Channel Based format, even when it is an Ambisonic or Binaural Room. That is the workflow at the time of writing (Spat Revolution v1.1). This makes most intuitive sense, when using Ambisonic format inputs as pre-encoded "3D sound field" type inputs into a room. You cannot just add them into an HOA room, even though they may be HOA format. Why is that?

This is because Ambisonic audio always needs to be *decoded* into a channel based format to hear the spatialised audio on speakers ( as we have mentioned a few times already) . And so in a virtual room, an Ambisonic source needs to be decoded to a *virtual* speaker configuration. The choice of what virtual speaker configuration you decode the Ambisonic audio to, will influence the way the source's direct signals will sound and its sound will be altered by the way it interacts with the Virtual Room Acoustics. If it is a 1st or 2nd Order B-Format source, then a Cube configuration for 3D or an Octaphonic for 2D is usually a good choice. The important thing to keep in mind, is that a higher order of Ambisonic encoding, will sound more accurate on a higher density channel-based configuration - and if the source Ambisonic encoding is 3D, then you really should choose a 3D type of channel configuration.

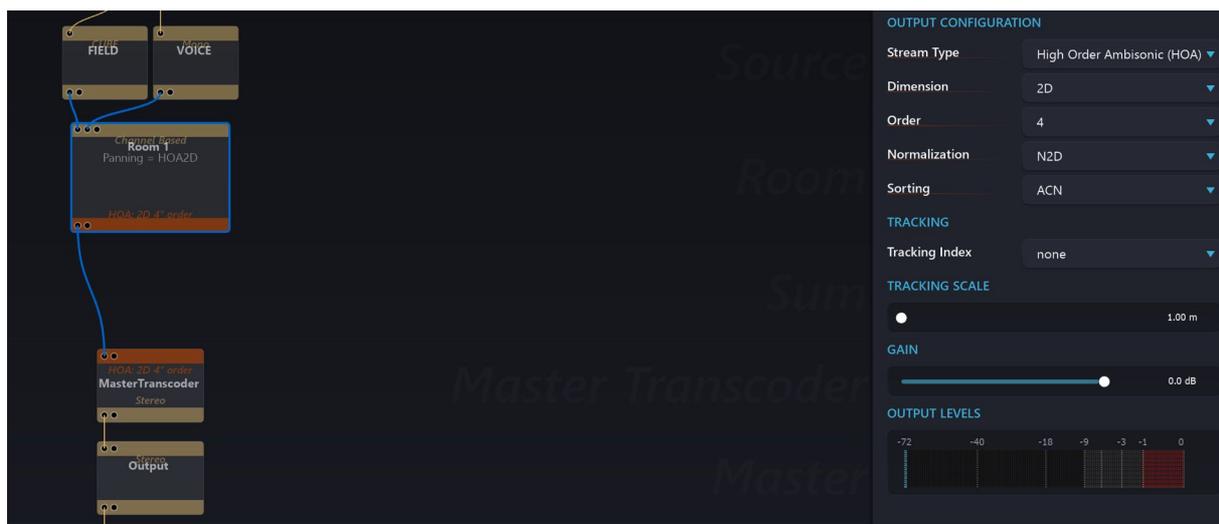
△ *Ambisonic encoded audio is not intended to be listened to without decoding*

The other important thing to keep in mind, is that by decoding Ambisonics to a vir-

tual Channel Based configuration, allows for some very special transformations and re-designing possibilities of the original Ambisonic source. When a B-Format source becomes an arbitrary configuration of virtual emitters in the virtual Room, the ambisonic sound-field becomes a group that you can manipulate and transform using all the source parameters - especially, the barycentric source parameters which are really useful for rotating and transforming the original sound field in many ways (see section 9.9).

### ★ Ambisonic encoding is a convenient way to share and archive spatial sound mixes

The output from an Ambisonic room must always be *decoded* into a channel based stream in order to hear the resulting spatial image (see section 4.7 introduction to Ambisonic). But even though it is not speaker compatible, it is still an audio data stream - so therefore it is quite possible to record and HOA stream to disk without decoding. This is a powerful way to work, as the HOA format encodes full sphere spatial information which can then be decoded/transcoded at a later stage.



In an HOA Room, the Order and other Ambisonic encoding parameters can be changed at the output configuration of the room very easily, by selecting a different value pull down menus. When you have correctly set up a speaker compatible decoding method for the HOA stream coming out of a Virtual Room, you should clearly hear the difference in focus and the way that the Artificial Reverberation behaves as the Order goes up. Do not be complacent about the simplicity of the HOA interface. There is plenty going on under the hood but Spat makes it feel simple to explore how different Ambisonic Orders sound. Similarly, other encoding options are available to apply and listen to directly.

An HOA Room simulates many different Ambisonic decoding options in realtime. When you decode (or Transcode) an HOA Room output into a Channel Based or Binaural stream, you will be able to hear how different Ambisonic encoding and decoding options (such as Method and Type) will sound. The topic of Ambisonic decoding options such as Type, Normalisation, Sorting and Method is a particularly large one but in the end, it comes down to how the sound field will come across - and this involves listening. Similar to how Sommeliers get to know the nuances of wine by opening lots of bottles, the best way to get to know Ambisonic options and understand for yourself how they affect the sound, is by decoding and listening on different setups. And perhaps also digging into some of the literature you will find online.

One thing we can already guide you in, is how each Order affects spatial precision. This knowledge may be beneficial to understand when you are rendering and mixing Ambisonic sound fields. The precision with which a source can be localised in Ambisonics is directly linked to the Ambisonic Order. Sources encoded as first order (FOA or B-Format) can be quite diffuse in their localisability when decoded. Higher Orders, such as 3 or 7 can render precisely localisable point sources in 3D. The cost is in computation and increasing audio channel count ( see Section 6.45 ).

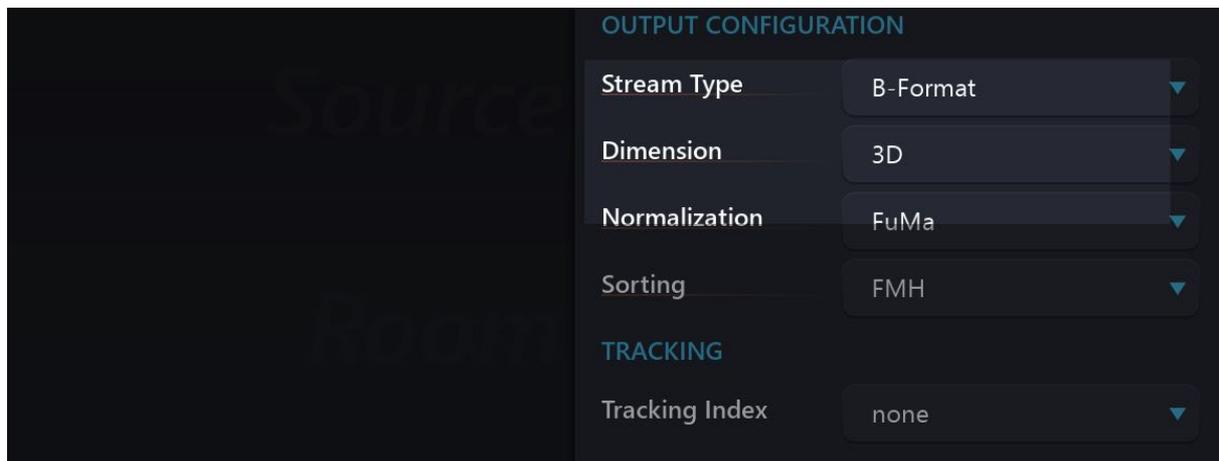
Some types of source material such as ambience or field recordings can benefit from the more immersive feeling of lower Orders, but point sources may need encoding in higher Order to get a more localisable result.

## 6.7.9

# B-Format Room

This is a First Order Ambisonic (FOA) room which is correctly suited for mixing B-Format Ambisonic signals. B-Format is 4 channel interleaved Ambisonic format, which is already widely used as a 3D audio format in the audio industry, as it can be decoded quickly and efficiently in realtime. If you are producing content that is intended for realtime decoding in B-Format, you can work in a B-Format Room.

The output configuration is preset for B-Format Room, although it is possible to change to 2D or 3D.



In the classic Furse-Malham (FuMa) sorting system for 1st order is: WXYZ. W is the mono sum and it is carried on channel 1. In the ACN channel order convention a 1st order signal is arranged as: WYZX.

## 6.7.10

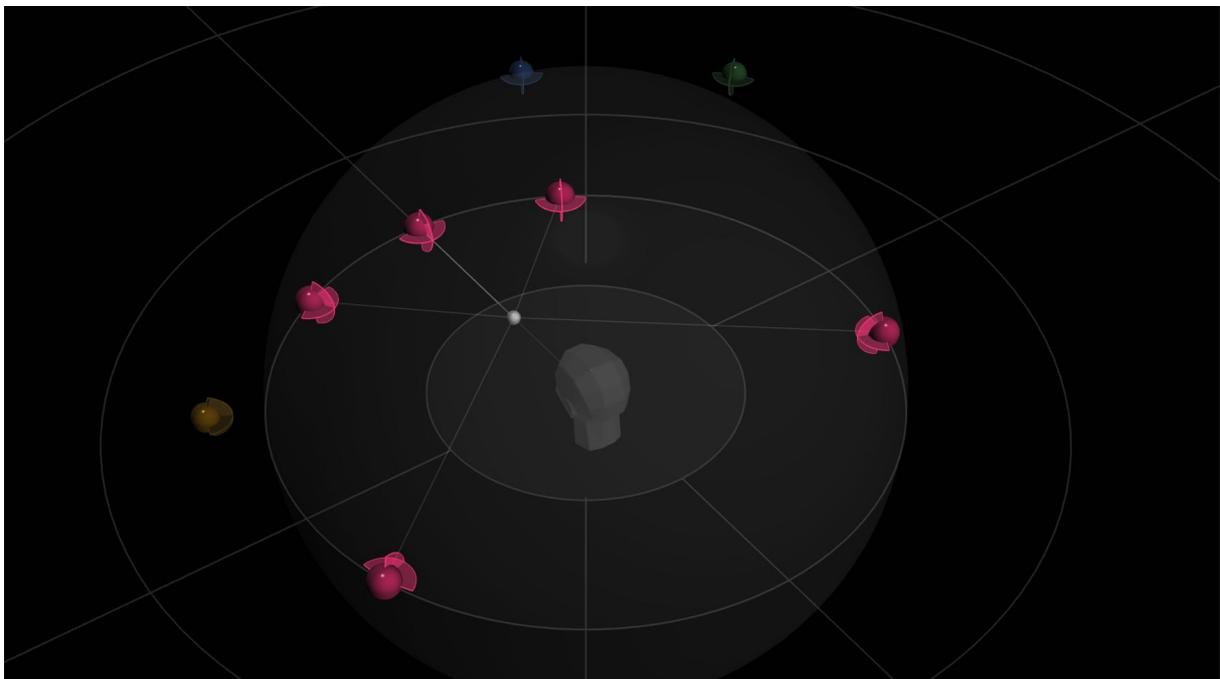
### **Mid-Side Room**

Similarly if you are working with two channel signals already encoded as MS format stereo, you can mix in an MS room. The outputs of MS rooms need to be decoded or transcoded into a suitable stereo format for reproduction on a stereo system.

## 6.8

# Barycentric Groups in Rooms

When a Virtual Source in a Room has more than one channel in its format, it will be represented as a single source with ONE unique index. It will be visualised as a group in its assigned channel based speaker arrangement.

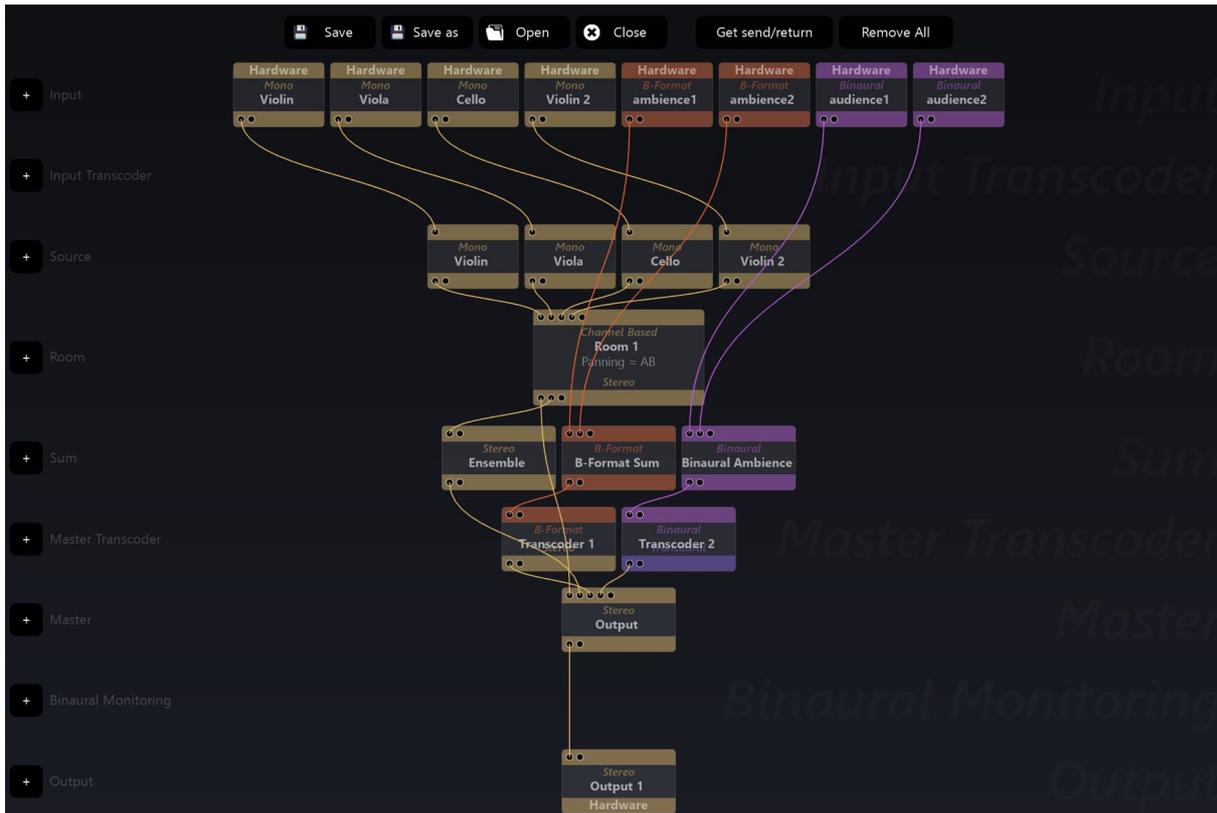


In the above screenshot, the red source consists of 5 channels arranged as a DTU 5.0 surround sound configuration. A multichannel cluster can be conveniently positioned and manipulated as a *single group* which maintains its correct internal spatial positioning relationships but moves in relation to the absolute listener position in the Room (the Dummy Head). The dot at the centre of each cluster, where each virtual 'channel emitter' is attached is called the 'BaryCentric' focus - In other words, a *relative* listener position that the virtual source configuration remains focussed on.

These complex spatial positioning algorithms are computed and controlled in real-time using Spat Revolution's advanced *Barycentric* and relative direction source parameters. A group that may contain many elements can be transformed, scaled, moved and manipulated in complex ways, through only one set of controls. See section 8 for a breakdown of the **Source Parameters**

# 6.9 Master Section

The **Sum** row of modules is used to mix the output of two or more Rooms of the same output configuration and in some contexts, to sum inputs directly without the use of Room.



The **Sum** module can handle different input configurations. It will Sum channels based on their channel names, so correct naming convention is important. Summing a 5.1 and 7.1 room together will output a 7.1 where their common L, C, R, Left and Right Surround rear channels will have content summed from both rooms but Left and Right Back surround will be only from the 7.1 room.

★ **Summing can also be done directly in Output Modules**

The **Master Transcoder** row of modules offers an opportunity to consolidate the various formats you might have been mixing into one (or more) formats for final output routing by the **Output** modules. The same options and routing are available as in the *Input Transcoder* modules.

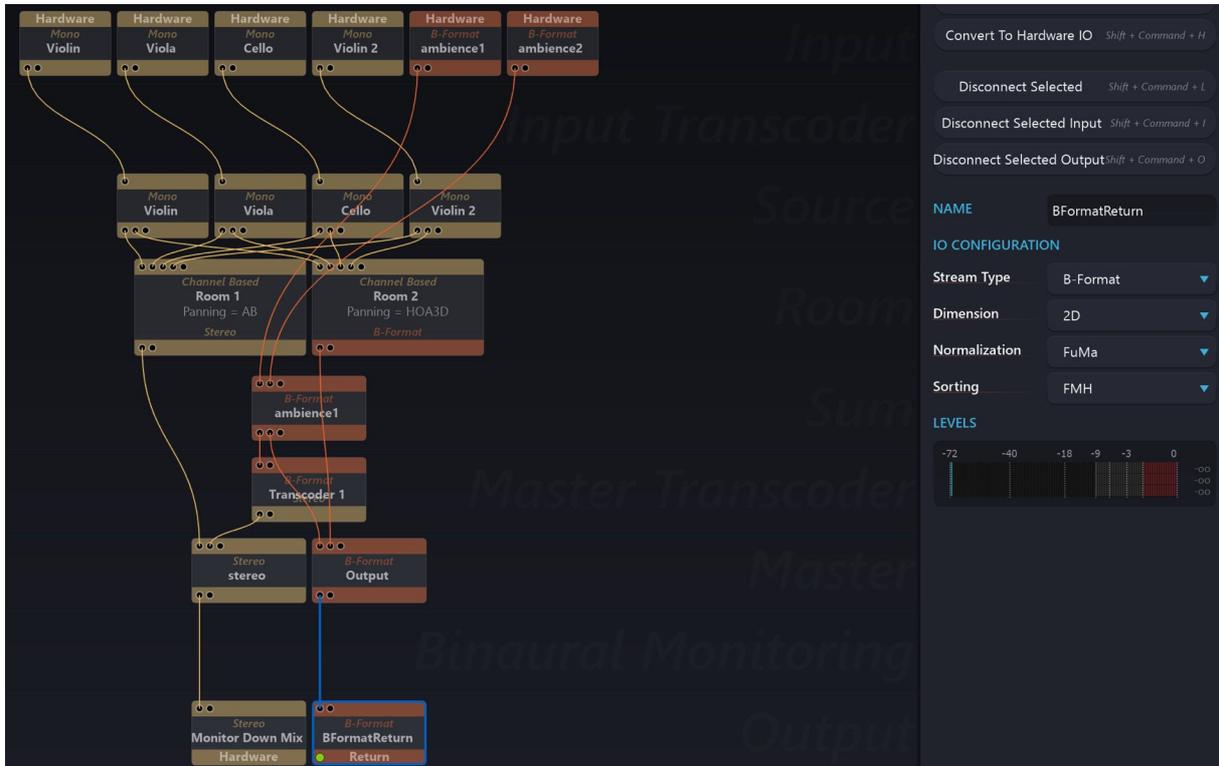
Finally, the two bottom rows in the *Environment Setup* graph are the output modules for the entire project. The **Binaural Monitoring** row provides a way to decode the whole scene to headphones only in binaural 3D (see section 5.1).

## 6.10 Output

The main thing to note at the output stage, is that you can have any number of output routes. Like the Inputs (see section 6.4) they may be either direct hardware routes, which returns audio streams (via Matrix routing) to the physical outputs of an audio interface connected to your Spat Revolution workstation. The Hardware Output workflow is the most direct way to render in realtime to an actual loud-speaker system (or headphones).

Outputs may also be linked to Spat Revolution RETURN plug-ins, which returns audio streams internally on the same workstation to a valid Spat RETURN plug hosted in your DAW. The software RETURN workflow offers an easy way to render the spatial scene to disk, as *Ambisonic encoded*, *Binaural encoded* or *Sound system encoded* multichannel audio.

★ Create multiple output routes to capture Ambisonic recordings at the same time as sound system specific rendering



# 7 SPAT Plugins

A computer running Spat Revolution becomes a spatial sound rendering engine, hosting the entire Spat software environment we are currently describing. Audio signals that flow in and out of Spat may be connected to a hardware multichannel audio device connected to the Spat host machine.

Interestingly, a multichannel audio stream flowing in and out of Spat may not be coming from hardware at all, but instead connected through *virtual IO* from another software application.

In the Spat environment this is made possible by a special set of plug-ins that enable audio to flow through 'virtual cables' between a multichannel DAW and Spat Revolution - *both running on the same machine*. This can become an ideal offline creation workflow for preparing content for an eventual complex hardware setup.

To understand the Local Audio Path flow better, let's breakdown three setup scenarios.

## 7.1

# Single Hardware Workflow

When you have a high channel count hardware IO connected to your Spat Revolution workstation, it is possible to receive signals from the hardware unit physical inputs into Spat at the top of the setup environment and route to the units physical outputs at the bottom of the setup graph - these are labelled as 'Hardware' IO in Spat.



Hardware inputs connections could include:

- Mic
- Line
- MADi
- DANTE
- AVB
- ADAT

*△ Make sure you have the Hardware Device selected in the Spat preferences!*

Hardware input formats could be mono, stereo or a format with any number of channels. Channels could be set up as single virtual sources, or as group multi-channel sources using only one input/source module.



## 7.2

# Distributed Hardware Workflow

Routing via hardware IO means that in principle the audio can arrive at Spat's hardware inputs via MADI or other high channel interconnect from another machine handling the audio playback (and/or recording). This is a powerful and reliable combination as it shares audio computational resources.

Furthermore, a single machine setup can be converted to a fully automated distributed using OSC to target the single or dual Spat engine for a real time controllable and interactive distributed hardware setup.

*△ Enable bi-directional OSC communication in Spat preferences to send automation over the network to another machine*

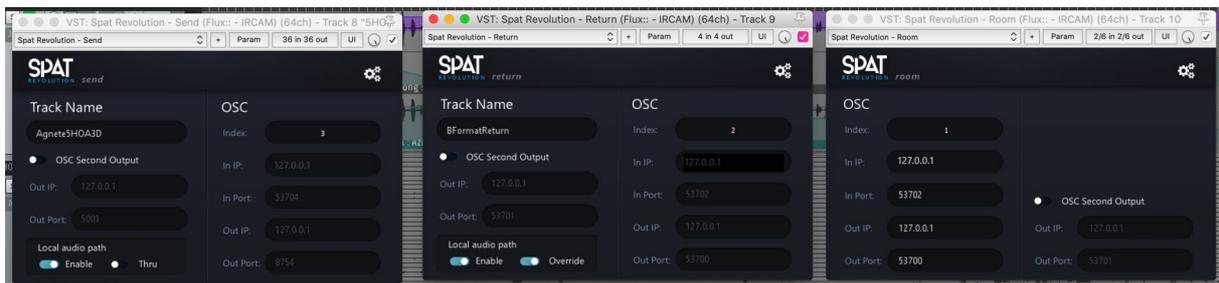
Contact Flux Support if you need to expert support on this kind of advanced distributed setup.

## 7.3

# Local Audio Path Workflow

You may only have the option to use one computer. In this case you need a software method to move multiple channels of audio from software running on the same machine as the Spat Revolution spatialisation environment. This is not normally a trivial task, and many previous solutions have been prone to drop outs and other problems.

In answer to this challenge, Flux have created three plugins for AU, VST and AAX hosts. The Spat Plug-ins offer a straightforward way to integrate the Spat Revolution spatialisation environment with other digital audio workstation environments - *running on the same machine.*



### Spat Send

- Send Multichannel audio from a DAW track to a Spat Send input module
- Enable DAW automation of all Source related parameters
- Index selects which Virtual Source to Automate

### Spat Return

- Returns multichannel audio from a Spat Return output module to a DAW track
- Enable OSC communication for Return parameters

## Spat Room

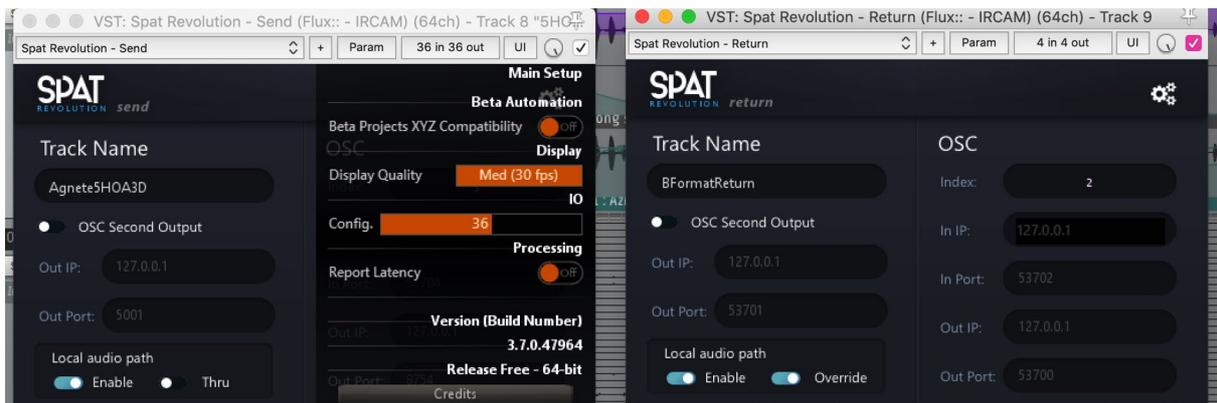
- Enable DAW automation of all Room related parameters
- Enable OSC communication for Room parameters
- Index selects which Virtual Room to Automate

In order for the audio software integration to function correctly, the user needs to take into account certain principles of configuration.

⚠ *Sample Rate must match in both Spat and the Plug-in Host*

⚠ *Buffer Size must match in both Spat and the Plug-in Host*

You can configure these settings in the Spat Revolution Preferences, and matching settings also need to be configured in the host DAW Preferences. Additionally, there is an IO configuration setting inside each plug-in, accessible from the small 'cogs' icon.



Set the IO Channel Count for each of the plug-ins this way. Each Plug-In instance can carry up to 64 channels to and from Spat. Once you have selected the channel count, *Enable* the software routing using the *Local audio path* switch.



If Spat is running, then a SEND or RETURN IO module will automatically appear in the *Environment Setup* labelled with the *Track Name* of the Spat plug-in, and set to the channel count you have configured in the plug-in. If all is well configuration wise, and a successful local audio stream has been established, the Send and Re-

turn modules will have a small green indicator.

★ *On some machines, you need to use the TAB key to register a new Track Name or IP address change in a Spat Plug in*

A few details on some of the other parameters;

**Index** - Relates the Plug in automation to a Virtual Source (see section 6.6)

- INDEX is assigned automatically and can only be changed manually to an Index number that is not yet in use by another Spat Plug In.

**Local audio path** - Enable the inter application software stream

**Thru** - Choose if you do not want to mute the send audio through the plugin

**OSC Second Output** - Set up a second parallel OSC destination

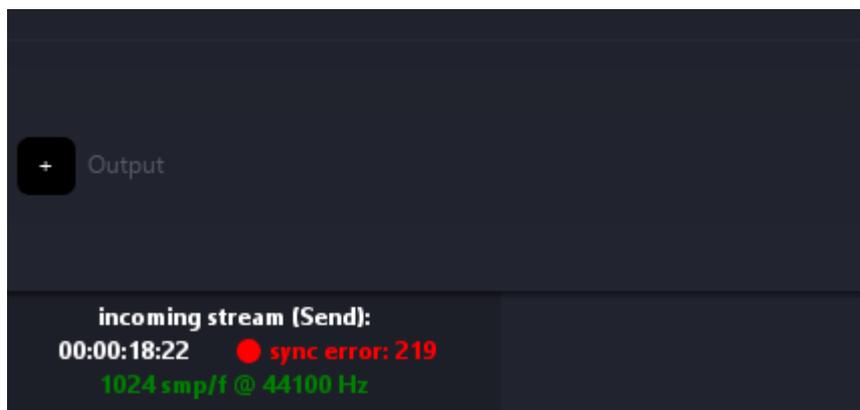
**Report Latency** - Activate latency compensation reporting for the DAW

**Override**- In Return only, over rides the DAW input path

## 7.4

# Troubleshooting Local Audio

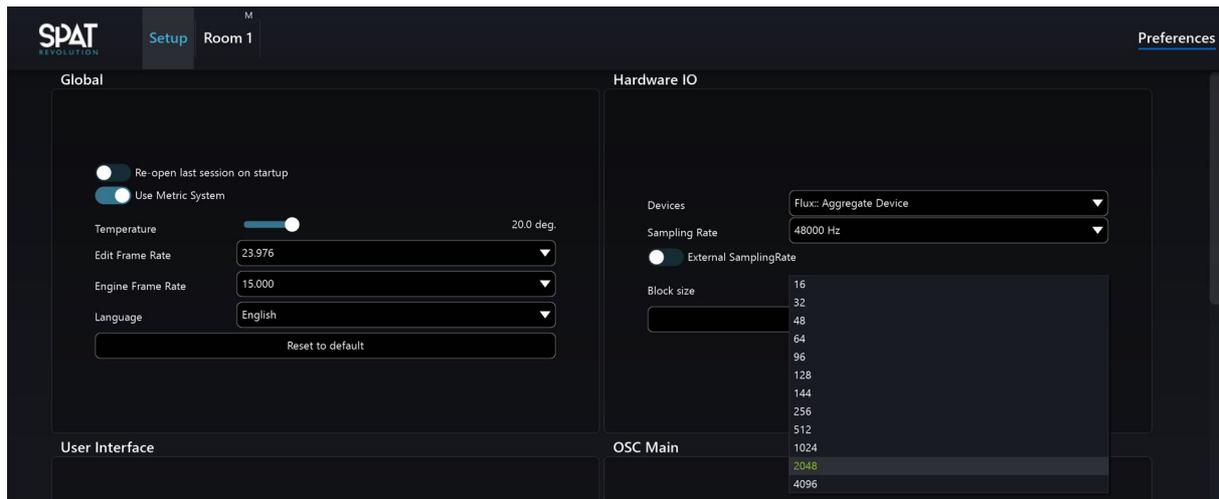
If you see red indicators on SEND and RETURN modules in Spat, and the Status bar is indicating dropped sync, then it is most likely to have a mismatch in Frame size or sample rate, or CPU performance is overloading.



Frame size ( sometimes called Buffer Size or Block Size) should be matched in the host DAW and Spat Revolution. If the audio processing is too demanding for your computer at the current block size and sample rate, you may also experience drop outs and sync problems due to CPU overload.

If you are experiencing lost sync when using Local Audio:

1. Increase the block size in the Spat preferences
2. Save your project and Quit Spat Revolution
3. Change the block size in your host DAW to match the new setting
4. Reopen Spat Revolution



There are also some performance preferences that may help in the case that your host machine CPU is overloading and causing audio glitches.

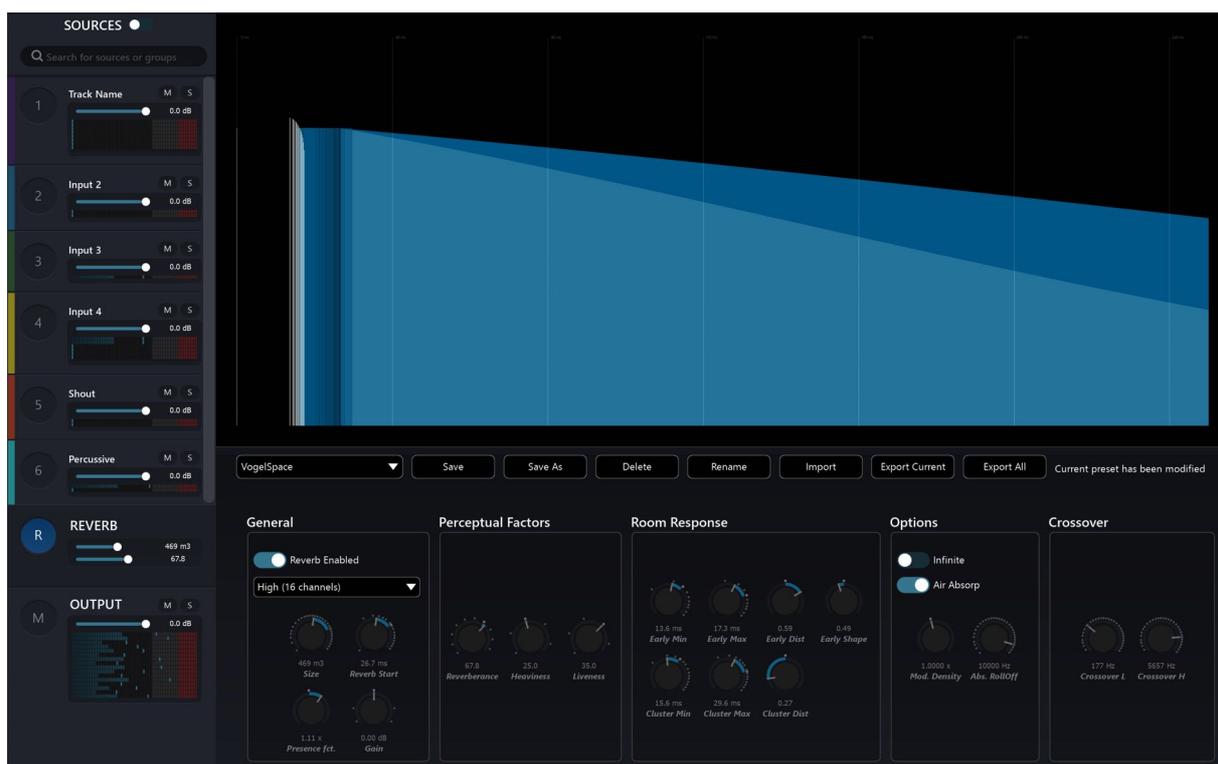
- Lower graphic frame rates
- Switching in the 'Parrallelised' processing options
- Turn the 'Nebula Alpha' to 0 in your Rooms
- Lowering Reverb Density to 8x8

To lower graphic frame rates, go to the Spat preferences. Changing the Engine Frame Rate will reduce pressure on the graphics updates and important when a host machine does not have dedicated GPU and CPU resource issues.

If you are still experiencing lost sync, then please take a look at the Appendix B of this guide for further troubleshooting tips. Also please read carefully the detailed advice in the Reaper and ProTools quick start guides, section 12 of this guide even if you don't use those DAW, as some of the advice there might apply to other set-ups.

# 8 Artificial Reverberation

Each *Virtual Room* in Spat can have its own artificial reverberation. Reverberation is a very important element in the psycho-acoustic perception of localised sources and immersive sound fields. The reverb processor in Spat is a multichannel algorithmic 3D reverb based on feedback delay networks. It also contains highly efficient multichannel real-time convolution without latency. The Spat reverberation engine is designed to synthesise the experience of the virtual sources and the listener all being placed within the same virtual acoustic space. Virtual spaces can be tuned, scaled and stored. Open the *Artificial Reverberation* graphical editor by clicking on the **(R)** index at the bottom of the list of Sources in the left side panel of the Room, or entering a room directly from a tab in the top *global bar* .



Internally the Spat Revolution reverb engine models many technical acoustic parameters, but the user interface has been simplified a great deal, to make artificial reverb design more straightforward and functional.

## ★ **Some Spat Reverb parameters control how the acoustics are perceived**

Alongside conventional tuning parameters which you might be familiar with, you will also find perceptual parameters, such as *heaviness* , *liveness* and *presence*.

These *Perceptual Reverb* parameters have been derived from the same IRCAM research experiments which were used to define the *Perceptual Factors* of sources - such as *warmth*, *envelopment* and *brilliance* . These can be found among the parameters for each virtual source (see section 9.3).

The Spat reverb designer can be used for a lot more than only simulating a "normal" acoustic space. For example, you could try to design a totally unreal space with continuously modulating acoustic properties or a space with infinite reverberation.

★ *Try switching the Infinite option for an immediately impressive immersive effect*

## 8.1

# Reverb Parameters

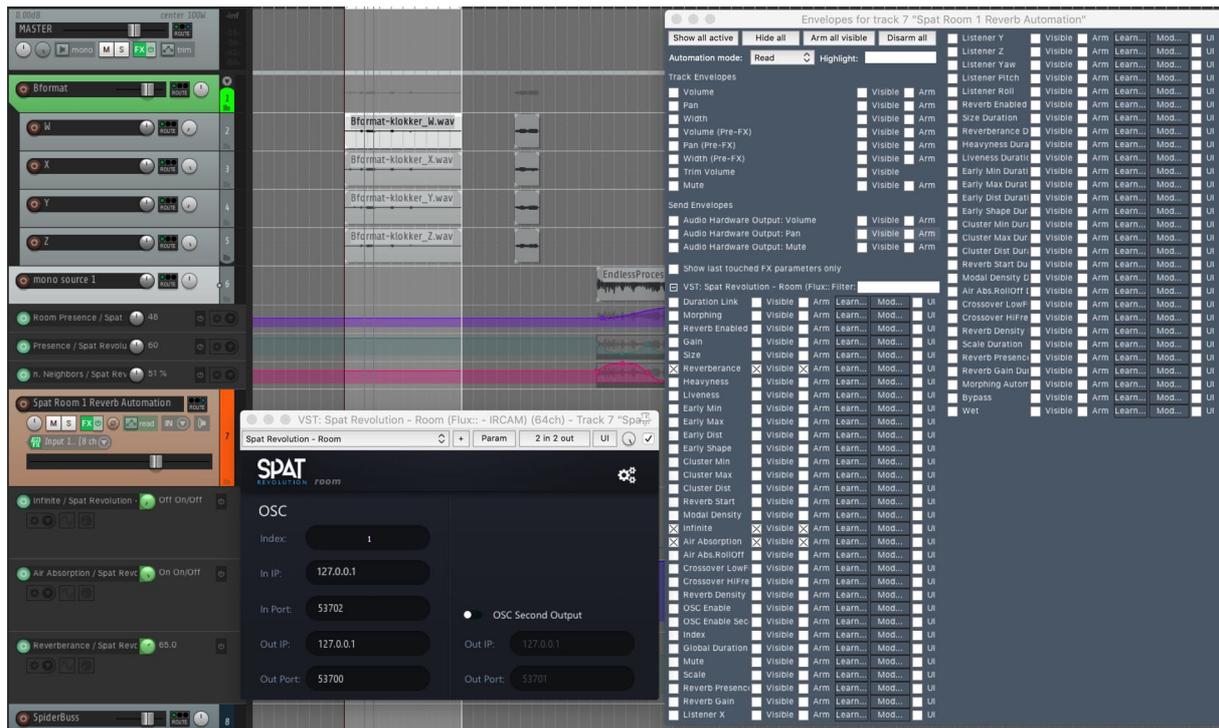
Every variable of the Virtual Room reverberation can be directly edited through the onscreen controls in realtime.

The reverb designer excels at creating static acoustic settings that will add all the dimensionality and immersive depth to a virtual scene. But it also invites more creative reverberation ideas. Remember it works in 3D and interacts deeply with the parametric design of all virtual objects that are expressed through it.

## **This is no ordinary reverb.**

The Spat Reverb is a true acoustic modelling multi-channel reverb, not just a so-called 'true stereo' reverb. Despite its internal complexity, the user is invited to morph and modulate the characteristics of the virtual acoustics. To make this process fluid and natural, the parameter controls have been carefully designed so that they do not glitch. This invites continuous parametric modulation ideas, for designing out of this world reverberant spaces, in realtime.

★ *Every variable of each Virtual Room reverberation engine can be smoothly and continuously controlled via DAW automation and OSC*

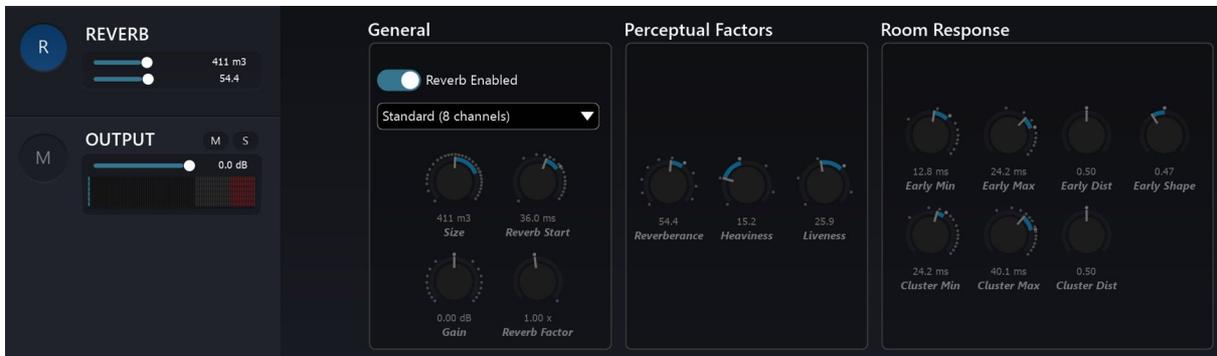


In order to automate settings from a DAW, you need to instantiate a *SPAT Room* plug-in which will open access to all of the parameter controls (see sections 7.3 and 10).

## 8.2

# Defaults

A double click on any Reverb Parameter dial will reset it to a Spat default setting. The default setting of a parameter is indicated around a dial as a larger tick mark than the other tick marks. Additionally a range is graphically indicated between the default setting and the current setting of a variable parameter.



★ Use the defaults as points of reference in your spatial sound design

## 8.3

# Preset Memories

Each parameter has the possibility to store useful preset settings of your own choosing. Right click on a parameter dial, and a contextual menu will pop up. From there you can store the current setting to a Memory Slot, or Recall a setting from a previously save memory slot.



## 8.4

# Reverb General



## Reverb Enabled

Toggles the entire reverberation engine for the room.

## Reverb Density

Internally, spatial variations are computed using a kind of 2D-network of reverbs, and this setting toggles between an 8x8 (standard) or 16x16 size (high). The choice of which sounds best is left up to you, as this depends on the source material at hand, although it must be emphasised that the high density setting consumes a little more CPU and that the colour of the reverb can be altered by this setting, particularly at some extreme parameter setting combinations.

## Size

This is a meta-parameter that takes care of varying several other parameters in order to quickly set the equivalent size of the virtual room, adjusting early, cluster and tail reverb parameters to match the room characteristics.

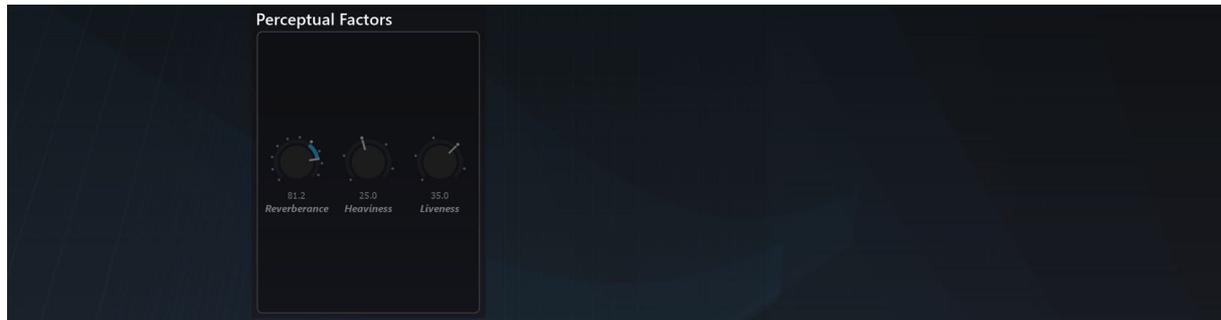
## Reverb Start

Reverb start sets the duration between the direct, dry source signal, and the first late reflections, or start of the reverb tail. Please note its value can never go below

that of the cluster minimum time as the reverb tail is fed with a signal derived from the cluster section.

## 8.5

# Reverb Perceptual Factors



## Reverberance

*Reverberance* affects the amount by which the listener perceives the music to be prolonged by the reverb, when the musical message suddenly stops. The effect of this setting is also obvious when the source material is of percussive nature. *Reverberance* is tightly related to overall decay time of mid- frequencies, which in turn is the time taken by the late reflections to vanish into silence.

## Heaviness

Relative decay time of low-frequency content, relative to the reverberance.

## Liveness

Relative decay time of high-frequency content, relative to the reverberance. Describes the liveliness and movement associated with the reverb tail (late reflections).

## 8.6

# Room Response Parameters



*Early* refers to *Early Reflections* stage of the Room response which is one of the most significant stages involved in our rapid aural perception of spatial properties and sound source localisation.

*Cluster* refers to a secondary iteration of room response reflections and is quite significant in the cognition of room acoustics.

*Tail* refers to the diffuse reverberations that eventually decay in a direct relationship with the size and reflectivity of an acoustic space. The tail section of a reverb does not contribute much to the localisability of a sound source in a space, but instead gives a sense of depth and ambience.

## Early Min

Early reflections minimum time, i.e. the time at which the early reflections start to appear, in milliseconds. This is the analogous of the ubiquitous "pre-delay" setting found on most reverberation processors. It represent the time between the direct sound and the first early reflection.

## Early Max

Early reflections maximum time, i.e. the time at which these cease to appear.

## **Early Dist**

Early reflections distribution. Determines the way early reflections are scattered in time, inside the Early Min. / Early Max. interval. The default setting of 0.5 corresponds to regularly spaced reflections, above these are more grouped towards the Early Max. value, and vice-versa.

## **Early Shape**

Governs the amplitude rise or fall of early reflections. The default setting of 0.5 corresponds to early reflections all having the same level. This mimics an acoustical space where reflective surfaces are all located at roughly the same distance to the listener. Below 0.5 early reflections decay with time, above 0.5 they rise with time. Early reflections of decreasing level would be typical of a space where most of the reflective surfaces are grouped at a range closest to the listener.

## **Cluster Min**

See Early Min. Please keep in mind the cluster is fed with the input of the early reflections processor section, as is shown accordingly on the display.

## **Cluster Max**

See Early Max.

## **Cluster Distribution**

See Early Distribution.

## 8.7

# Reverb Options



## Infinite

When activated, decay time temporarily rises to infinity, making the signal recirculate indefinitely inside the reverberation engine. This is best suited for one-off special effects such as “deep-freezing” audio material, or if you’re looking to create something a little less conventional than a fade-out for the end of your track.

## Air Absorb

Simulates the frequency-dependent absorption of air, where high frequencies roll-off quicker than low-frequencies with respect to distance. You’ve most probably noticed this real-world phenomenon when you’re far away from a concert venue and only able to hear the bass, and gradually start to hear the whole mix as you get closer.

## **Modal Density**

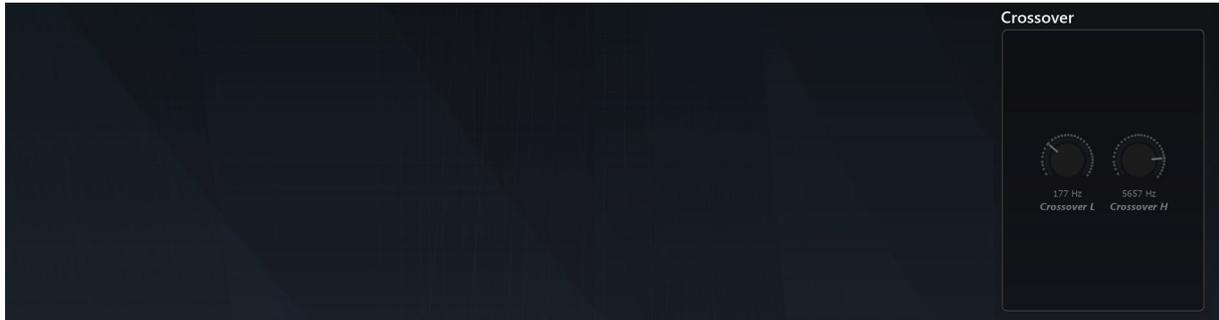
Scales the modal density with respect to the current setting, which is internal to the plug-in engine, and depends on other parameters such as reverberation time, etc. The modal density governs the frequency “smoothness” of the verb engine. Increasing this setting reduces the graininess of the reverberation. Adjust to taste, depending on the source material and desired result.

## **Abs. RollOff**

Roll-off frequency for the air absorption simulation. Signal content above this frequency vanishes faster.

## 8.8

# Crossover



## Crossover L

Sets the frequency below which decay time is determined by the Heaviness setting, expressed in Hertz (Hz). Default value: 180 Hz

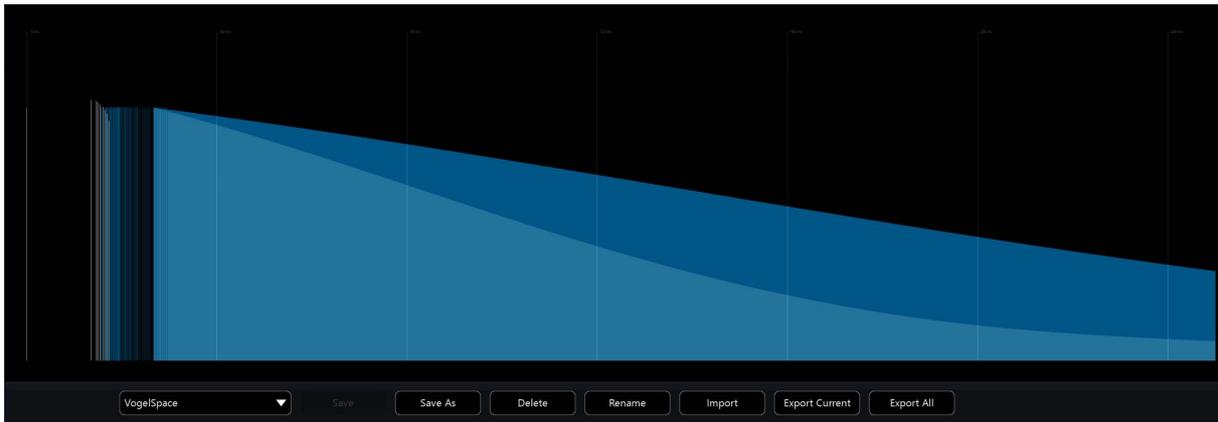
## Crossover H

Sets the frequency above which decay time is determined by the Liveliness setting, expressed in Hertz (Hz). Default value: 5657 Hz

## 8.9

# Reverb Design Presets

The Artificial Reverberation editor has its own preset management system, where you can save pre-designed models into a user defined preset list or to disk.



This is useful for building up a collection of pre-designed reverberation spaces and for designing models that might closely match the measurements of actual spaces you already know.

# 9 Source Parameters

Every virtual Source in a Room has its own set of variable parameters which define its simulated positional information, psycho acoustic properties, virtual acoustic properties and other options.

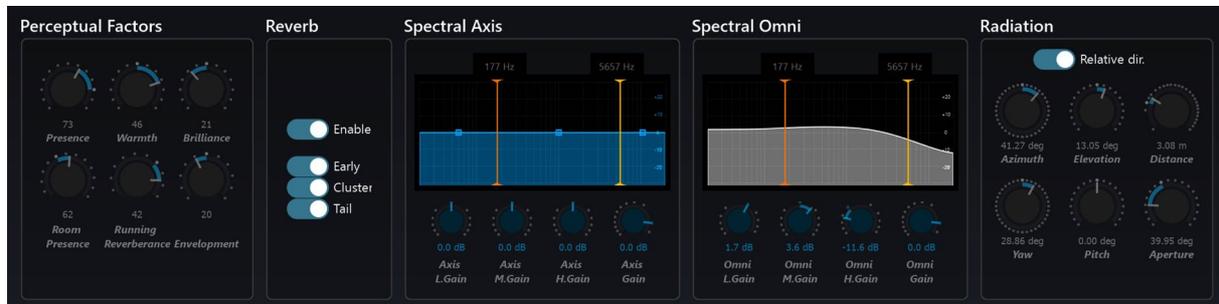
To edit the variables of a source in the *Source Parameter* editor, you must first be inside a Room. Select the source you want to edit from the list on the left side panel of the Room editor by left clicking on its Index number. Alternatively, grab its 'emitter' object in the 3D Room visualisation ( or just one of them, if the source is a multichannel group). When you select a source, the *Source Parameter* editor will pop up as a set of categorised groups with which you can alter the properties of the Virtual Source in the Room.

Additionally a *right click* on a Source Index number will bring up some further options, especially useful is the **Colour** option, which allows you to set an identification colour to a Source or Group.

## 9.1

# Defaults

A double click on any Source Parameter dial will reset it to a Spat default setting. The default setting of a parameter is indicated around a dial as a larger tick than the other tick marks. Additionally a range is graphically indicated between the default setting and the current setting of a variable parameter.

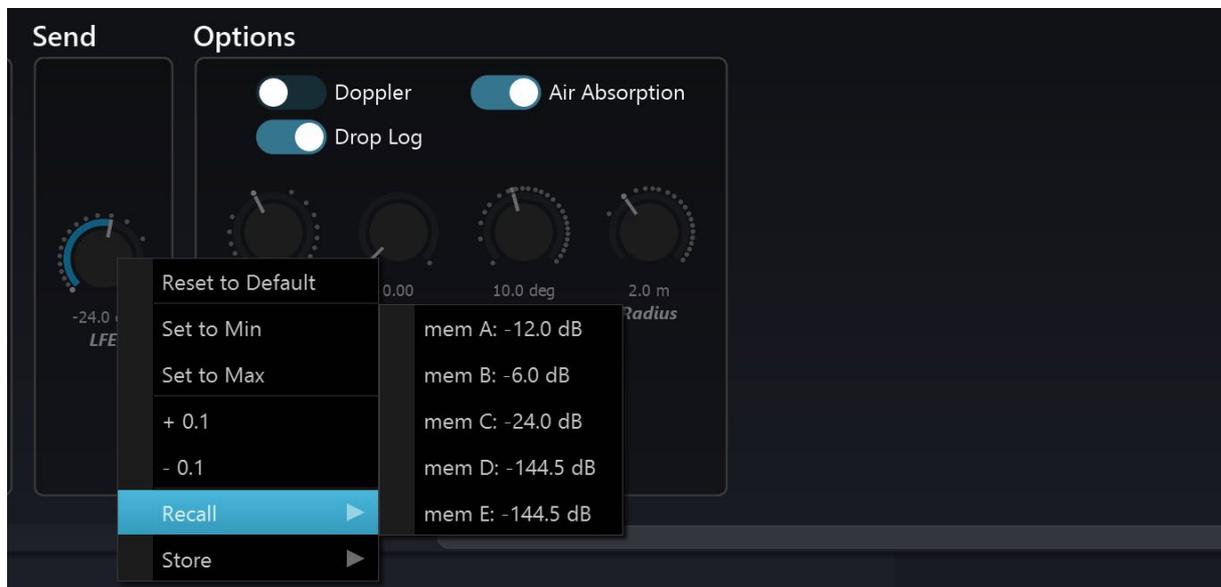


★ Use the defaults as points of reference in your spatial sound design

## 9.2

# Preset Memories

Each parameter has the possibility to store useful preset settings of your own choosing. Right click on a parameter dial, and a contextual menu will pop up. From there you can store the current setting to a Memory Slot, or Recall a setting from a previously save memory slot.



## 9.3

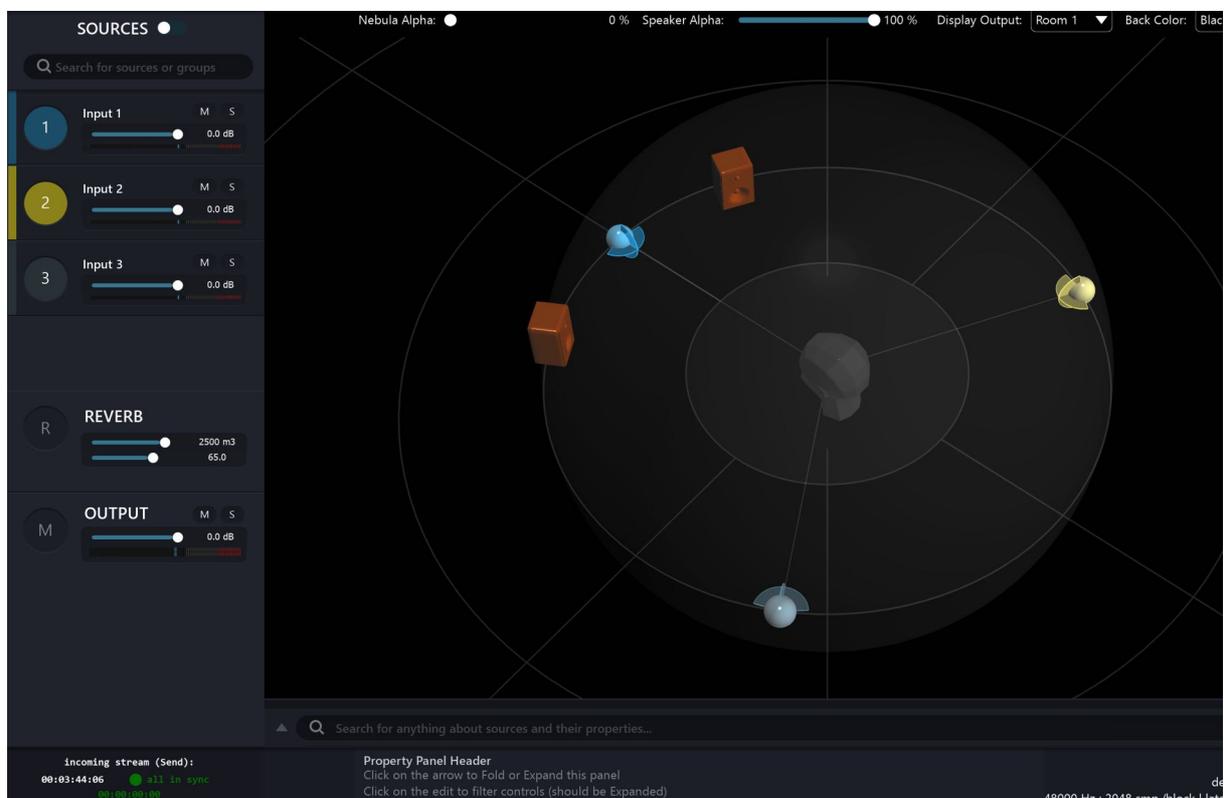
# Multiple Source Selection

You can shift-click on the Index number of separate Sources to create an ad-hoc edit group. When you have group Sources in this way, you can perform a number of group edit actions.

When you Right Click on an ad-hoc group selection a menu will pop up where you can:

- distribute the sources in the group evenly in a circle spread
- generate different colours for the sources
- reset the positions of the group

When you have selected an ad-hoc group using the shift-click technique, you can then pop-up the *Source Parameter* panel by clicking on the property panel header 'fold arrow' as shown in the screenshot below.



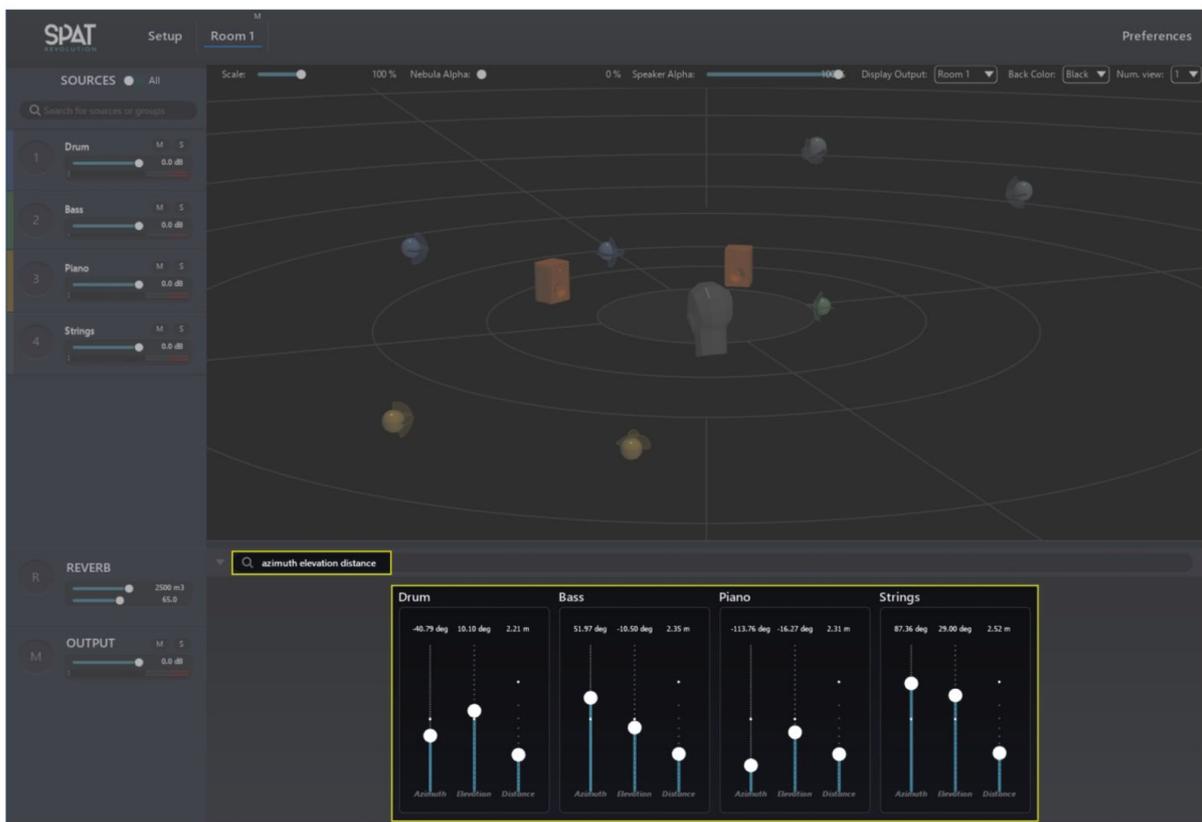
Any Source Parameter variables you adjust manually will assign that same setting on all selected sources in the group. A barycentric will then become practical to work from a center of mass perspective. For example, transformations like scaling, distance, rotation and directivity of the group is managed by Spat controlling each member of the group a barycentric relationship. Consider referring back to section 6.8 about groups in Rooms to read more about how they are represented.

## 9.4

# Smart Property Filter

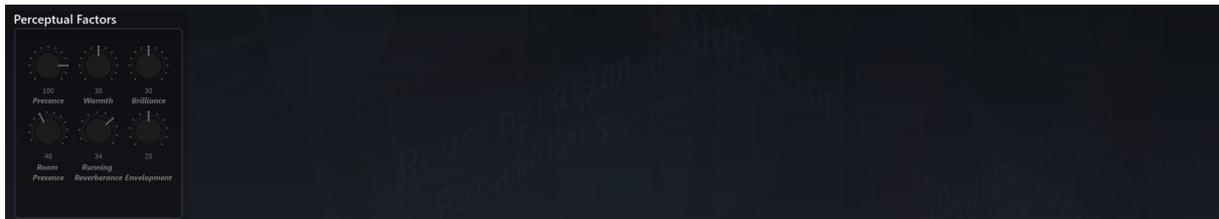
This feature allows you to display one or several parameters for all the sources that are in the same Room. It is a useful feature for fast editing.

Type "azimuth elevation distance" in the filter box for example, and you will see faders appear for only these properties, grouped for each of the sources as demonstrated in the following screenshot.



## 9.5

# Perceptual Factors



This parameter group holds settings affecting the way the sources direct and reverberated acoustic properties are perceived by the listener.

As touched on previously, these are not simply names stuck onto a single internal parameter dictated by the inner workings of the algorithm. Instead, a true perceptually-oriented approach was used in the design, where a test panel of listeners was presented with a test-set of sounds, constructed from several different variations of the reverb engine inner parameters. The listeners were then asked to rate each set onto a few different scales with perceptually and aesthetically meaningful names. Using principal components analysis (PCA) and optimisation techniques, we then built an algorithm which reverses the process and automatically maps a given set of perceptual factor values to the many internal reverb engine parameters.

As a general guideline, we encourage you to learn the meaning of these parameters by carefully listening to the audible characteristics when adjusting them. We do provide a short explanation of each of them below, but training your ears is really the best way to be able to use these in context.

## Presence

Source presence refers to the prominence of the direct sound with respect to the reverberated sound. It is not just equivalent to a dry/wet ratio, and is influenced by

other settings such as distance, radius and drop-factor.

## **Warmth**

Presence of the low frequency content part of the source.

## **Brillance**

Presence of the high frequency content part of the source.

## **Room Presence**

Prominence of the reverberation with respect to the source, or in other words, how much the room sound dominates the overall sound.

## **Running Reverberance**

This parameter controls the amount of perceived reverb when feeding a continuous music message, where the overall sound is a tight blend of the dry and wet signals and the reverb part cannot be mentally separated. It is linked to the early reflections decay time of the Spat Revolution Reverb engine. Note: this setting is not the same as the 'reverberance' in the *Reverb Properties*.

## **Envelopment**

Envelopment corresponds to the the perceived notion of how much the listener feels that they are surrounded or immersed by the ambient sound. In a multichannel configuration, one could describe this as the feeling of being wrapped inside the imaginary "sound sphere" that the listener pictures in her mind. It can also be described as the energy of the early room effect with respect to direct sound.

## 9.6

# Reverb Options



## Reverb Enabled

Toggles whether a source will use the reverberation engine.

## Early / Cluster / Tail

Toggle whether a source will use only some or all of the different reverberation stages.

*Early* refers to *Early Reflections* stage of the Room response which is one of the most significant stages involved in our rapid aural perception of spatial properties and sound source localisation.

*Cluster* refers to a secondary iteration of room response reflections and is quite significant in the cognition of room acoustics.

*Tail* refers to the diffuse reverberations that eventually decay in a direct relationship with the size and reflectivity of an acoustic space. The tail section of a reverb does not contribute much to the localisability of a sound source in a space, but instead gives a sense of depth and ambience.

## 9.7

# Axis Omni Filters



These two spectral processors can be considered as being equalisers that have been especially designed for virtual sound emitters simulated in virtual spaces.

## Spectral Omni

This filter section is for equalising the omni-directional part of the sound radiated by the virtual source. This equalizer mimics the global frequency response of the source, similar to how a loudspeaker colours the sound.

## Spectral Axis

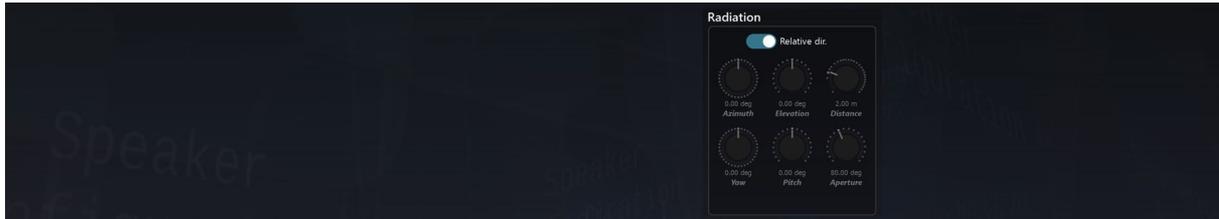
This filter section is for equalising the on-axis part of the sound radiated by the virtual source. Most, if not all commercially available loudspeakers do exhibit a radically different frequency response whether a listener or microphone is right in front of it or to the side.

Setting a rather flat on-axis equalizer curve, and maybe cutting the treble and mids for the omni response would be a good starting point to emulate a real-world speaker, so this is the default setting for these filters.

★ ***Spectral Axis* performs more like conventional mixer EQ**

## 9.8

# Radiation



This section controls the simulation of acoustic radiation in relation to the location and emitter orientation of the virtual source. Use these parameters to simulate how a source will interact inside the reverberant environment.

## Relative Direction

This switch is a very simple way for a user to achieve a more consistent result as regards the natural conservation of a source's presence. The algorithm will continuously maintain the on-axis focus for each virtual source - or for every emitter in a grouped source - so that it is consistently oriented towards the listener position. When not engaged, a source remains in the same constant direction which is what might be preferred if something is passing through in a spatial design.

## Distance

Distance from the source to the center reference point (listener position), in meters.

## Azimuth

Angle between the source location and the listener position front reference axis, in degrees.

## Elevation

Elevation angle, in degrees.

## Yaw

Angle of the source direct orientation relative to the listener-source axis, in degrees.

## Pitch

Source direct orientation pitch angle, in degrees. Think of *pitch* in the nautical sense of the word, how a boat *itches* up and down in stormy seas.

**★ Pitch and Yaw can be used to make a source more diffuse by turning its direct sound away from the listener**

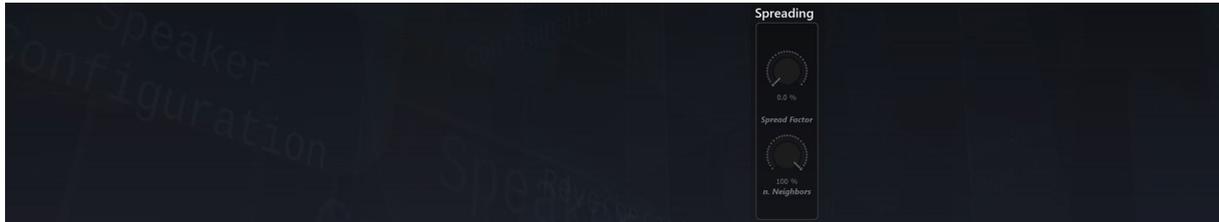
## Aperture

The aperture parameter relates to the “sound cone” projected by the virtual source in the acoustic space, and is measured in degrees. It determines whether the source will be very directive (small aperture), or omnidirectional (large aperture) inside the reverberant environment.

**★ Aperture can make a source 'activate' more of the acoustic space**

## 9.9

# Spreading



## Spread Factor

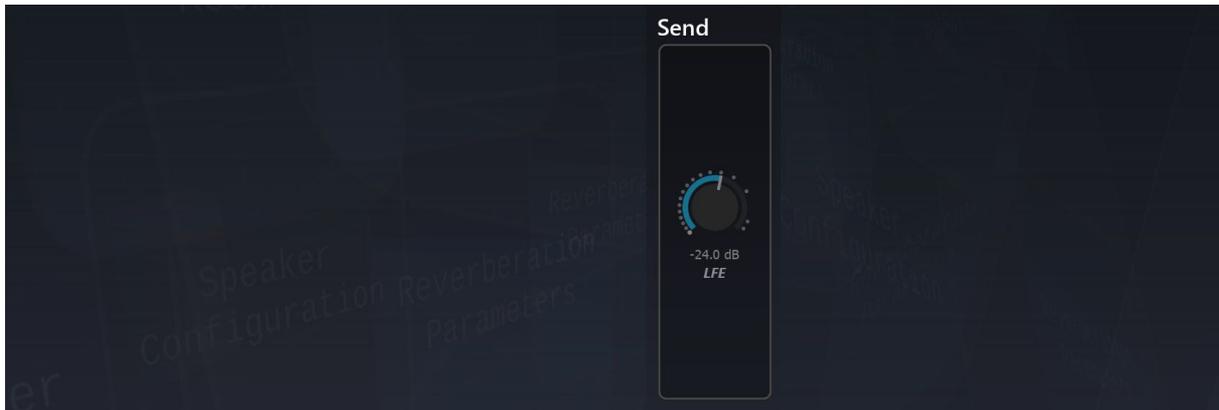
Spreading is a percentage factor that defines how a sound source will appear to spread out across speakers or virtual speakers. It is similar to *Aperture* in its focussing effect but will translate differently across certain channel based speaker arrangements according to how many speakers are involved.

## Nearest Neighbours

This parameter is only available to a source, if the room it is simulated in has been specified to be using the *K Nearest Neighbour* panning type (see section 5.64). It sets a maximum limit to the number of speakers that the algorithm can use as neighbours in its search for speakers to activate in relation to a virtual source. On a 10 speaker setup, 1-10 % will be the closest speaker to source. 11%-20% will be 2 and so forth.

## 9.10

### Send LFE



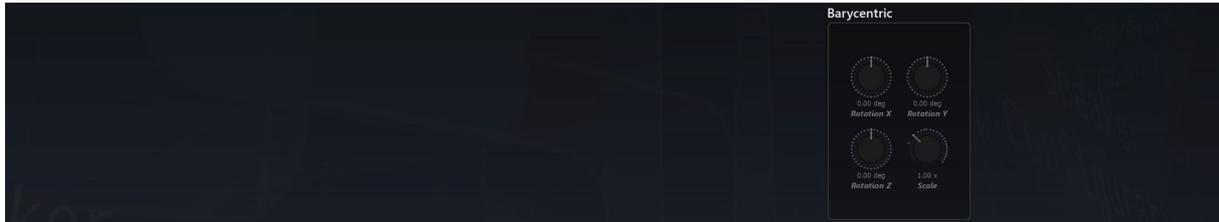
This variable parameter will only be available when the room is a Channel Based Format which features an LFE speaker in its configuration, such as DTU7.1 / 5.1 / AURO and ATMOS for example.

It will send an amount of the source into the dedicated LFE speaker channel of the output channel based configuration. Take a look at Appendix C for more sub bass / LFE workflows.

**★ Automate the LFE send for dynamic low frequency effects**

## 9.11

# Barycentric



These rotational transformations will only work on a virtual source that consists of more than one emitter in a grouped channel based arrangement. They will also become active, when you *shift-select* mono sources together to form an ad-hoc group or shift-select mono-sources in combination with grouped channel based sources. The 3 dimensional group rotations are calculated using a 'barycentric gravity' method to transform a network of sound sources constrained in a group relationship.

## Rotation XYZ

Rotate a group cluster around the XYZ axis of their common barycentric pivot point.

## Scale

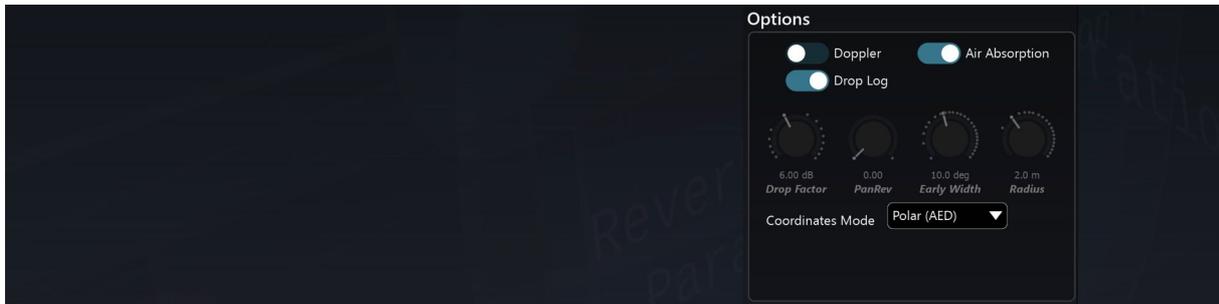
Scale the group cluster, maintaining their barycentre and relative relationships.

## Relative Direction

The barycentric transformations will continue to orient their on-axis energy towards the listener position, if the relative direction algorithm is enabled (see section 9.6)

## 9.12

# Options



Finally, there are some options available for each source.

## Doppler

The Doppler effect is a well-known wave propagation phenomenon where the height of a sound perceived from a listener standpoint rises when the source is accelerating, and falls when decelerating. This is the fire siren pitch going up then down when passing you. It will only be heard if you rapidly move the sources locations quite fast, but thanks to the virtual nature of the Spat, you can bypass Physics' laws and manually inhibit it using this switch, should it be unsuitable for the particular application you are dealing with.

## Air Absorption

Simulates the frequency-dependent absorption of air, where high frequencies roll-off quicker than low-frequencies with respect to distance. You have most probably noticed this phenomenon when you are far away from a concert venue and only able to hear the bass, and gradually start to hear the whole mix as you get closer.

## Drop Factor and Drop Log

Owing to a fundamental law of acoustics and geometry - namely energy conservation - sound pressure drops in level as one moves away from the source. Enable *Drop Log* for an acoustically accurate setting, which corresponds to a drop value attenuation every time the distance from the source is doubled (logarithmic behaviour). The default *Drop Factor* of 6db is also the acoustically accurate setting.

## Radius

Specifies the radius of a sphere or disc in meters, centered around the listener position, where the drop attenuation is not taken into account, and the sound level is kept constant with regards to distance. This is not only useful to prevent any dramatic sound level peak when placing a source too close to the listener, it also reflects real-world behaviour quite accurately, where sources do have a certain physical size, unlike point sources that are commonly used to model far-field acoustics. This “no-drop” zone is displayed as a transparent sphere of matching radius in the Room graphics.

## PanRev

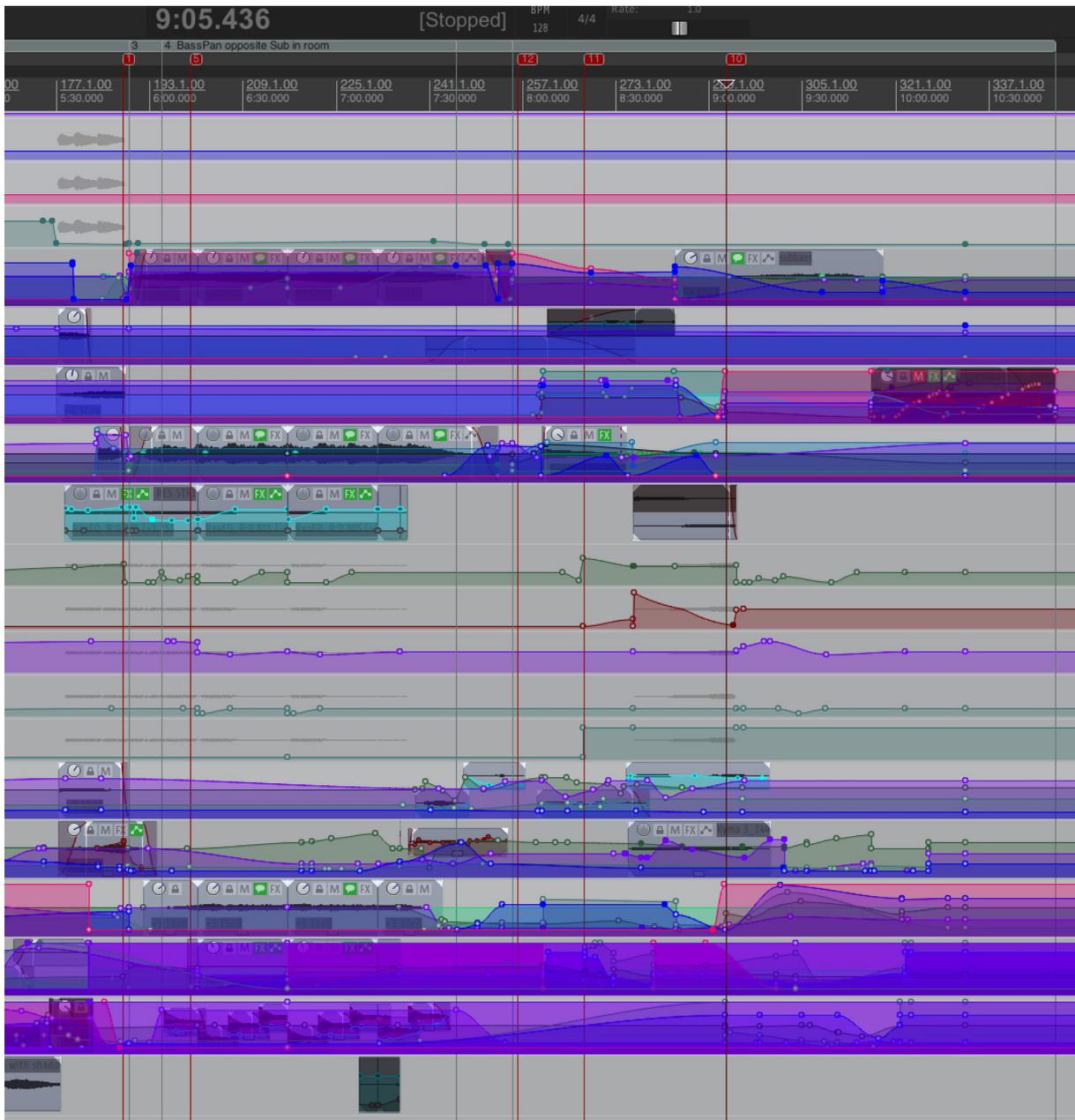
By default, only early reflections are panned, and the cluster reflections, which form the diffuse part of the early reverberation, are panned dead center. *PanRev* allows you to modify cluster panning, thus imparting some directionality or perceived direction to the diffuse part of the sound.

## Early Width

Controls the width of the sound projection lobe of the early reflections from a source in the virtual acoustic space, in degrees. The minimum setting, 1°, gives a very directional source, whereas 180° makes it omni-directional.

# 10 Automation

Almost all parameters of Spat Revolution can be continuously controlled in real-time using Spat's high resolution *automated control* features.



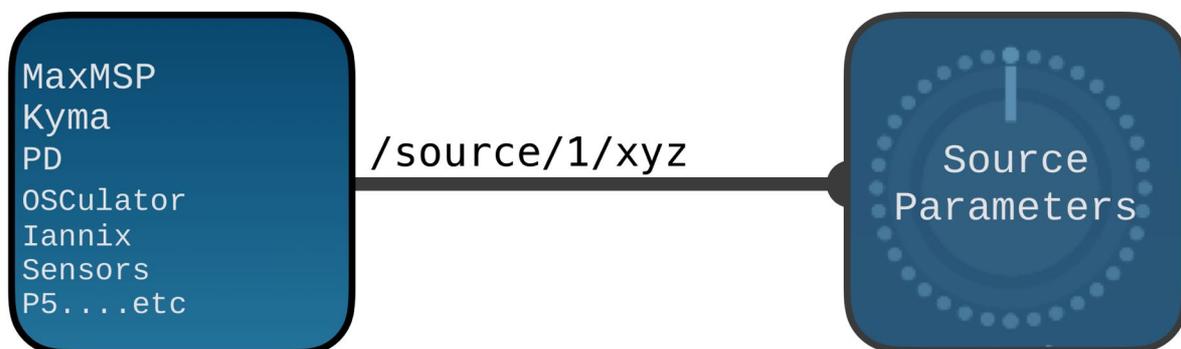
Control data can be sent via *Open Sound Control (OSC)* or through multiple SPAT plug-ins in your **DAW** . Parameter control can be played back from pre-composed

timelines, or performed, generated and captured in a variety of ways. One of the most straight-forward ways is to set DAW automation to Write mode on a track that contains a Spat Send plug-in and use the graphical interface of the Spat Room Editor to drag Virtual Sources around or turn source parameter pots. The movements will be captured into the DAW timeline in precisely the same way you would capture automation from a conventional DAW plug-in.

Alternatively, a wireless OSC control surface such as *LEMUR* running on an iPad could be used to control the parameters of Spat in realtime whilst you stand in the middle of the sound system away from the computer. There is already a fully programmed *Lemur* template for Spat ( [see Appendix Lemur](#) )

If you are using *Figure53 Qlab* for show control, spatial effects can be sent along with the rest of the audio visual cues for a big show or event ( [see Appendix QLab](#) )

If you are working with algorithmic gesture generators and modulators then your controls signals can be easily sent into the Spat Renderer via OSC to distribute and control spatial sound sources in realtime using your own control programs.

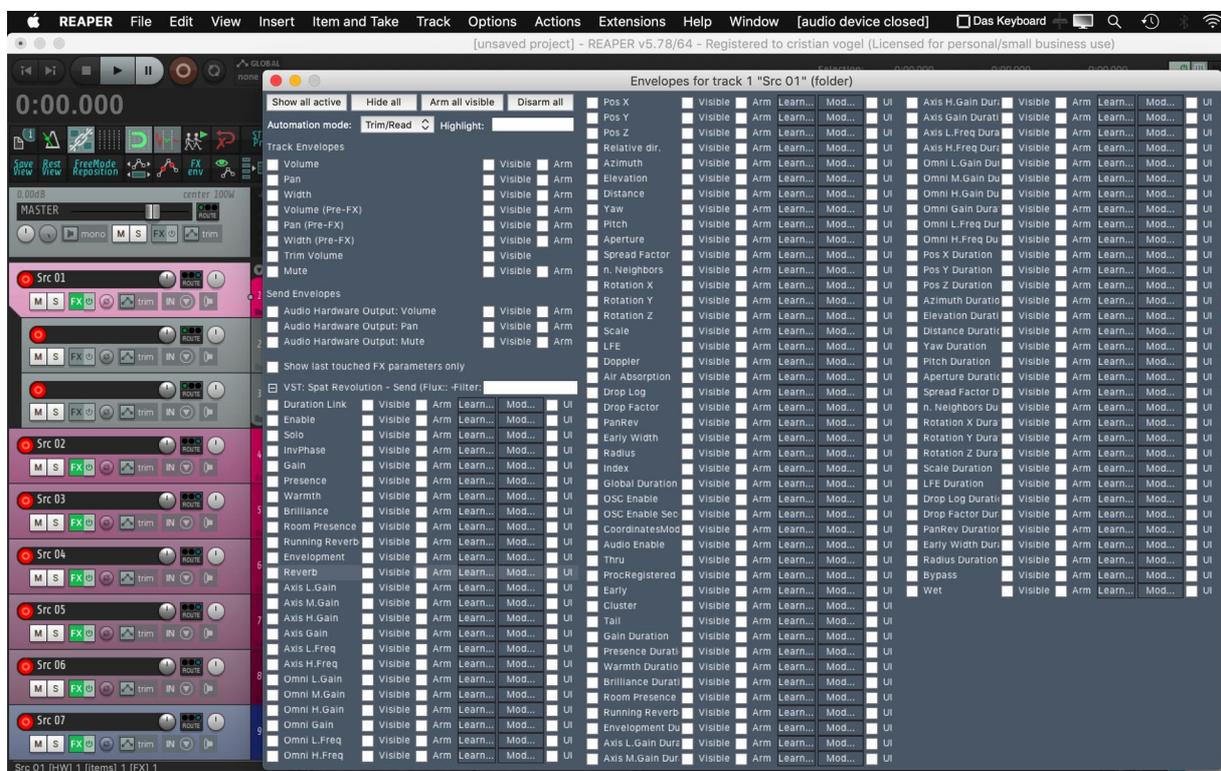


★ **Get creative with spatial sound design using OSC and Spat**

## 10.1

# DAW Automation - Local Audio Path

Whenever a Spat SEND plug-in is initiated in your DAW and you have enabled the local audio path workflow (see section 7.3) the plug-in automatically establishes a parameter control connection with the current Spat Revolution project. You can then automate every parameter available via the Spat plug-in from your DAW automation lanes, with no further configuration.



In the above screenshot, you see the *Reaper* DAW display all available automation parameters in a Spat SEND plug-in. Just arm and record or manually create automation lanes for the parameters you want to automate.

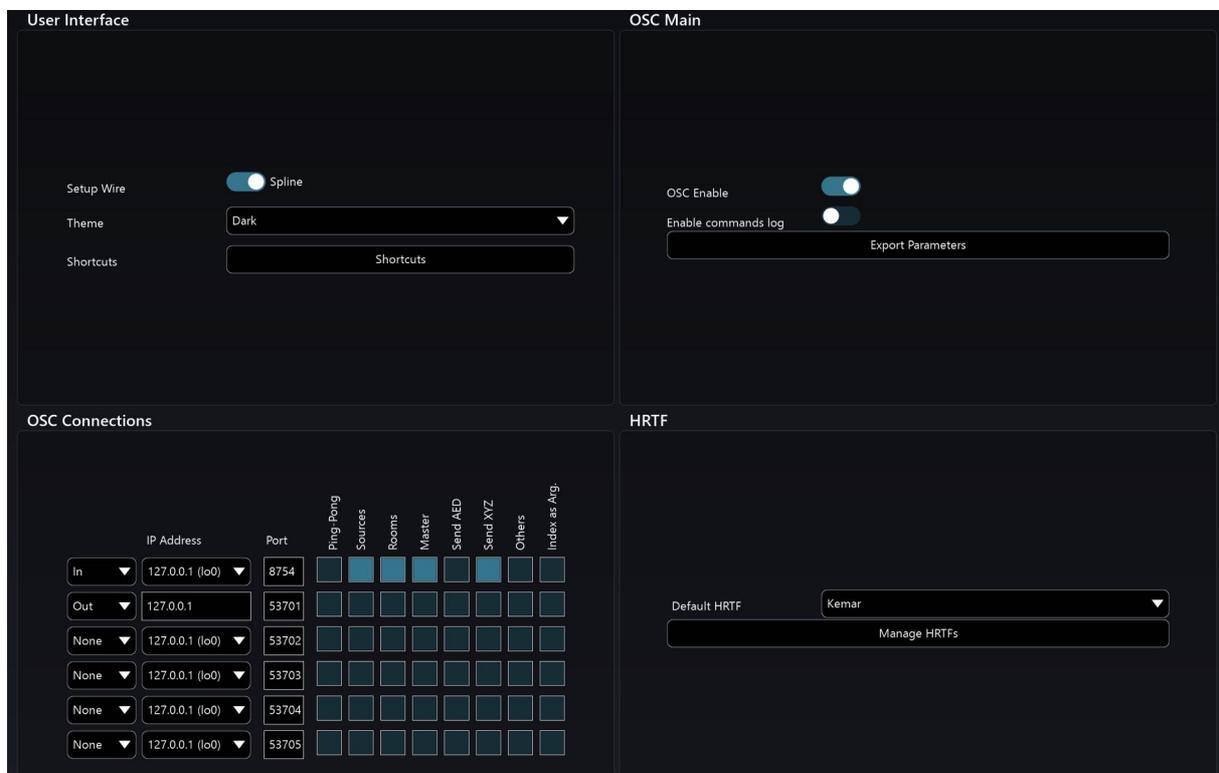
△ *Spat ROOM Plug-in automation requires manual OSC set up*

## 10.2

# DAW Automation - Manual setup

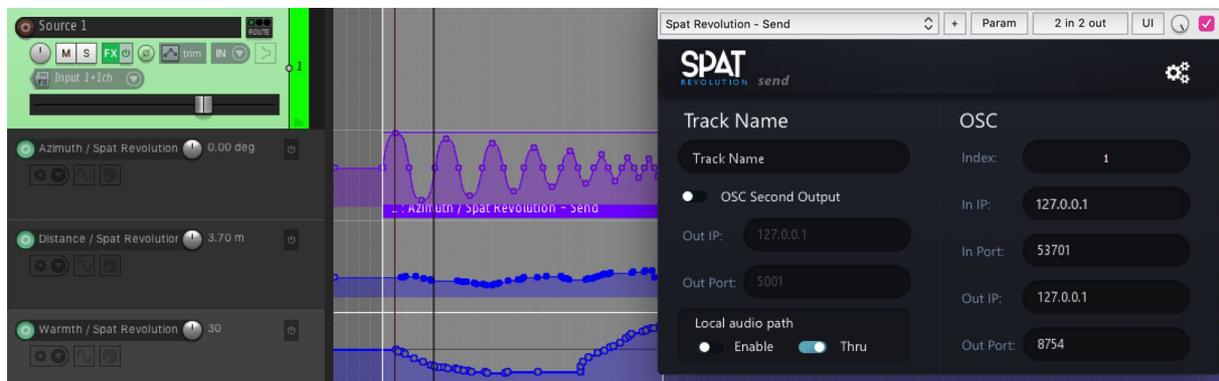
When you are not using the local audio path workflow, and instead working directly with a connected hardware interface, then you must setup automation manually.

The first step is to go to the *Spat Preferences* and *Enable OSC*



You then need to set up an OSC port and host route, in order to connect with Spat plug-ins via OSC. The port and host should match that of the Spat plug-ins. If you are running Spat and the DAW software on the same machine, then the IP address is always the so called *LocalHost* (127.0.0.1) - the port must be the same as in the Spat Plug-ins. Remember that the index number of each Spat plug in links it to a virtual source with the same index - or in the case of a Spat Room plugin - it is used to identify which Room you wish to control

⚠ **Some systems require you to press *TAB* key and not *ENTER* after editing a field in the Plug-in**



★ **All SEND plug-in instances in one DAW arrange will have the same IP in the DAW but different and unique *Index* numbers**

## 10.3

# OSC Connections Matrix

There is a lot of flexibility in the OSC Connections matrix. You find it in the Spat Preferences page. It is great for setting up networking, given the possibility of multiple network destination, source and message configurations.

	In	Out	IP Address	Port	Ping-Pong	Sources	Rooms	Master	Send AED	Send XYZ	Others	Index as Arg.
	In		127.0.0.1 (lo0)	8754	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Out		127.0.0.1	53701	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	None		127.0.0.1 (lo0)	53702	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	None		127.0.0.1 (lo0)	53703	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	None		127.0.0.1 (lo0)	53704	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	None		127.0.0.1 (lo0)	53705	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The matrix switches serve as a way to filter some messages out, if you are not using them or to alter the message format if you need to. For example, if you do not want to automate Room and Master parameters, you can switch them off in the matrix. These switches are useful for debugging and improving critical network performance.

The Matrix pull down menus identify a specified IP and Port to be either an **In** receiving data into Spat or an **Out** destination where Spat will broadcast control messages generated from user interactions with parameter dials and source positional data.

For example when you want to record movements of a Source's parameters and positional data, the Spat Send plug-in must be configured to be receiving on the same IP and Port number specified in (only one) **Out** destination in the OSC connections matrix.

About the activation filters:

*Send AED* - Activates Azimuth Elevation and Distance Polar co-ordinates

*Send XYZ* - Activates XYZ Cartesian co-ordinates

*Sources* - Activates all Source related automation

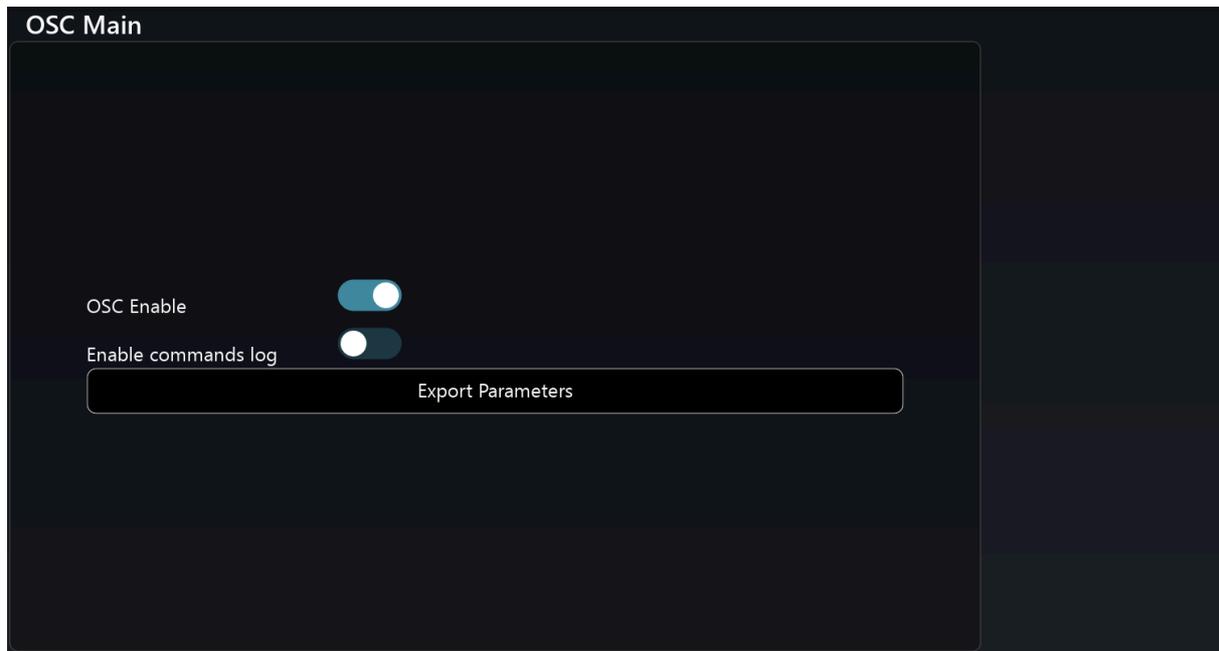
*Rooms* - Activates Rooms and reverb related automation parameters, only sent by Spat Room Plug in (not Spat Send)

*Master* - Activates master gains, mutes and solo related automation

-

## 10.4

# Introduction to Generic OSC



If you are developing your own control systems to integrate with Spat, you might find it useful to know that it is possible to export a detailed description of all OSC patterns, syntax and usage to a text file for reference. You will find that option in the Spat preferences.

**★ Enable commands log to confirm you are receiving data (Shift + F7 will open the log window)**

In general, Spat OSC patterns have the form of

```
/source/[index]/x
```

where [index] represents the Index of the Virtual Source or Room you wish to control with the message.

The three positional formats can be packed into one message if that option is set in the OSC Connections Matrix;

```
/source/[index]/xyz
```

*Cartesian co-ordinates in meters*

```
/source/[index]/aed
```

*Polar co-ordinates (azimuth, distance and elevation)*

**⚠ Take care to automate EITHER Cartesian OR Polar not both**

Sometimes, it is more convenient to have the [index] parameter as an argument of the OSC message. This option is available in the OSC Connections Matrix, namely `Index as Arg`. If this option is switched in, then the messages will be of the pattern;

```
/source/xyz iff
```

where `i` is an integer denoting the index of the target. `f` according to convention is a float denoting the values of the message.

For more details about the Spat OSC dictionary and its usage syntax please refer to [Appendix D](#)

# 11

## Set Up Examples



In this section, we will demonstrate a few possibilities for signal graph setup in the Spat Environment signal graph editor, representing examples of different working scenarios. There are many alternative ways to set up, so these examples are by no means exhaustive.

Before getting into specific situations, we will start with a few basic workflow features to keep in mind when designing your own setups.

## 11.1

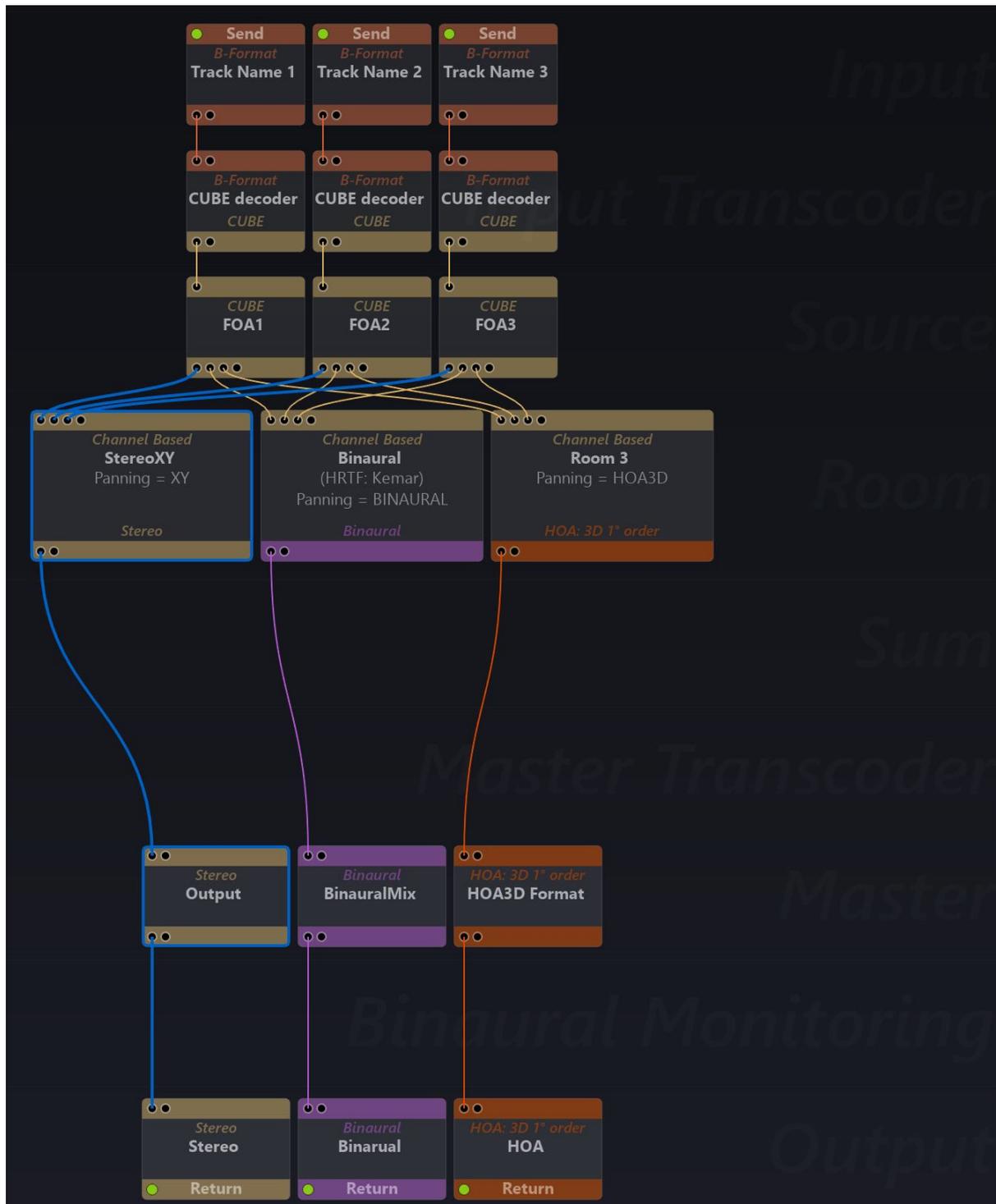
# Parallel Room Mixes

Use the lasso-selection or command-click selection to make a selection of all Source modules in the Sources row of the graph. Use the Connect Action to wire them into another Room module. This can be repeated, according to how many alternative Room mixes are needed.

The mix and automation of all sources will be rendered identically in each room, except of course the Room format, panning type and artificial reverberation could all be different.

The parallel Rooms technique is used to create simultaneous versions of the project mix at the outputs. For example a scene can be computed as a Channel Based mix, an HOA encoded mix and a Binaural mix simultaneously.

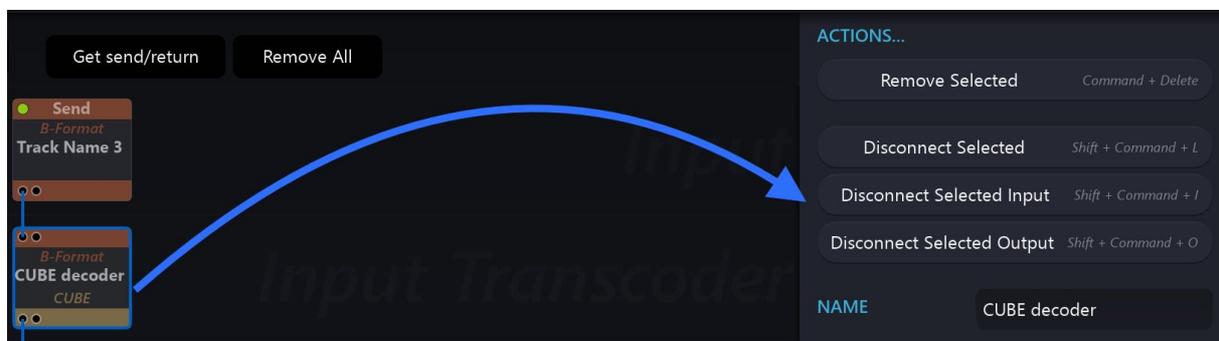
See example below.



## 11.2

# Disconnecting and Connecting

As mentioned earlier on, it is recommended to use *Disconnect Selected* type Actions before going for a *Remove Selection*. This is because the graph will automatically re-organise itself when Modules are deleted and this could be difficult to revert. It is also worth knowing that you can multi-select modules (using *Command-Click* or *Lasso select*) before choosing an action, and all modules in the selection will be affected by the action.



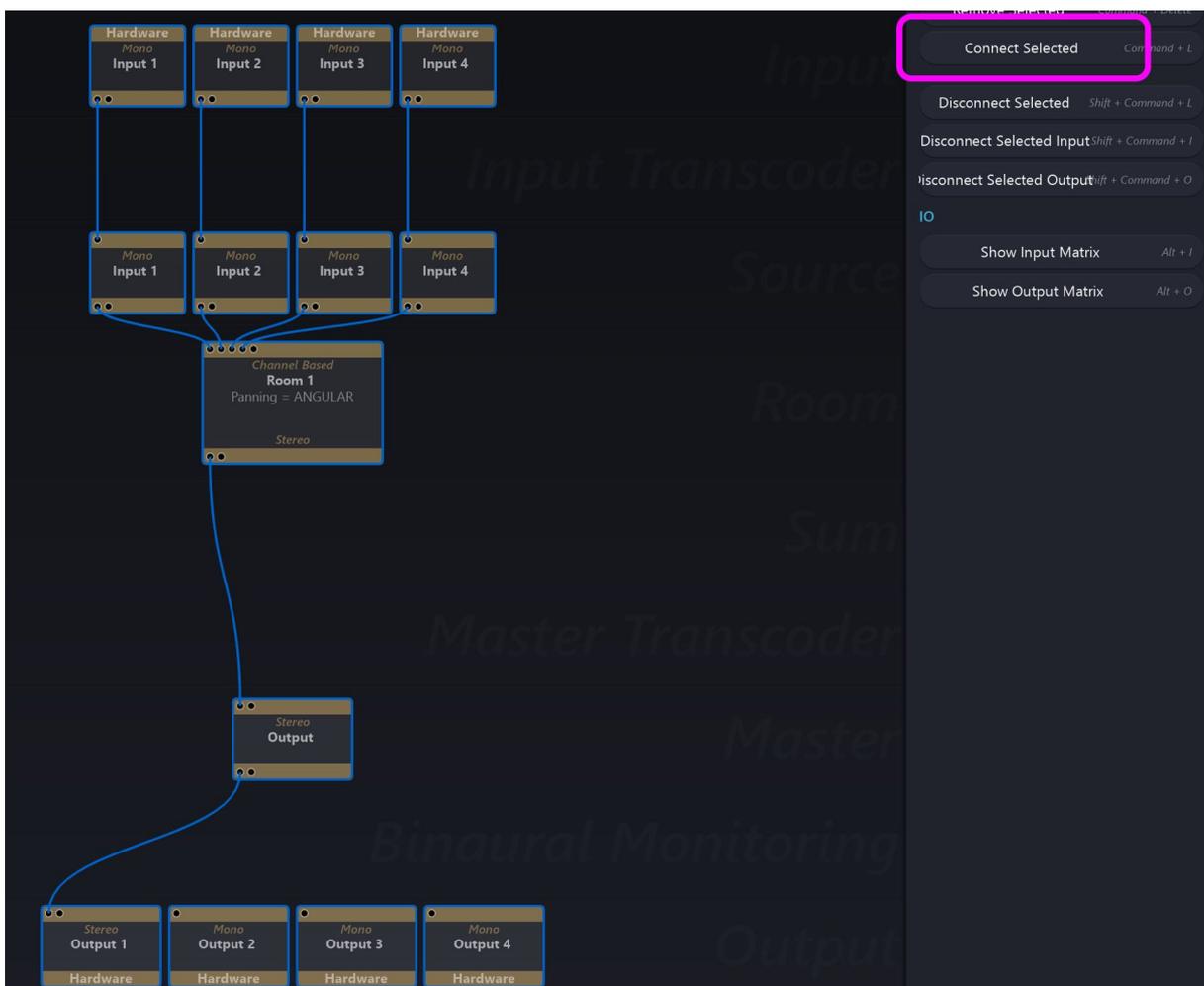
★ **Try not to Remove Modules until you are certain that this is the correct edit action.**

## 11.3

# Quick Start Graph

Spat will automatically wire up a suggested signal graph, adding modules in-between as needed when you choose a *Connect Selected* Action between modules. This is particularly useful for fast setup from scratch. Add some inputs on the top row, add some outputs on the bottom row. Select all and choose *Connect* . A starting signal graph will be generated automatically.

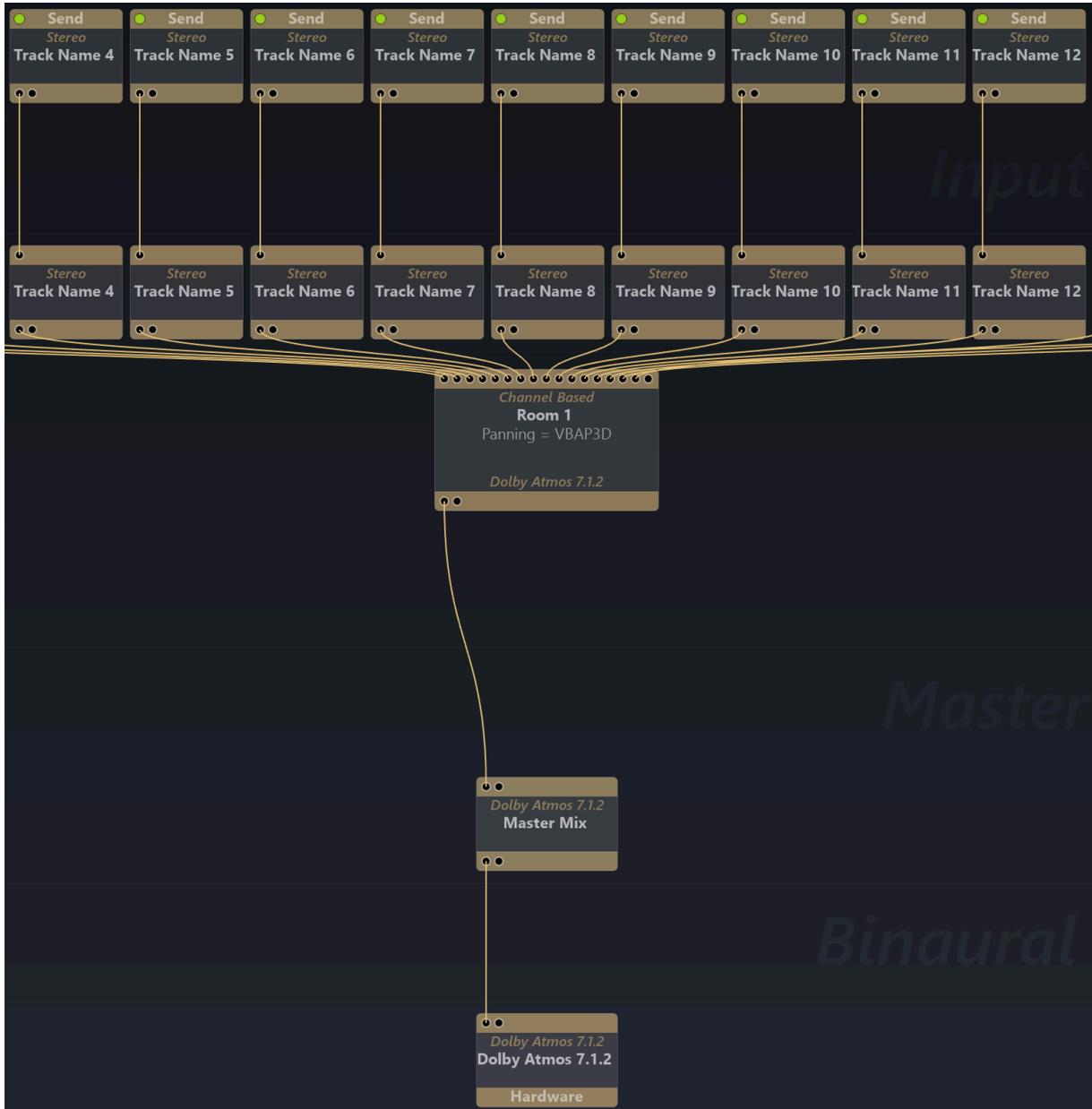
See example below.



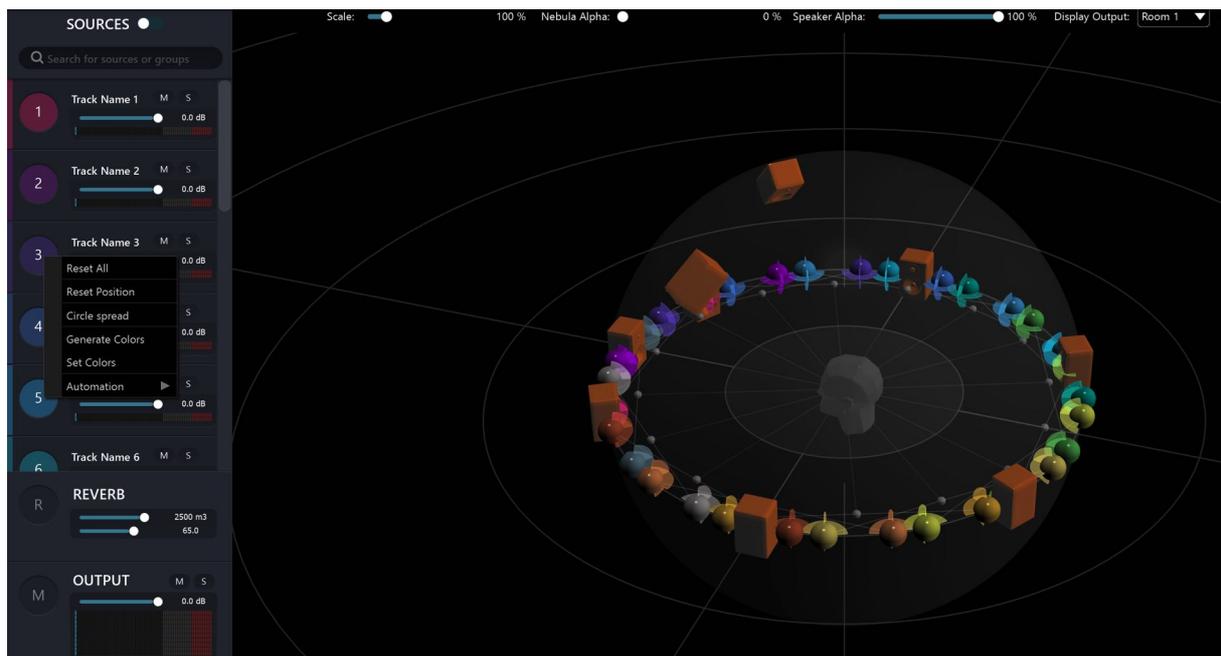


# Dolby Atmos 7.1.2 Room with 16 stereo sources

## Setup View



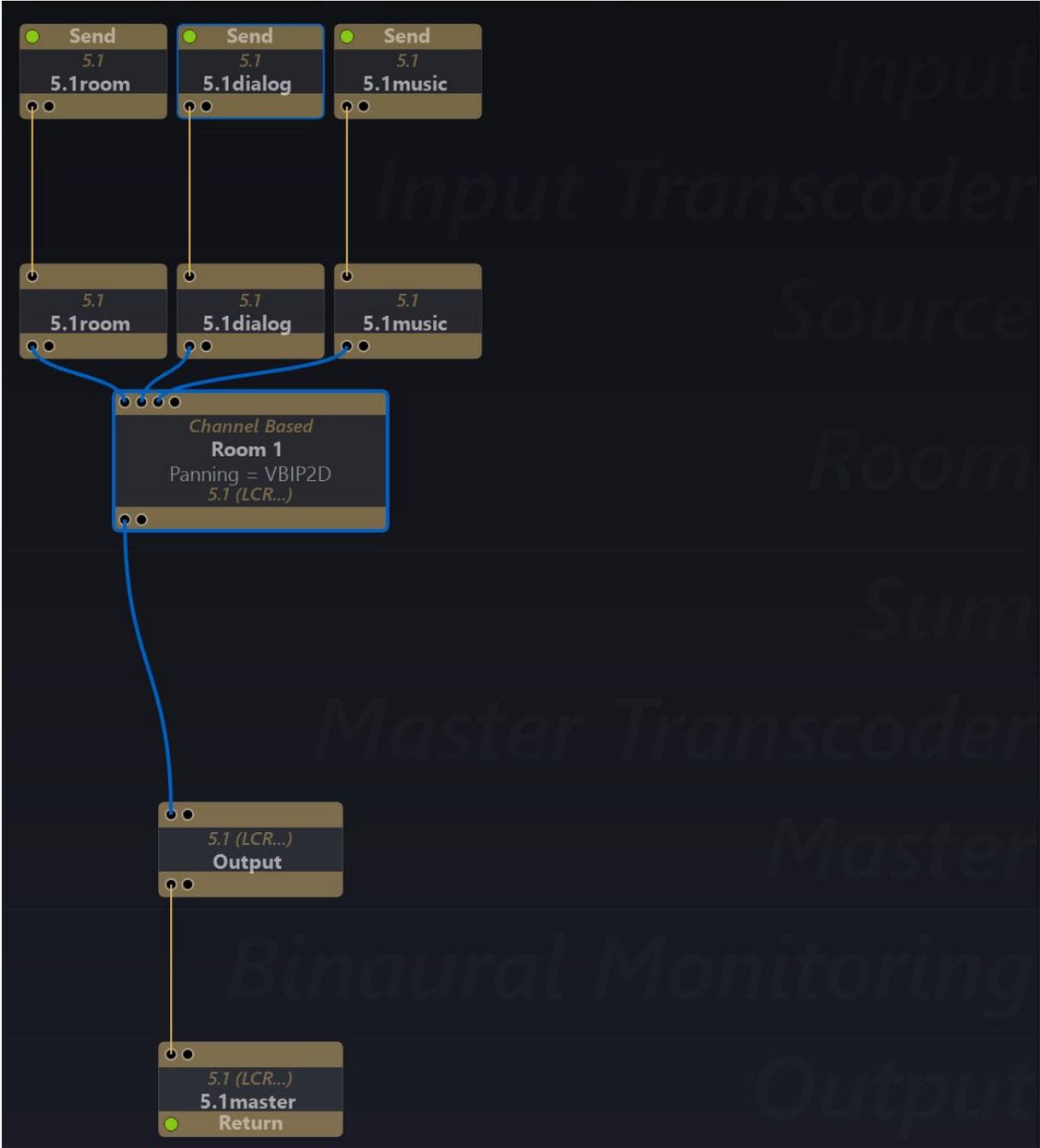
## Room View



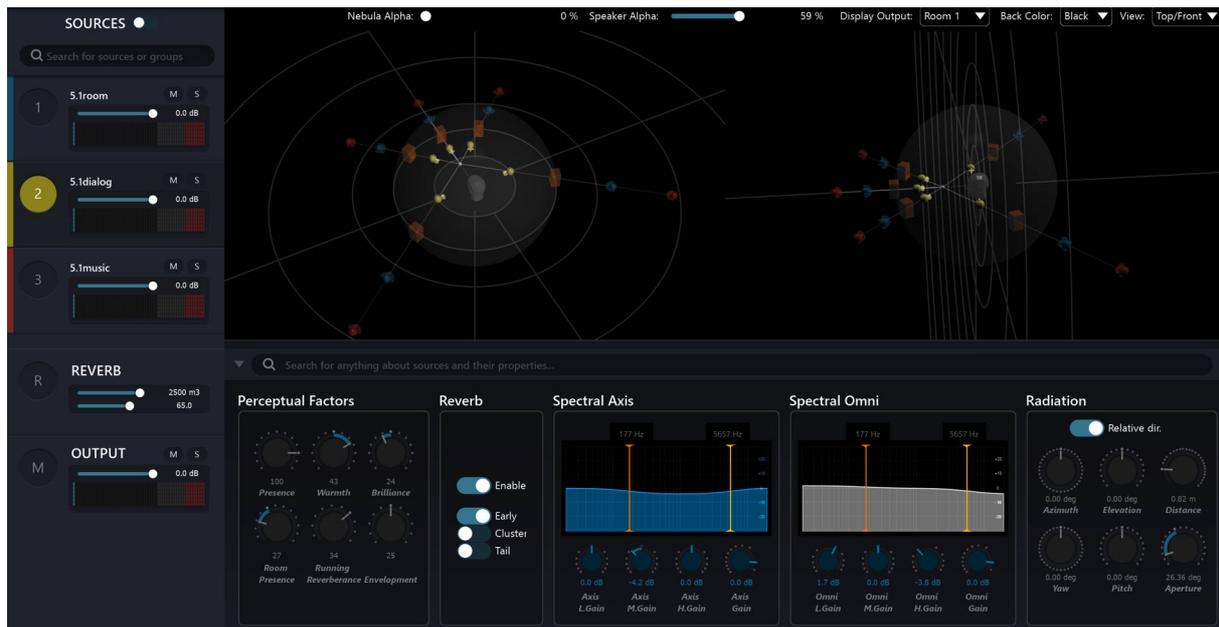
- Select All (*Command-A*)
- *Generate colours*
- *Circle spread*

# Mixing Surround Sources

## Setup View



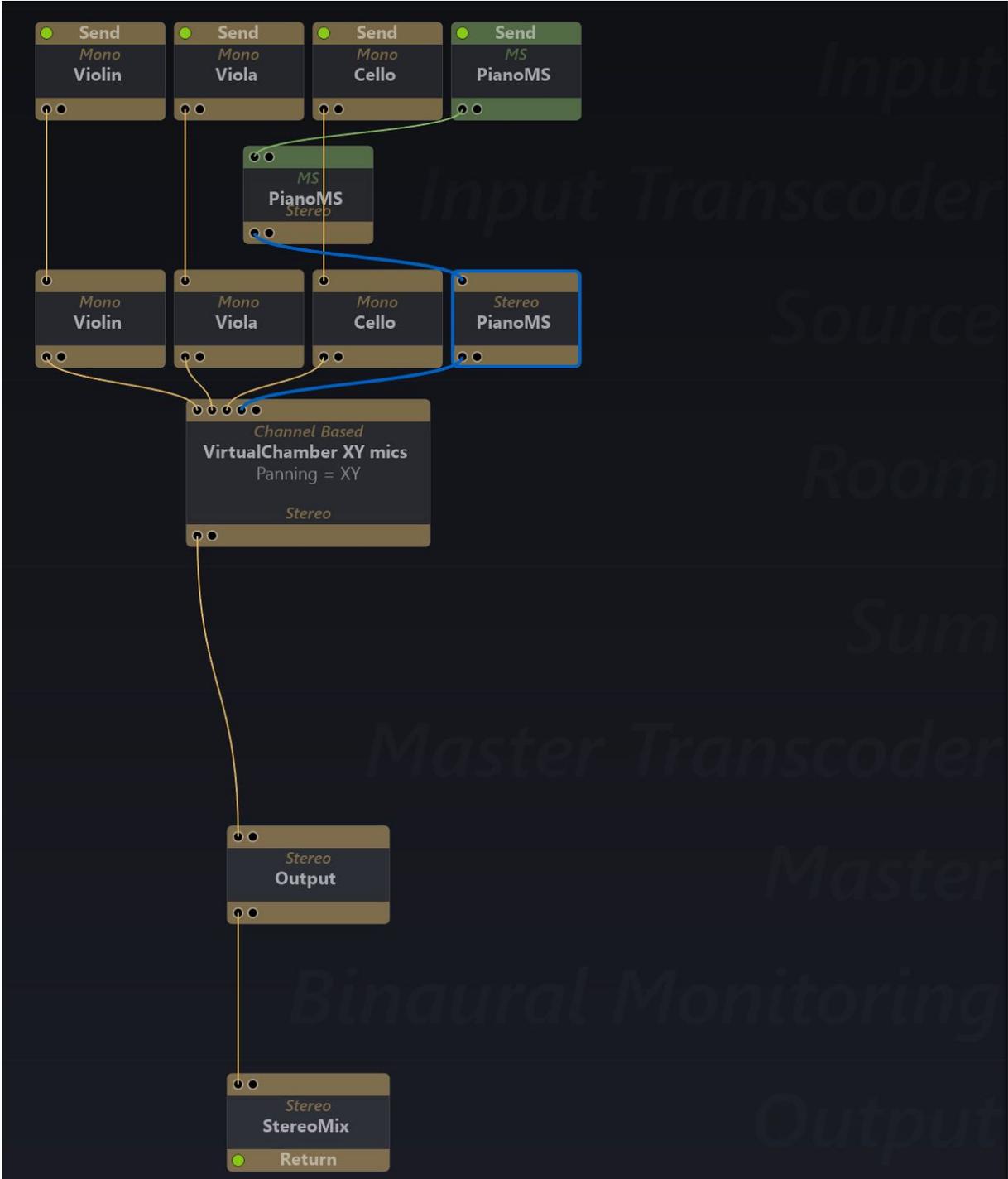
# Room View



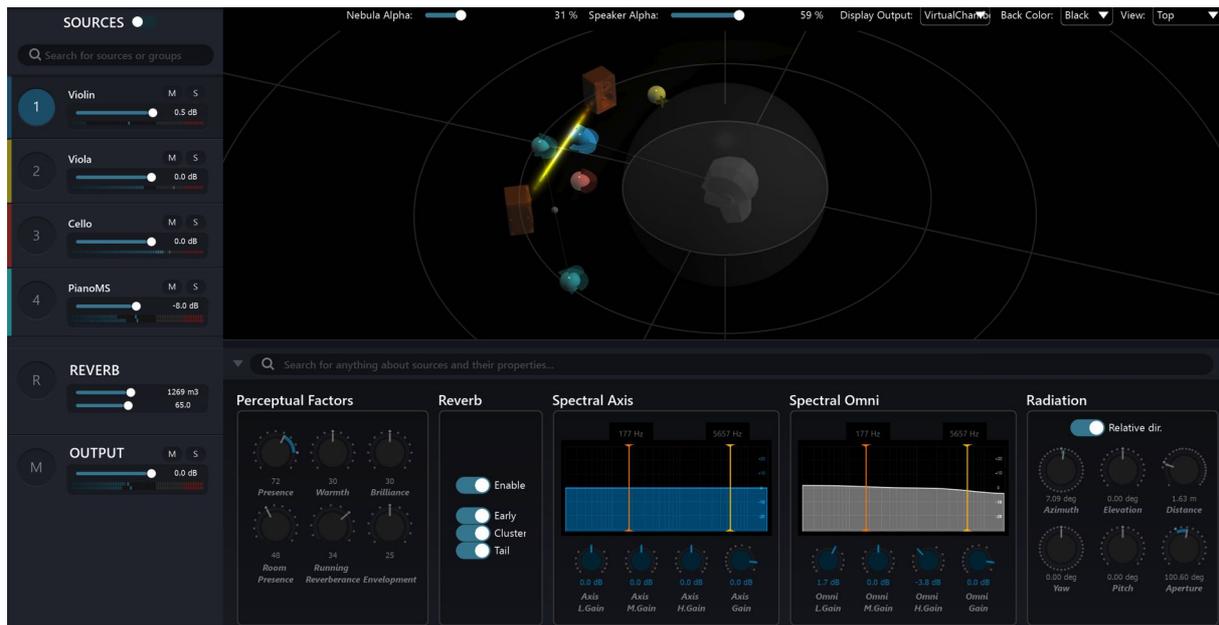
- Using *Scale* and natural distance drop off to mix
- Using *Aperture*, *Filters* & *Perceptual Factors* to mix
- Different Reverb sections enabled for each 5.1 group

# Mixing Acoustic Music in Stereo

## Setup View



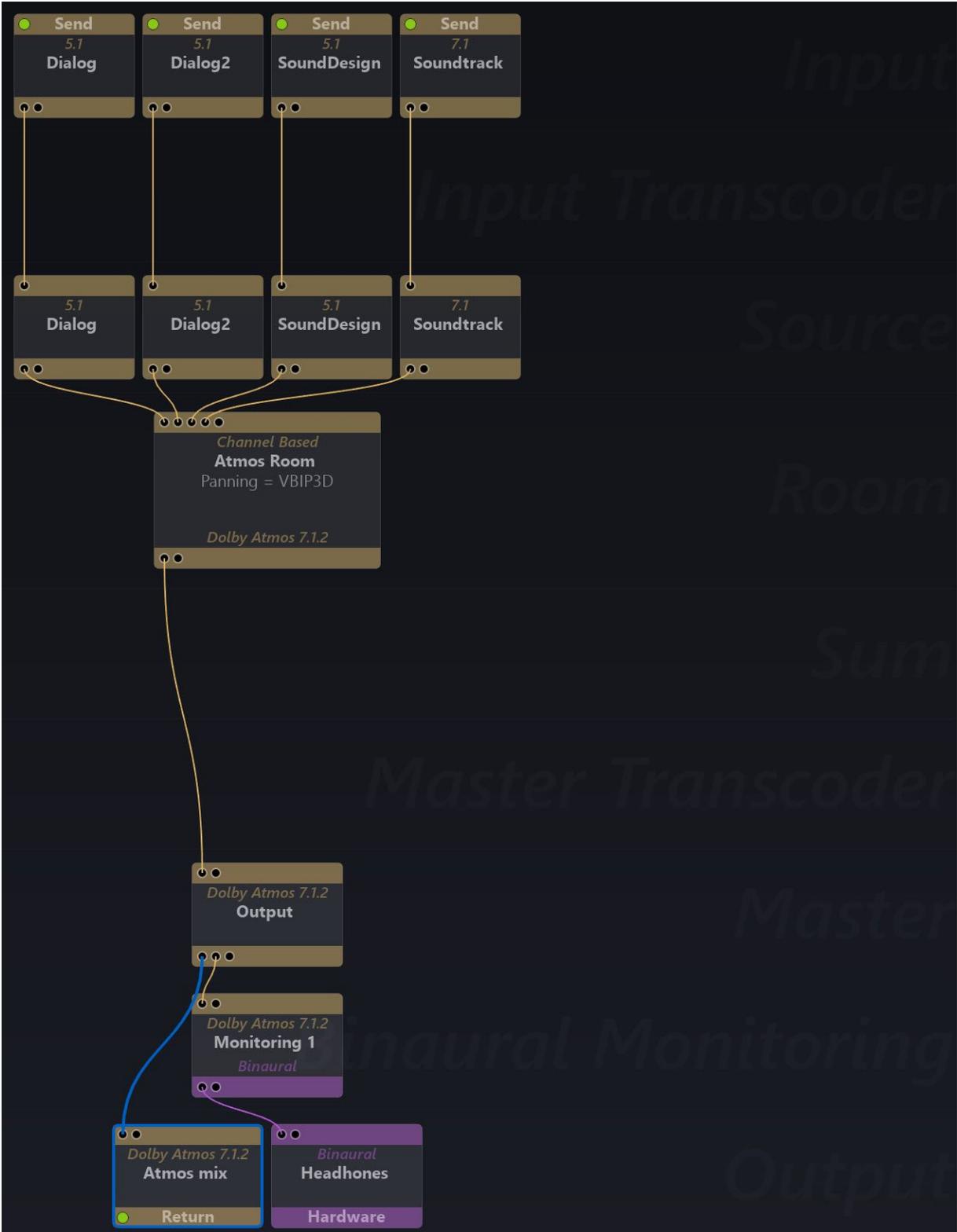
## Room View



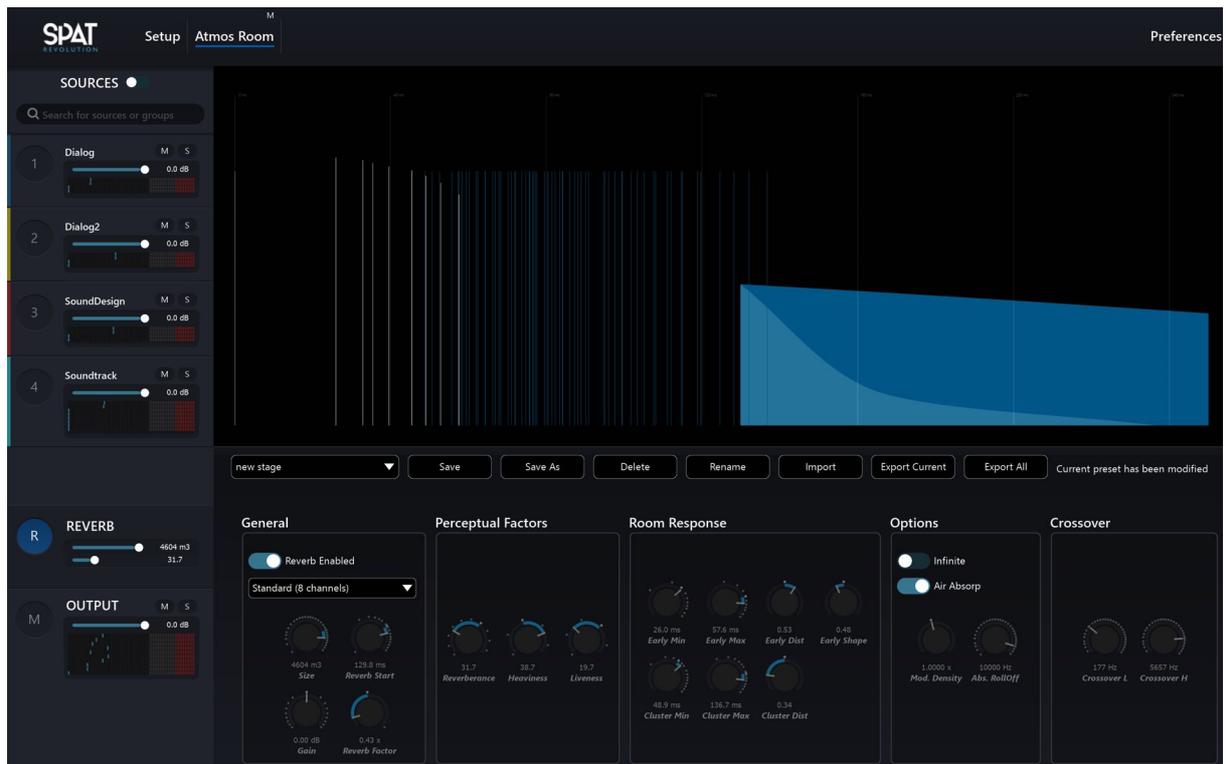
- Stereo Room with XY Microphone Panning Type
- Using *Filters Position & Perceptual Factors* to mix
- Small 'Chamber' like Reverberation design

# Upmixing Surround Formats

## Setup View

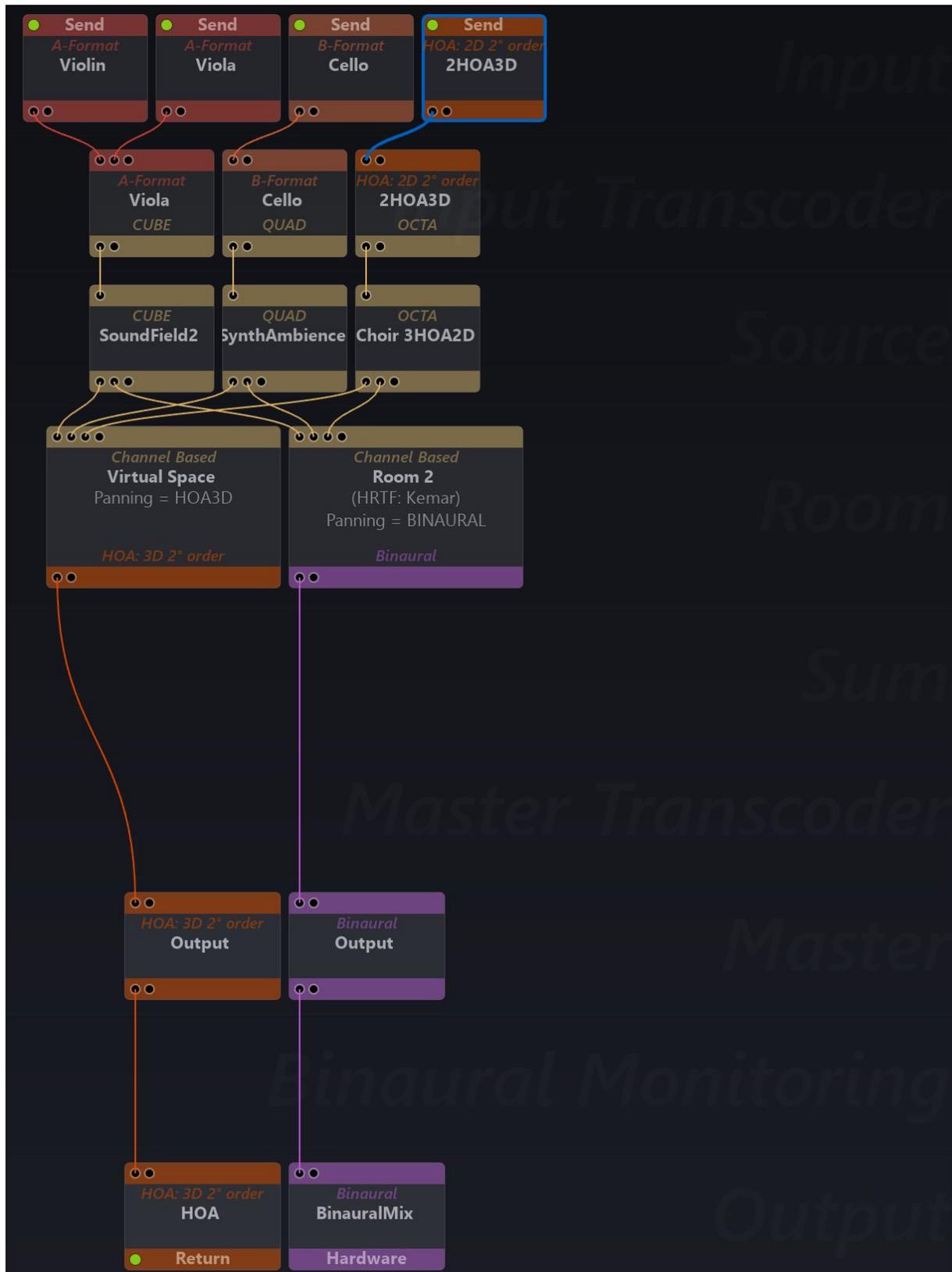


# Room View



- Design a new sound stage acoustic to up-mix into
- Mix using Source Gains, *Barycentric* and *Radiation*
- Mix using *Filters* (on and off axis)
- Mix using different *Reverb Send* sections

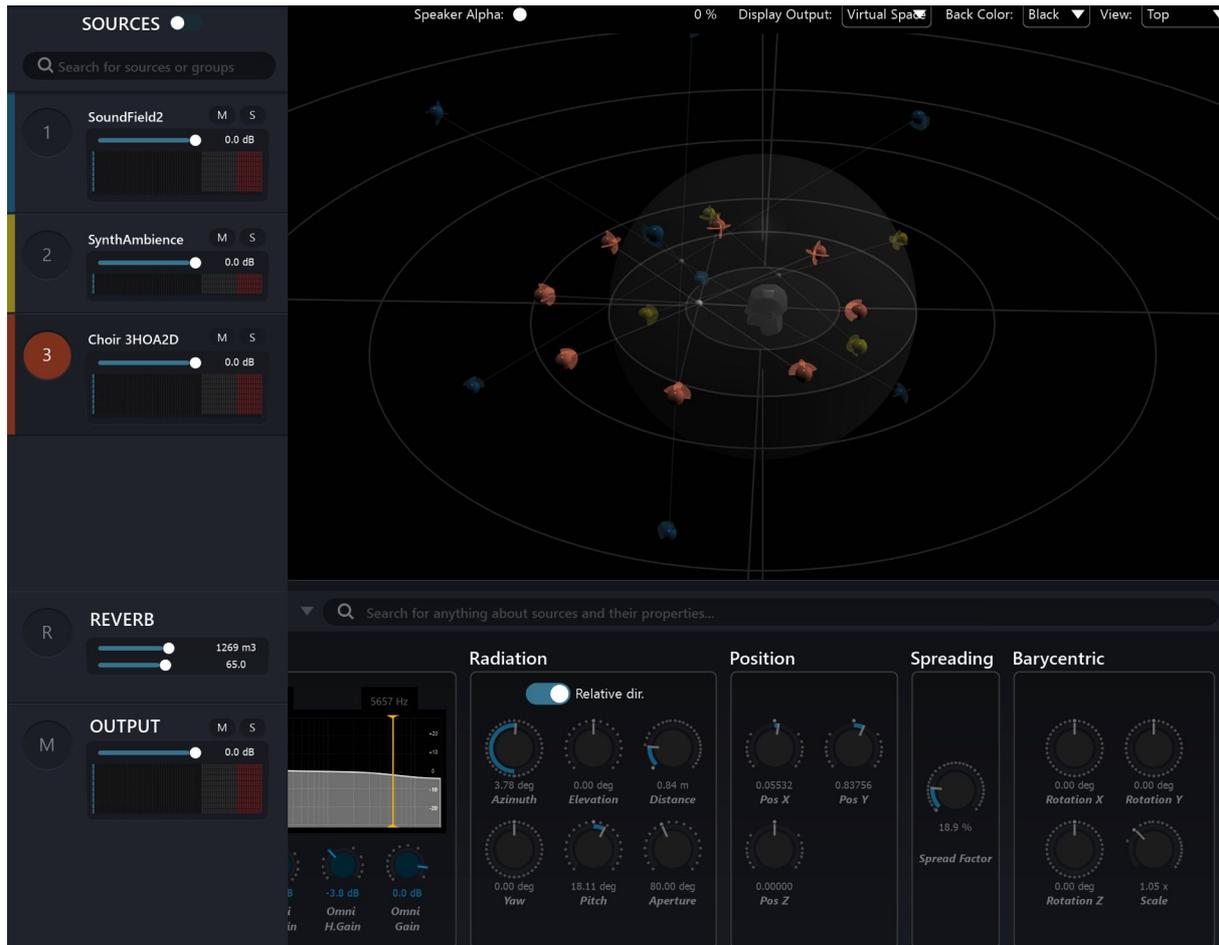
## 11.6 Ambisonic Mixing



## Setup View

- A-Format SoundField microphone sources need input transcoding
- Ambisonic format inputs transcoded (decoded) to virtual configurations
- Suggest CUBE for BFormat3D / Quad for BFormat2D / OCTA for 2nd Order HOA2D

## Room View



- Mix by manipulating virtual channel based Groups in HOA Room using Source Properties
- Record to HOA and/or Binaural
- Monitor to Binaural

# Explore Ambisonic Decoding Methods

The screenshot displays a software interface for audio processing. On the left, a signal flow diagram shows two 'Send Stereo pair1' and 'Send Stereo pair2' blocks at the top. These connect to 'Stereo pair1' and 'Stereo pair2' blocks, which then feed into a 'Channel Based Room 1' block. The 'Room 1' block is configured with 'Panning = HOA2D' and 'HOA: 2D 3rd order'. Its output goes to a 'MasterTranscoder' block, which is also labeled 'HOA: 2D 3rd order' and 'Stereo'. The 'MasterTranscoder' block's output goes to a 'Stereo Output' block, which then feeds into a 'Stereo mon1516' block. The 'mon1516' block is labeled 'Hardware'. On the right, a control panel for the 'MasterTranscoder' block is visible. It includes buttons for 'Disconnect Selected', 'Disconnect Selected Input', and 'connect Selected Output'. The 'NAME' is 'MasterTranscoder'. The 'INPUT LEVEL' section shows a meter with values from -72 to 0. The 'INPUT CONFIGURATION' section includes: 'Stream Type: High Order Ambisonic', 'Dimension: 2D', 'Order: 3', 'Normalization: N2D', 'Sorting: ACN', and 'Yaw Angle: 0°'. The 'HOA DECODING OPTIONS' section includes: 'Method: AllRAD' and 'Type: InPhaseMaxRe'. The 'HOA DECODER CROSSOVER' section shows a slider set to 700 Hz. The 'OUTPUT CONFIGURATION' section includes: 'Stream Type: Channel Based' and 'Speaker Arrang: Stereo'. The 'OUTPUT LEVEL' section shows a meter with values from -72 to 0, and two values: -34.9 and -34.3. A blue arrow points from the 'MasterTranscoder' block in the signal flow to the 'Method' dropdown in the control panel.

## Setup View

Explore how different Ambisonic decoding Methods and Type options actually *sound* in realtime. Here - even just decoding 3rd Order Ambisonics down to basic stereo - many balances of mix separation can be achieved exploring different Ambisonic decoding methods.

## Room View



- Filter some faders to create a multi source interface
- Mix using selected parameters and graphic positioning of source
- Design a High Order Ambisonic reverb that works for well for the HOA decoder you have selected

## 12

# Third Party Integration

In this final section we will cover a few working scenarios integrating SPAT with different setups. Of course, third party systems are subject to many changes, and we cannot cover everything available. Please consider the following as guiding principles which should be applicable in general to other situations not featured here. If you need more specific help, don't hesitate to contact Flux Support and we will try to assist you.

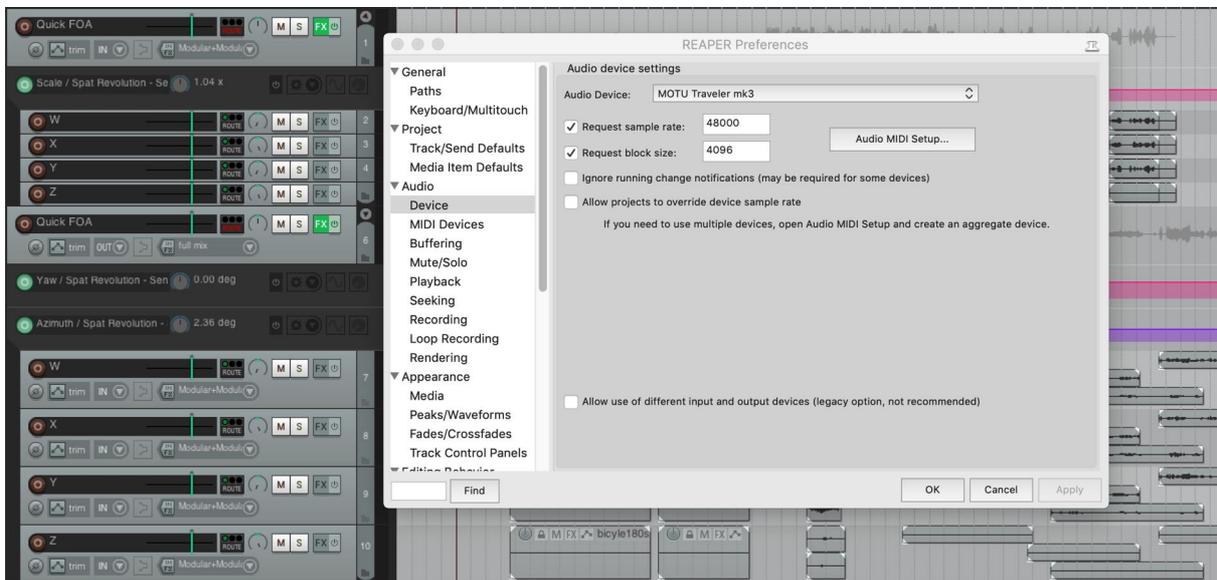
## 12.1 Cockos Reaper DAW

Reaper is a reliable and highly customisable digital audio workstation. It is available to evaluate freely from <https://www.reaper.fm/>. Many composers and designers working in spatial audio use Reaper because it is very good at handling multi-channel interleaved audio (up to 64 channel WAV) and has a straightforward approach to routing. It also features a great automation system which is important for precise composition in Spat and has native OSC support for navigation, transport and more.

Reaper seems to integrate well with the Spat Plug-ins and the Local Audio Path workflow described in section 10.1. If you are using the Local Audio Path, it is also important to taking a look at Appendix B of this guide.

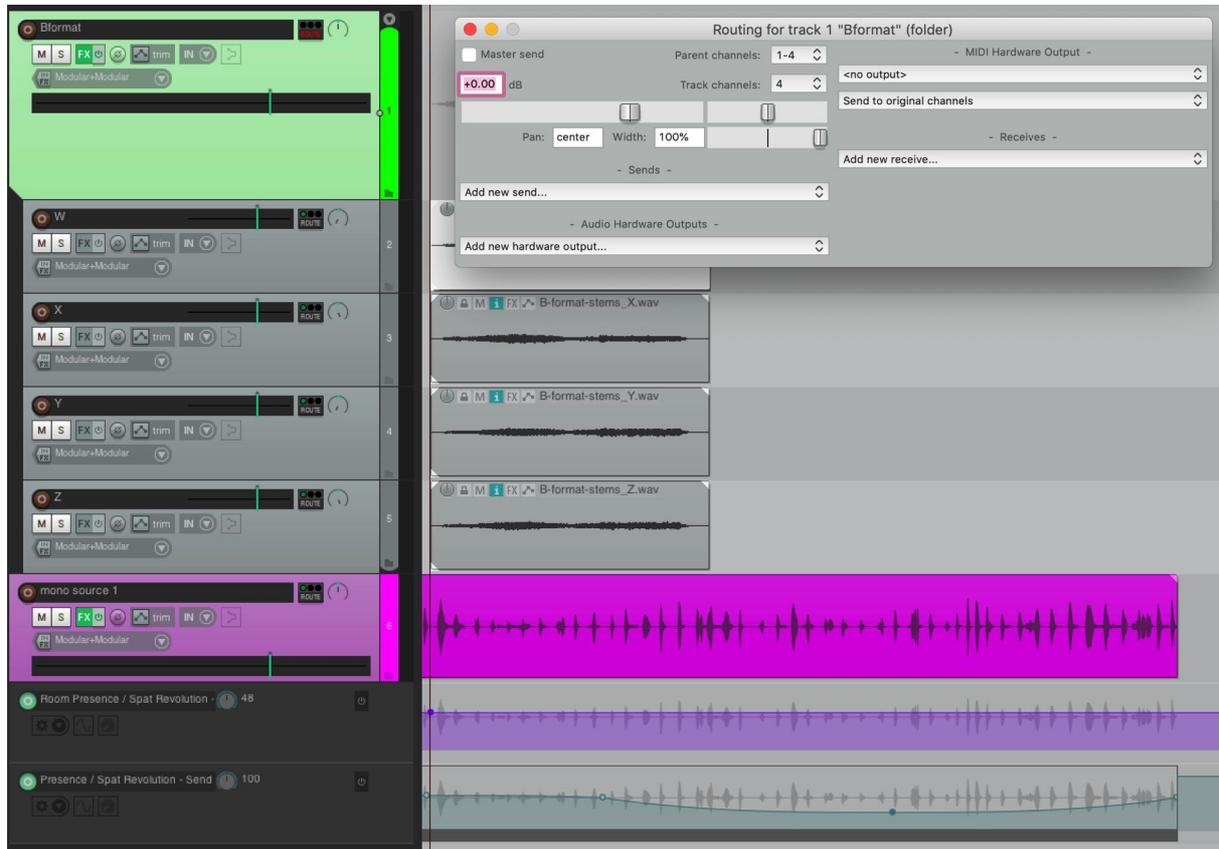
## Setting Up Sync in Reaper

As described in section 7.4, when you are using the Local Audio path, the buffer size and sample rate must be matched in both Spat Revolution and Reaper. In Spat you set this in the preferences and in Reaper in the Audio Device preferences. If they don't match at first, you *may* need write the correct settings to the preferences by quitting and restarting both applications to get the correct green sync status between the apps.



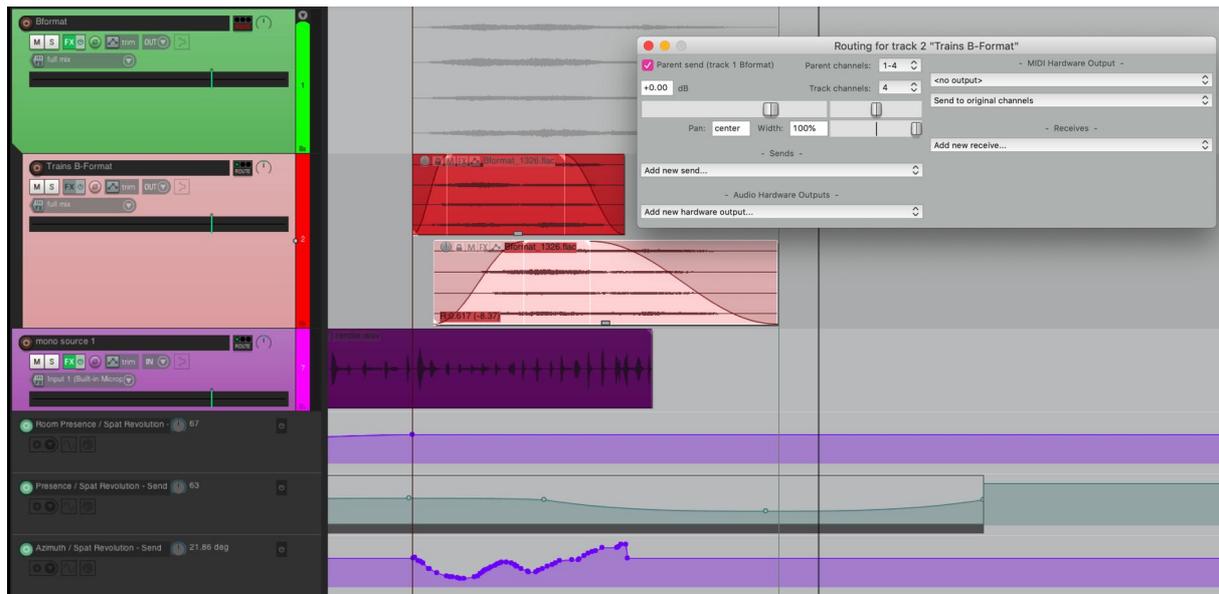
# Setting Up Tracks in Reaper

It is a good idea to work with Track Folder structures in your Arrange.



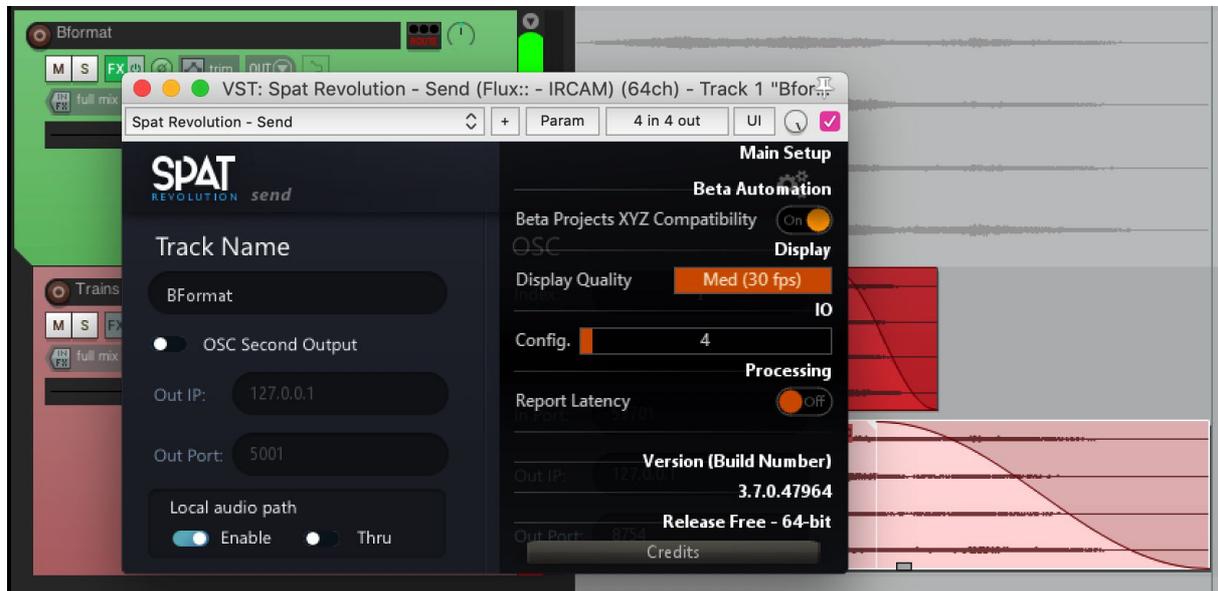
In the screenshot above, the B-Format Master has been set to be a Folder Parent with 4 Track Channels. Reaper channel routing is set on a track by track basis, using the *TrackIO Route* button of each Track. All the Child tracks that route to the Parent can be assigned to one of the four receiving channels on the Parent track. In this example, the W is assigned to Track 1 by setting the track Pan to the left and routing to Parent Channels 1-2. Similarly the X to Track 2 ( setting the track Pan to the right ). The Y Track is assigned to Parent Channels 3-4 and hard panned left and so on.

Alternatively, an interleaved four channel audio file (B-Format audio in this example) can be placed on one Child Track, which has been specified to have 4 channels.



Now different interleaved files in the same format can overlap and be composed on the same Track, and they will be summed to the Folder track, and therefore to same Virtual Source in Spat. All automation for that source should happen in the Envelope Lanes of the Parent Folder track.

## Setting Up Spat SEND in Reaper

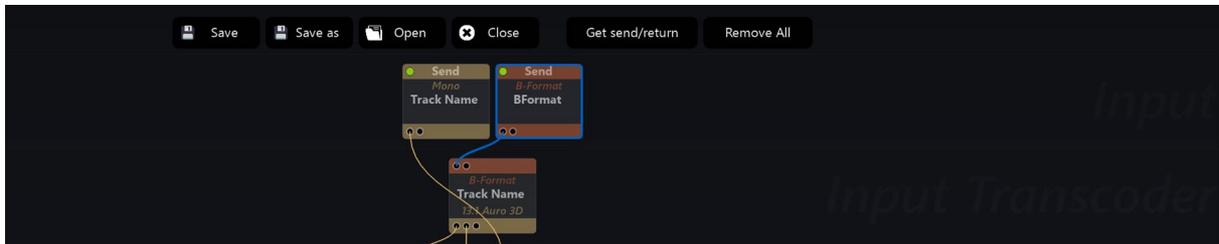


Put the Spat SEND on the Parent Track. This means that as the composition grows, you may have multiple Child tracks sending different audio material to the same virtual source in Spat, through one SEND plug-in on the Parent Folder Track.

This is the least complicated way to build up a large project because then the automation for the spatialisation is managed in one place even though source material may be different at different stages of the composition. Maybe it is not always necessary to have a different virtual Send for every single audio file. The source automation can change in the envelope lanes of the Parent Folder, making it so that one Spat SEND is shared by child tracks with the same format. This will make more sense in practice.

Check the number of channels streamed between plug-in and Spat. Press on the little COG wheel icon at the top corner of the SEND plug-in to open the plug-in setup menu. The Spat *IO Config* should have inherited the Channel count of the Track, which you have specified in the *TrackIO* of the Reaper Track.

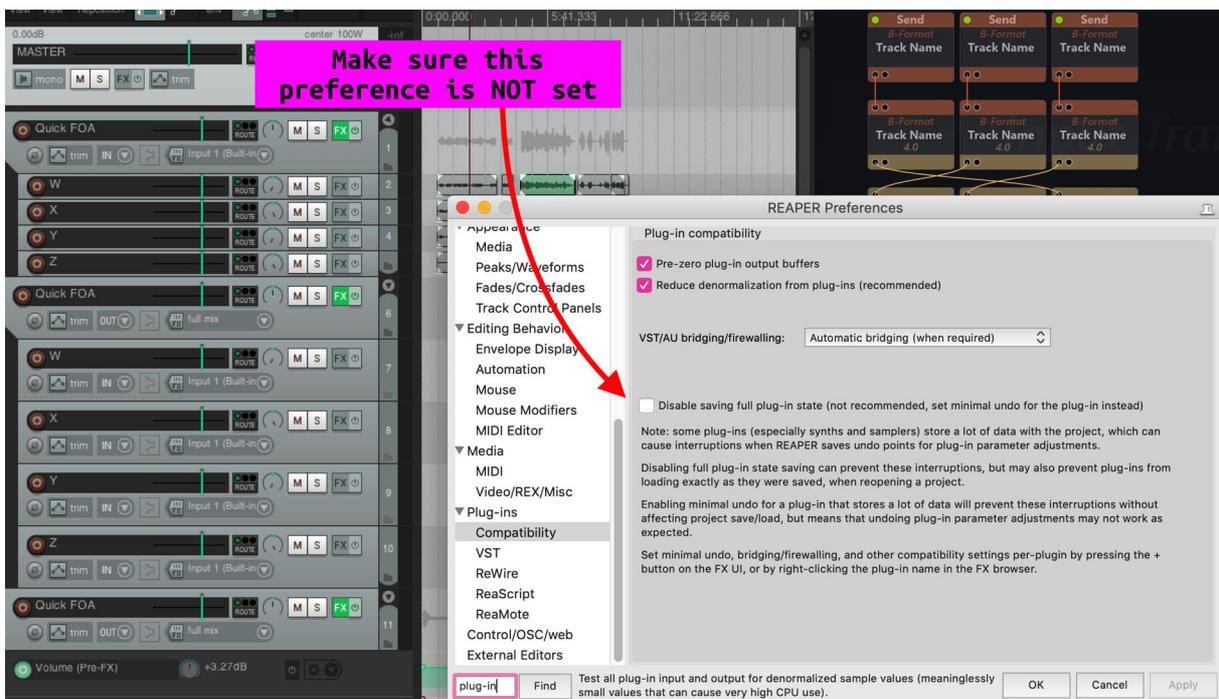
If all is well, then you will see the Send appear as an input at the top of the signal graph in your Spat project. If you do not see it, then try clicking on *Get send/return* to force Spat to look for virtual IO.



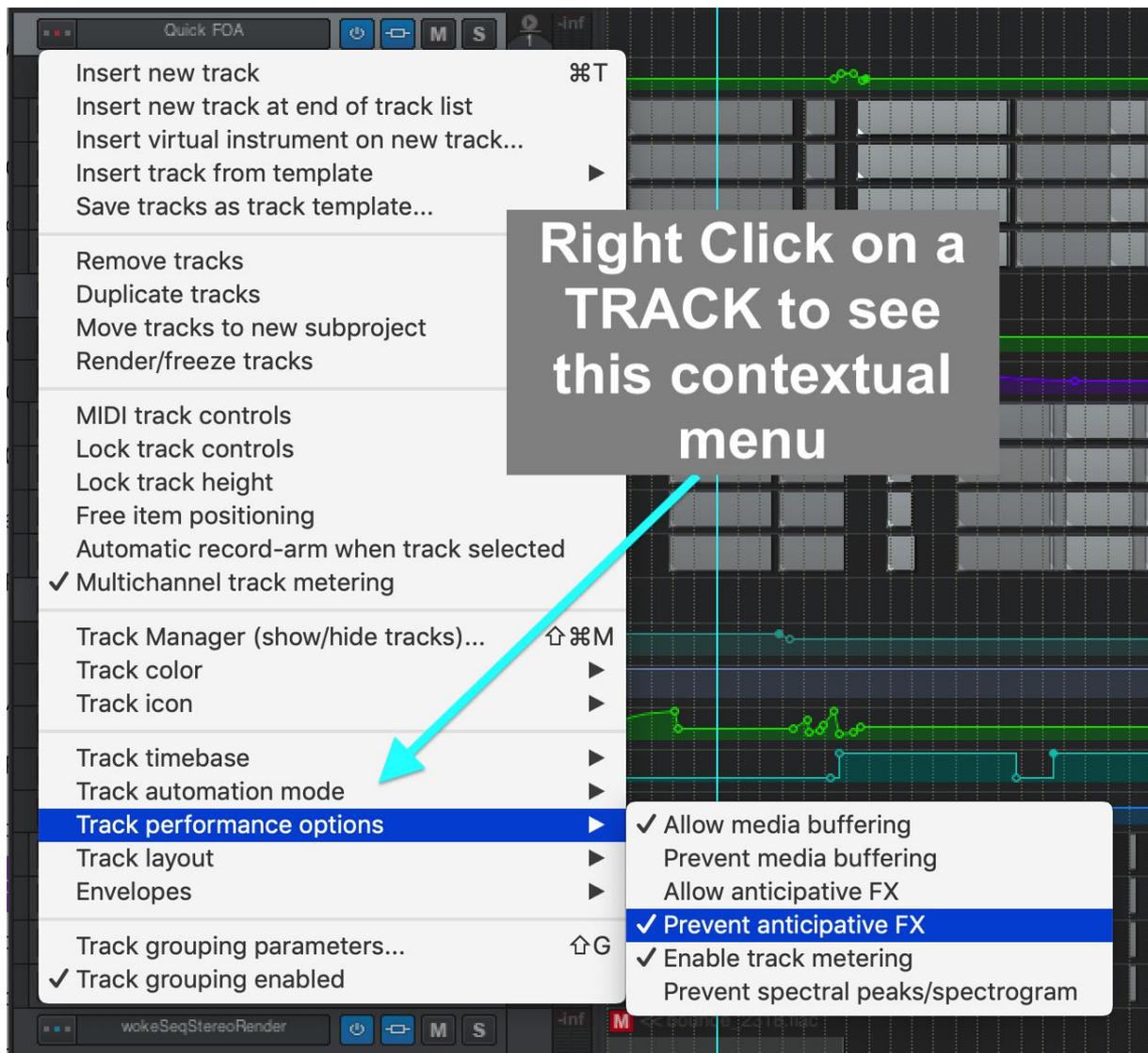
In the screenshot above, the Orange 'BFormat' Send input is the one being routed by the SEND plug on the Parent Track we have set up. At first it will appear in Spat as a Channel Based input by default - in this case it appeared as a 4.0 QUAD. The B-Format Input Stream Type was assigned manually (see section 6.4).

There are a couple of things to watch out for which cause inconsistent behaviour;

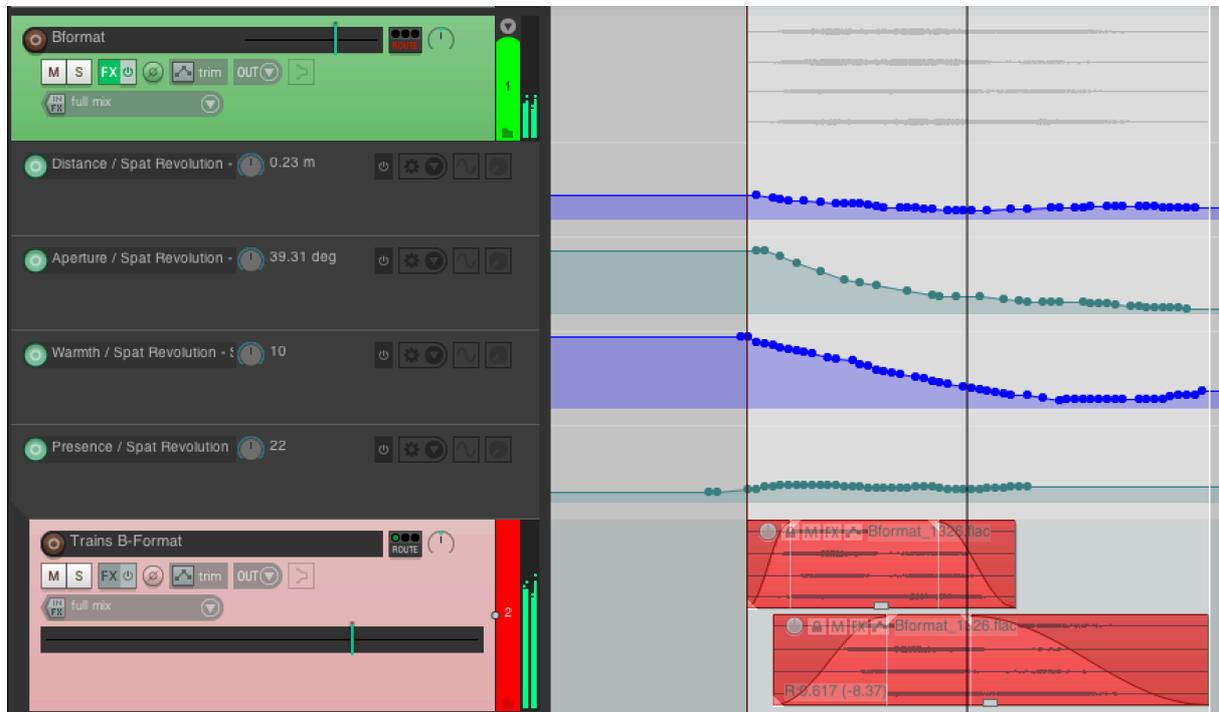
### ⚠ **Full Plug-in State Save is set in Plug-in Compatibility Preferences**



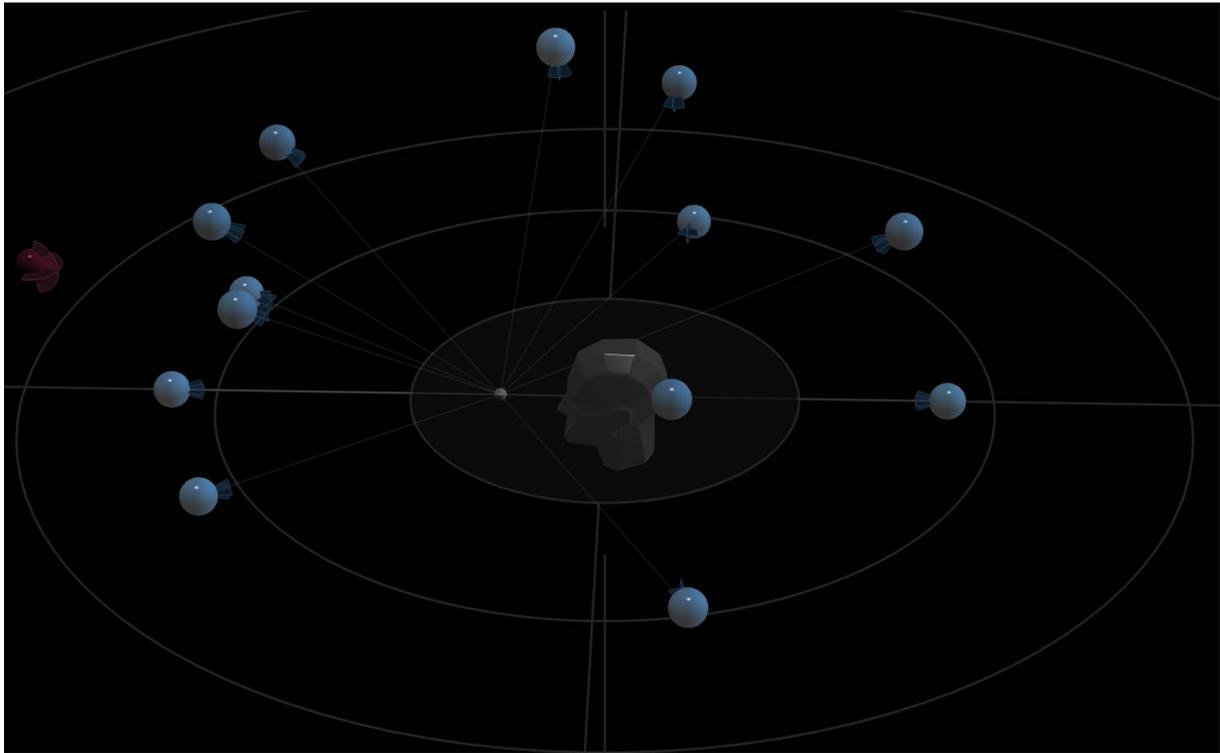
△ **"Prevent Anticipative FX" should be enabled on tracks where SEND / RETURN are inserted**



## Spat Source Automation in Reaper



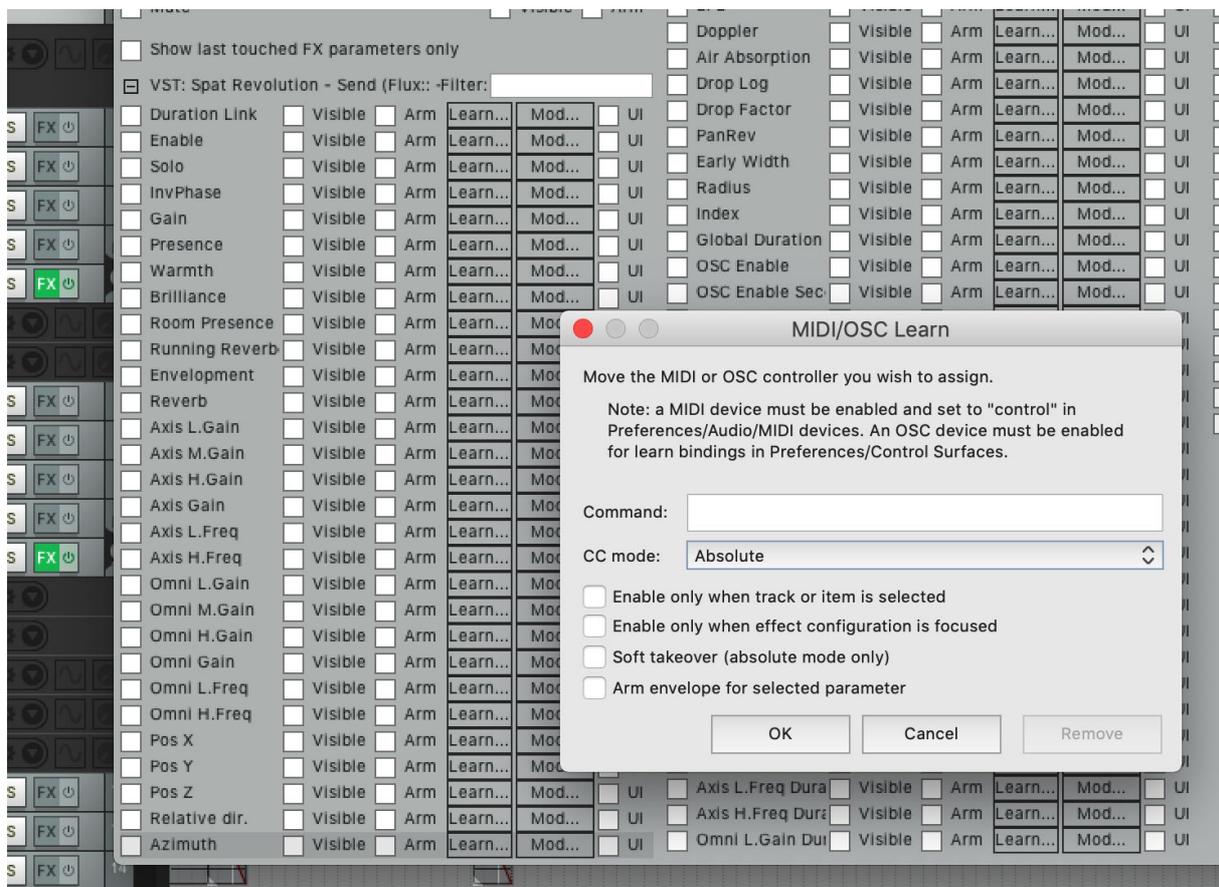
Now it is simply a case of adding some Envelope lanes for parameters you wish to automate from the DAW timeline. Here the B-Format source is being introduced from a distance automating *Aperture*, *Warmth* and *Distance*.



In the Spat HOA Room, we see the B-Format source (transcoded into an *Auro3D virtual speaker configuration* in this example) fly gradually into the scene with the whole sound field transforming slowly as it comes over.

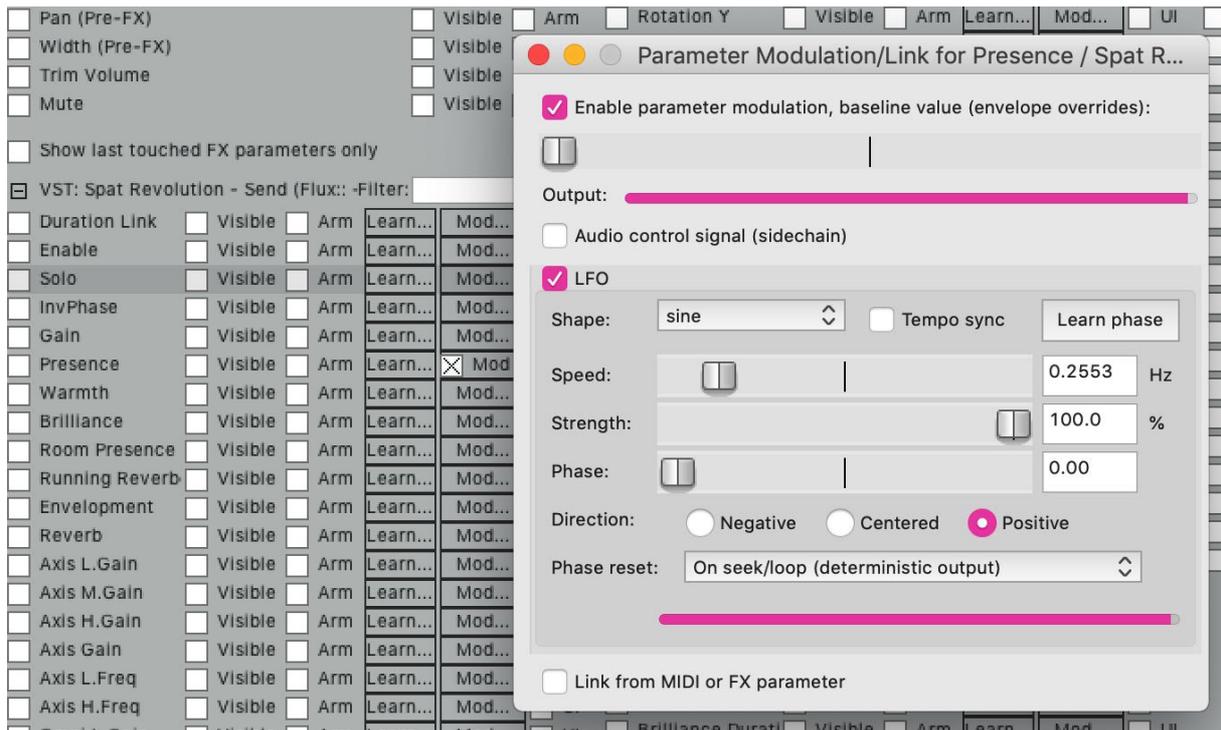
## Setting Up Controllers in Reaper

In Reaper it is easy to map any MIDI controllers (including 14-bit) to a Plug-In parameter. This is a great way to control source properties like *Azimuth* or *Yaw* using external MIDI controllers, so you could control some of the sources live by hand during a performance.



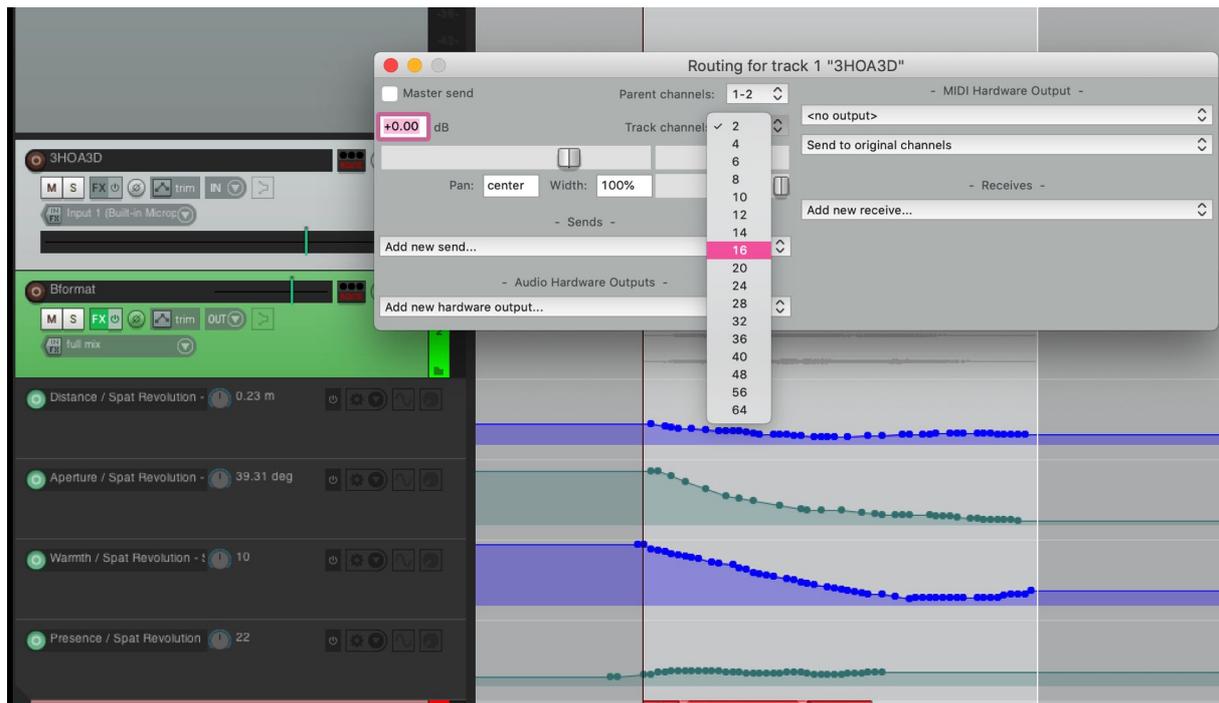
In the Automation parameter list for Spat SEND just click on the LEARN... button to manually assign an external controller that you have set up in the Reaper Controller preferences.

Also from this page by clicking on the MOD... button of an automatable parameter it is possible to animate sources with independent LFOs that run all the time in the background, quick and dirty way to spatialise live inputs sources with autopan type effects for example.

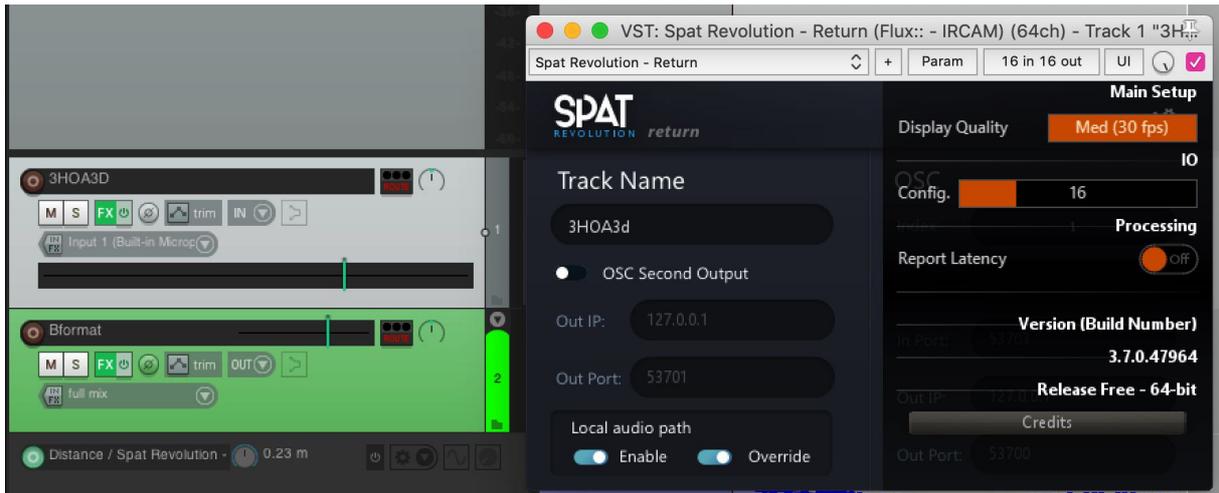


## Setting up Spat Return in Reaper

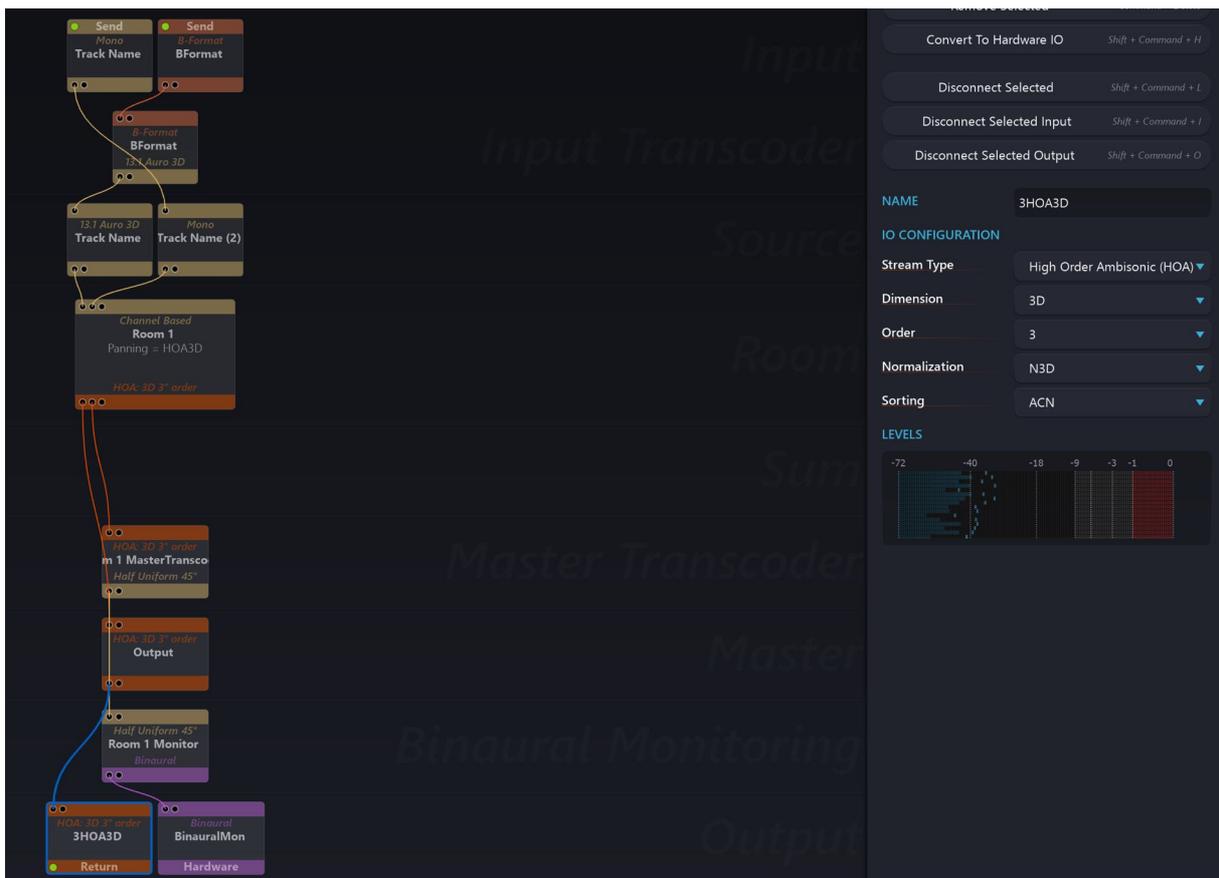
Now we want to render that scene from the Room to disk as an interleaved 3rd Order HOA, for example. From the formula in section 6.45, we can calculate that 3HOA3D needs 16 channels.



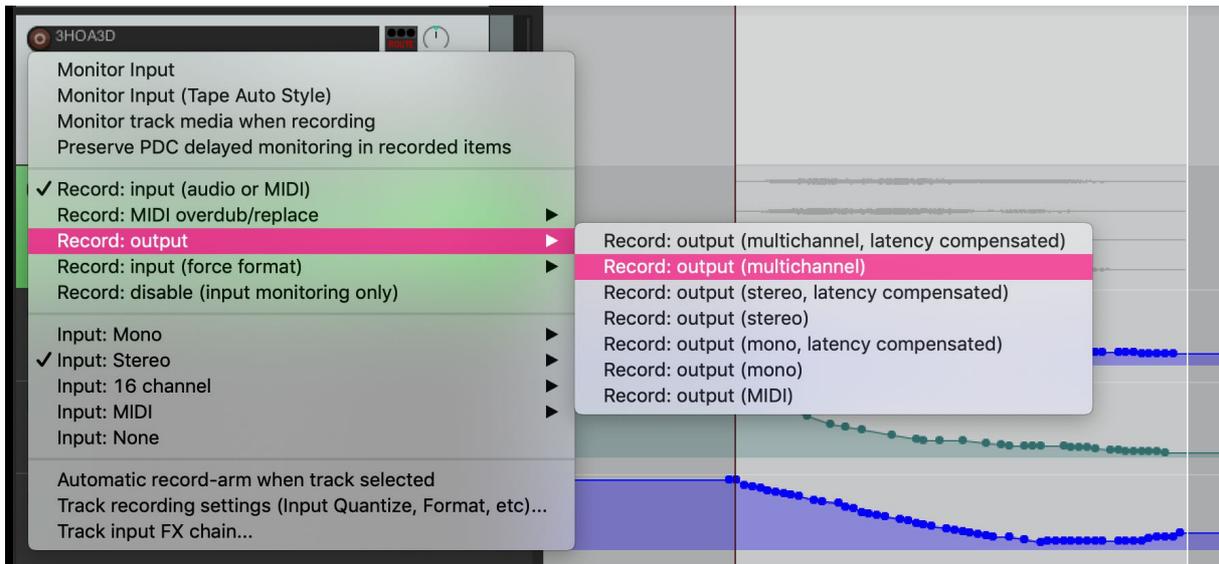
First of all set up a Track in Reaper that can handle 16 channels at once. Then add a Spat RETURN plug-in on that track. It should automatically inherit the channel count, if not do it manually using the IO config of the plug-in, available from the little cog wheel in the top corner of Spat RETURN.



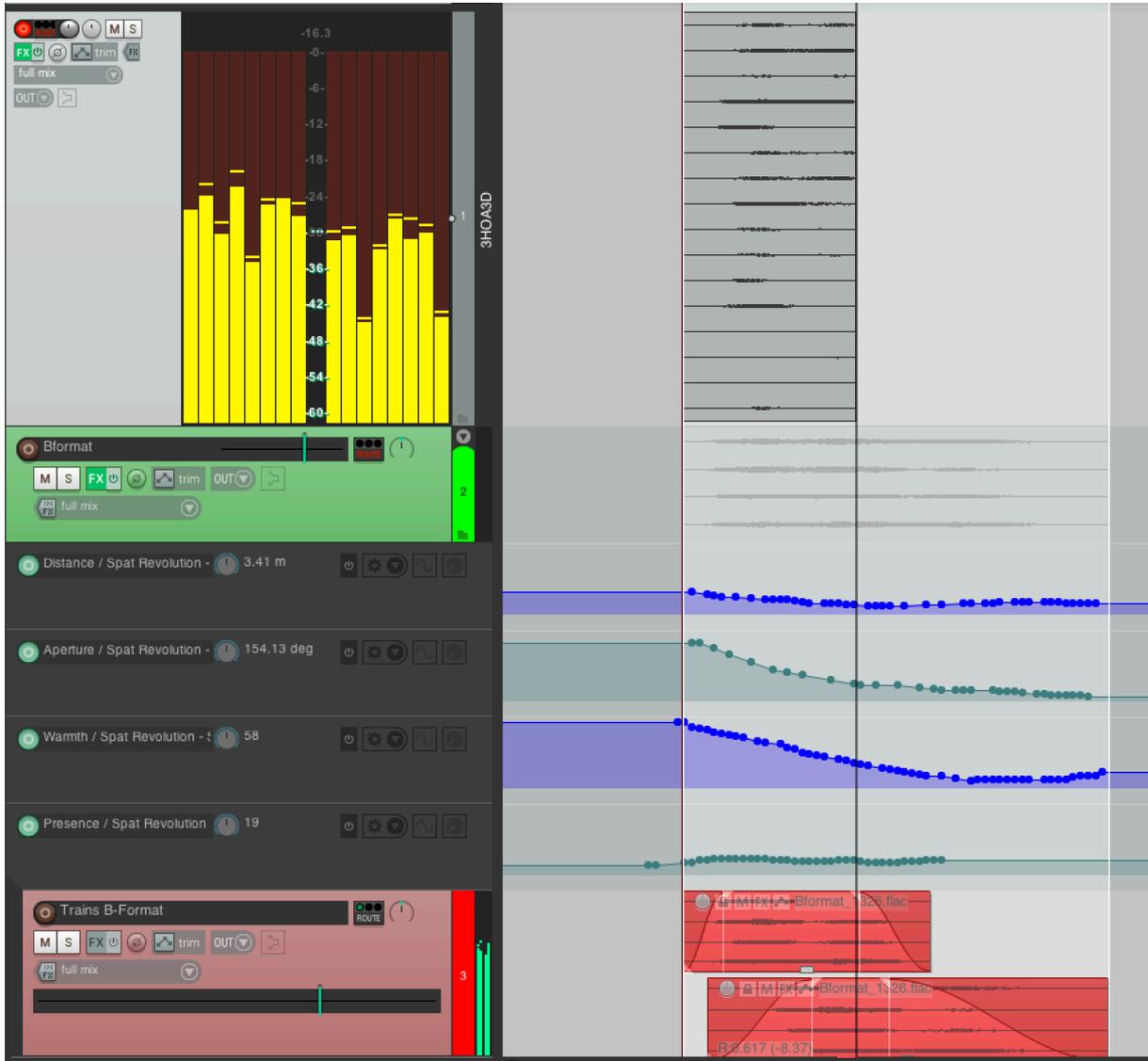
Enable the Local Audio Path and you should see a Return output module appear in the Spat signal graph. Connect it to the 3HOA3D stream output from the Room (or Rooms in a mixer/transcoder) and it should inherit the format.



Now it is necessary to make Reaper record the software OUTPUT of the audio track - the audio coming into the Track is a virtual audio path through the Spat RETURN plug-in so it will *not* appear at the *Inputs* list of the Track. Right click on the record arm button on the track, and the Track record contextual menu will appear.



Arm the track to record, press play and render the scene to a 3HOA3D interleaved 16 channel WAV file (avoid using FLAC for higher than 8 channels)



The same process can be followed to render any Channel Based or other stream type from Spat to disk for further composition, mastering or final delivery.

## 12.2 Bitwig Studio

Bitwig Studio is a DAW designed in Berlin with a live hands on philosophy to it. Along with its highly animated and intuitive graphic interface, it offers well designed clip-based and timeline arrangement paradigms for composing. The Bitwig designers have included a complete suite of powerful and great sounding native effects and digital instruments, with many performance and modulation features - for many users, its the parameter modulators and their well designed routing system that makes it compelling to create music and sound design in BitWig.



## Setting Up Sync in BitWig

As described in section 7.4, when you are using the Local Audio path, the buffer size and sample rate must be matched in both Spat Revolution and Bitwig Studio. In Spat you do this in the preferences and in Bitwig you do in the audio engine settings. If they don't match at first, you *may* need write the correct settings to the preferences by quitting and restarting both applications to get the correct green sync status between the apps.

CRISTIAN   **SETTINGS**   PACKAGES   HELP   ×

---

User Interface

Behavior

**Audio**

Recording

Controllers

Synchronization

Locations

Plug-ins

Shortcuts

### Audio System

Driver model: Core Audio

Input Device: None

Output Device: Flux:: Aggregate Device

---

### Flux:: Aggregate Device

Sample rate: 48000 Hz

Block size: auto 1024 samples / 21.33 ms

Recording offset: 0 samples / 0.00 ms

### Output busses

	Role	1	2	
Speakers	Speakers	✓	✓	×

Add Mono   Add Stereo   Add Speakers   Add Headphones

Preferences

### Hardware IO

Devices: Flux:: Aggregate Device

Sampling Rate: 48000 Hz

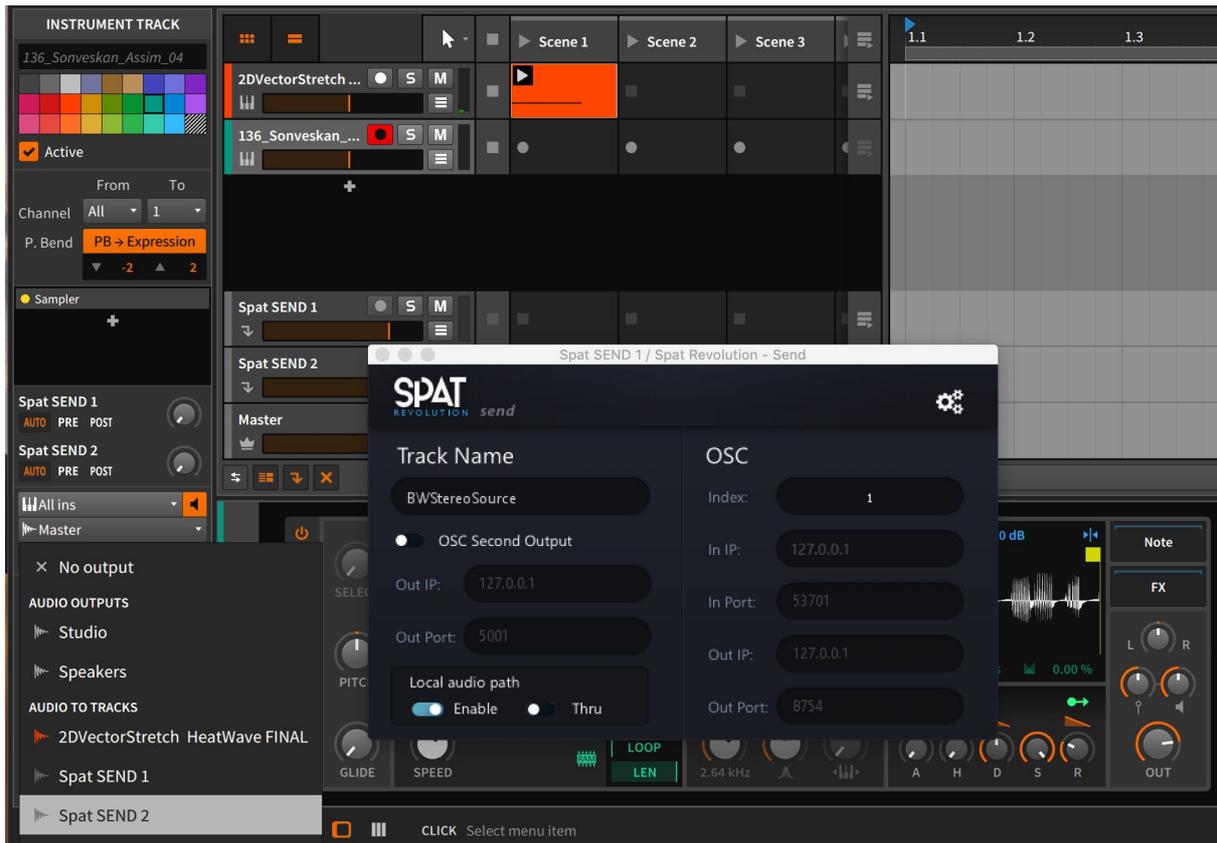
External SamplingRate

Block size: 2048

Show settings...

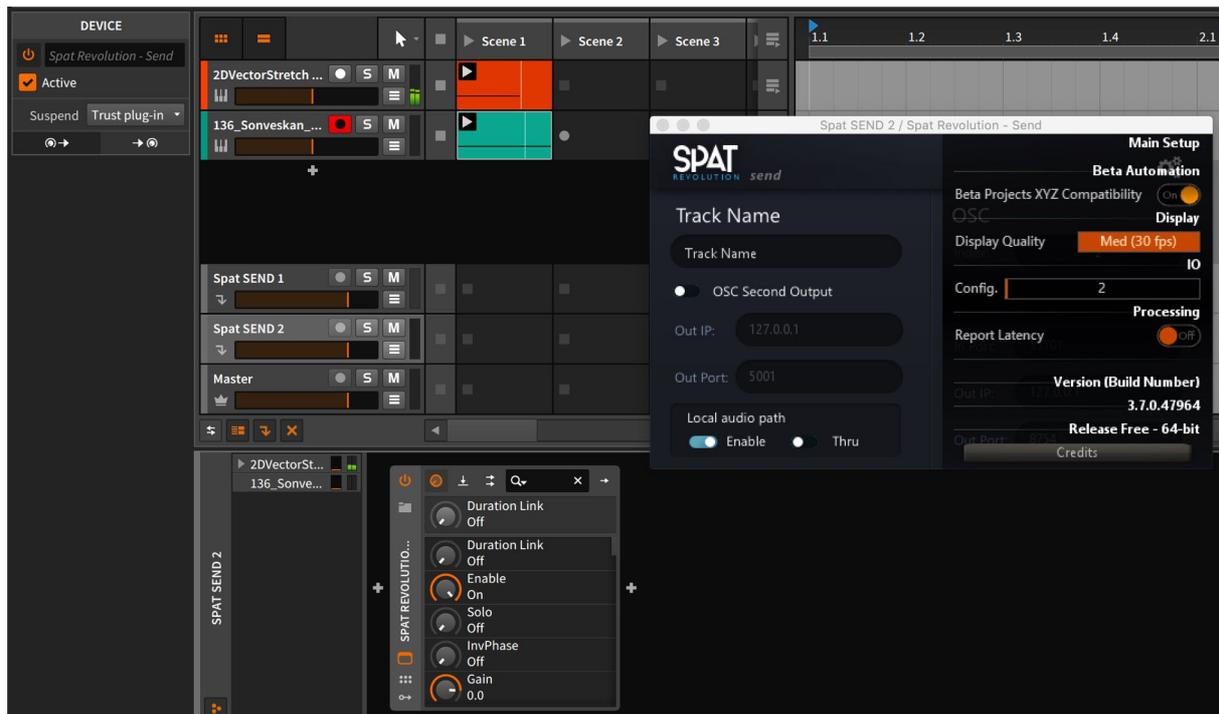
## Setting Up Tracks in Bitwig

One good way to work with Bitwig and Spat together, is to set Bitwig tracks to output their audio to *Effect Track* types - they are like Aux busses in other software - you do that routing from an audio track output assignment settings.



## Setting Up Spat SEND in BitWig

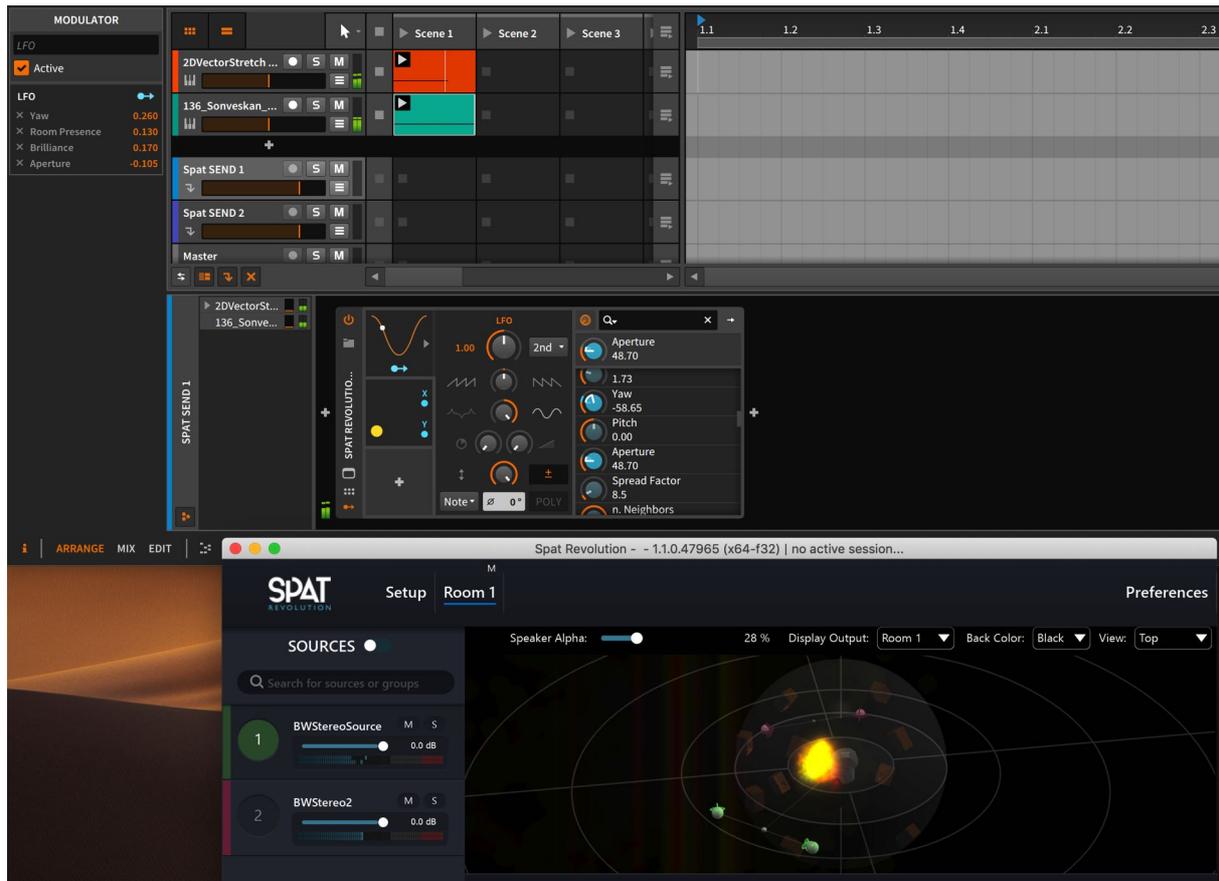
Put the Spat SEND plug-ins on individual *Effect Tracks* , enable the local audio path with **THRU** set to off, so all audio streams are rendered to output in Spat.



Like in all Local Path Audio workflows, you should see Spat SEND inputs appearing in the Spat Environment Setup graph which relate directly to the plug-ins hosted in the other software environment, reflecting their TrackName and channel count.

## Spat Source Automation in BitWig

Now the fun starts - on the Bitwig Send tracks, which host the Spat SEND plugins, you will see all the parameters of Spat sources available as dials. Use the Bitwig ( + ) to open the Device Parameter Modulators and assign the many and varied modulation sources to control Azimuth, Distance or other Source Parameters.



## Setting Up Controllers in BitWig

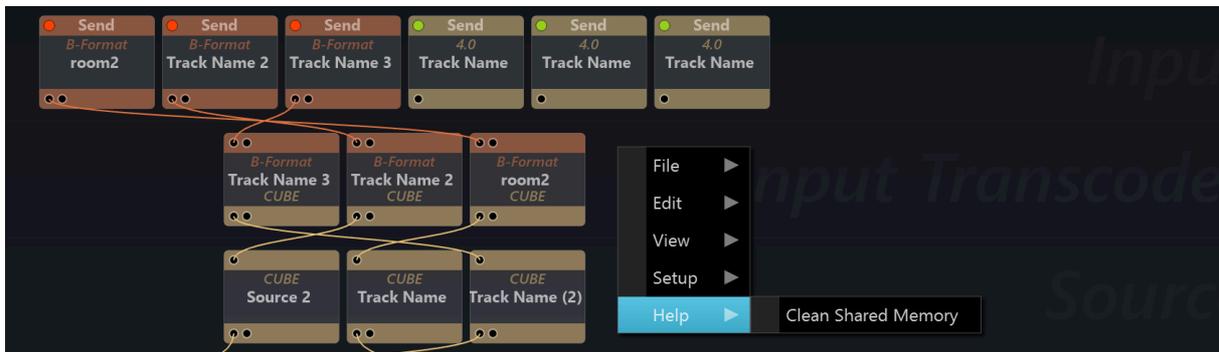
BitWig has comprehensive support for many popular MIDI controller surfaces. These can be very easily mapped to Spat SEND parameters for hands on control of virtual objects and rooms.

## Clearing Shared Memory

Currently with Spat Revolution v1.1 some users have experienced an issue where Spat SEND and RETURN plug-ins are not cleared from RAM after using certain third-party hosts. BitWig is one of those.

If this happens, Spat SEND and RETURN modules will appear in the Spat setup graph editor, when there is no host software running in the background. It can cause problems, when a host with plugins is launched and more SEND and RETURN plug-ins appear to be doubled.

The workaround, if this is happening with your particular 3rd party software, is to invoke a special debug action in the Spat setup editor. It is called *Clean Shared Memory*. It is available by right-clicking anywhere in the background of the signal graph editor. A pop-up menu will appear with various options and shortcuts. Scroll to the bottom and choose *Help/Clean Shared Memory*



If this command is executed, Spat **and** the plug-in host will then need to be restarted.

# 12.3 ProTools

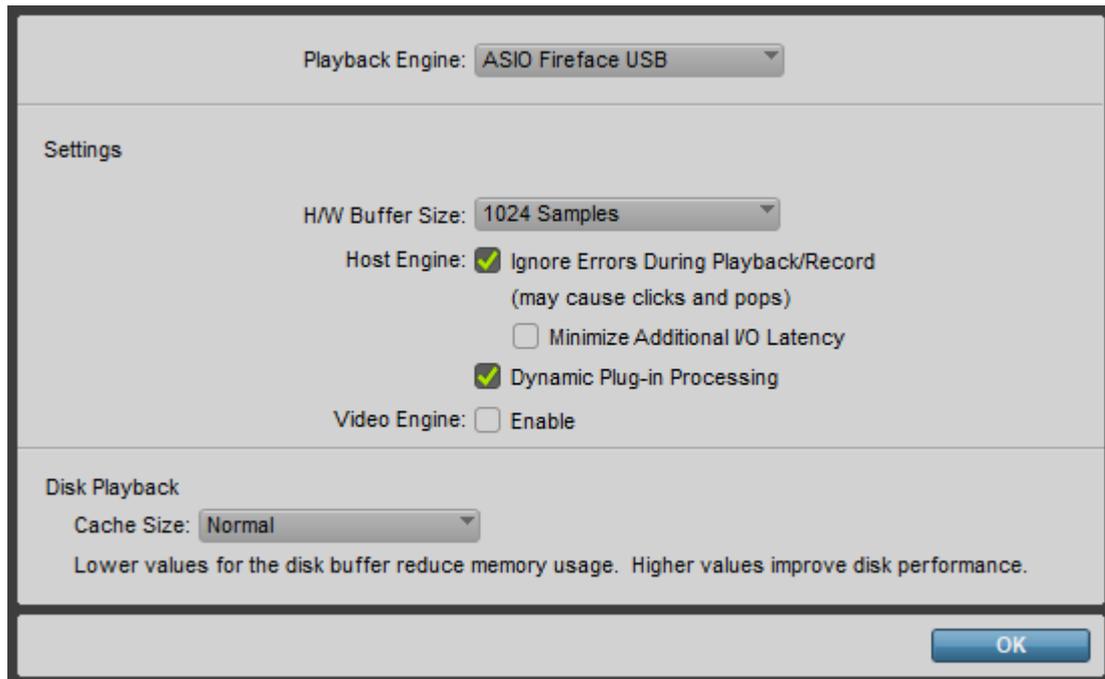
Spat Revolution can use the AAX version of its plug-in suite to establish connection between Avid ProTools and the Spat Revolution environment. Here is a rough guide to getting things running smoothly when using the local audio path and ProTools.



## Setting Up Sync in ProTools

We have found that sync issues are best avoided when the playback engine buffer size is set to 1024 samples, in both ProTools and Spat Revolution.

In ProTools, go to *Setup/Playback Engine* then set H/W Buffer Size to 1024 samples.

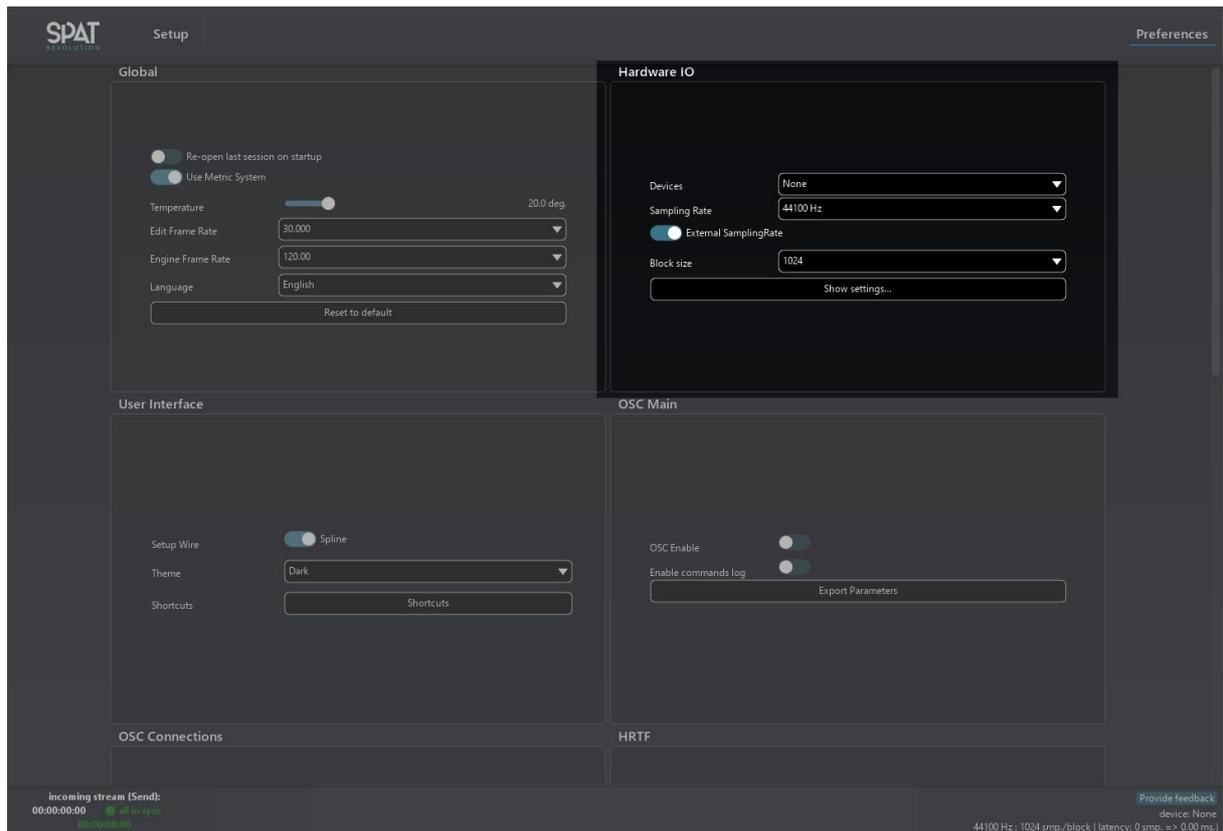


In Spat, go to *Preferences/Hardware IO*, then set :

Device : None

External Sampling Rate : Enabled

Block Size: 1024



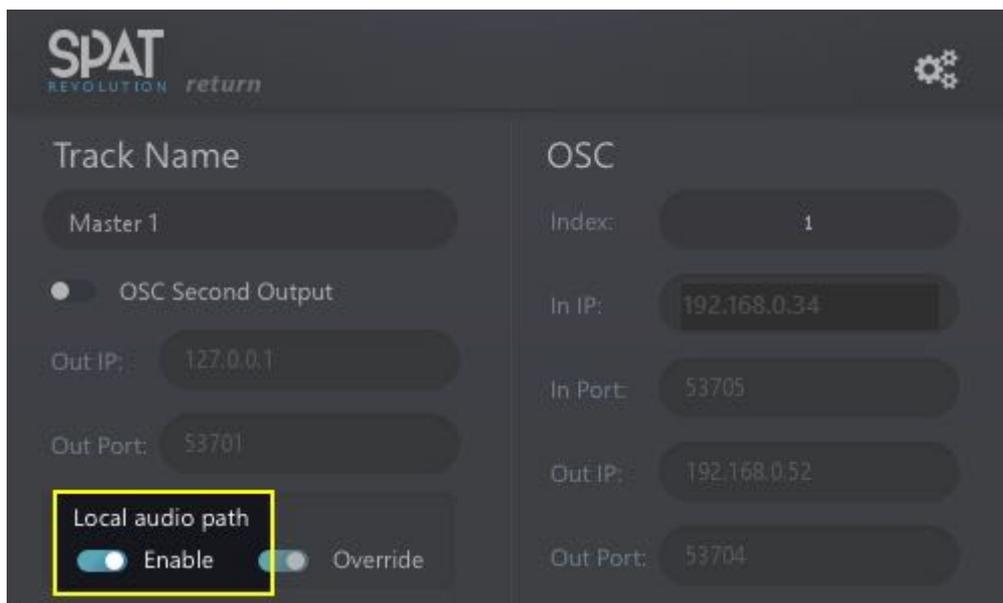
Please note that you *may* have to close and restart Spat to ensure that everything is properly saved and applied.

## Setting Up Tracks in ProTools

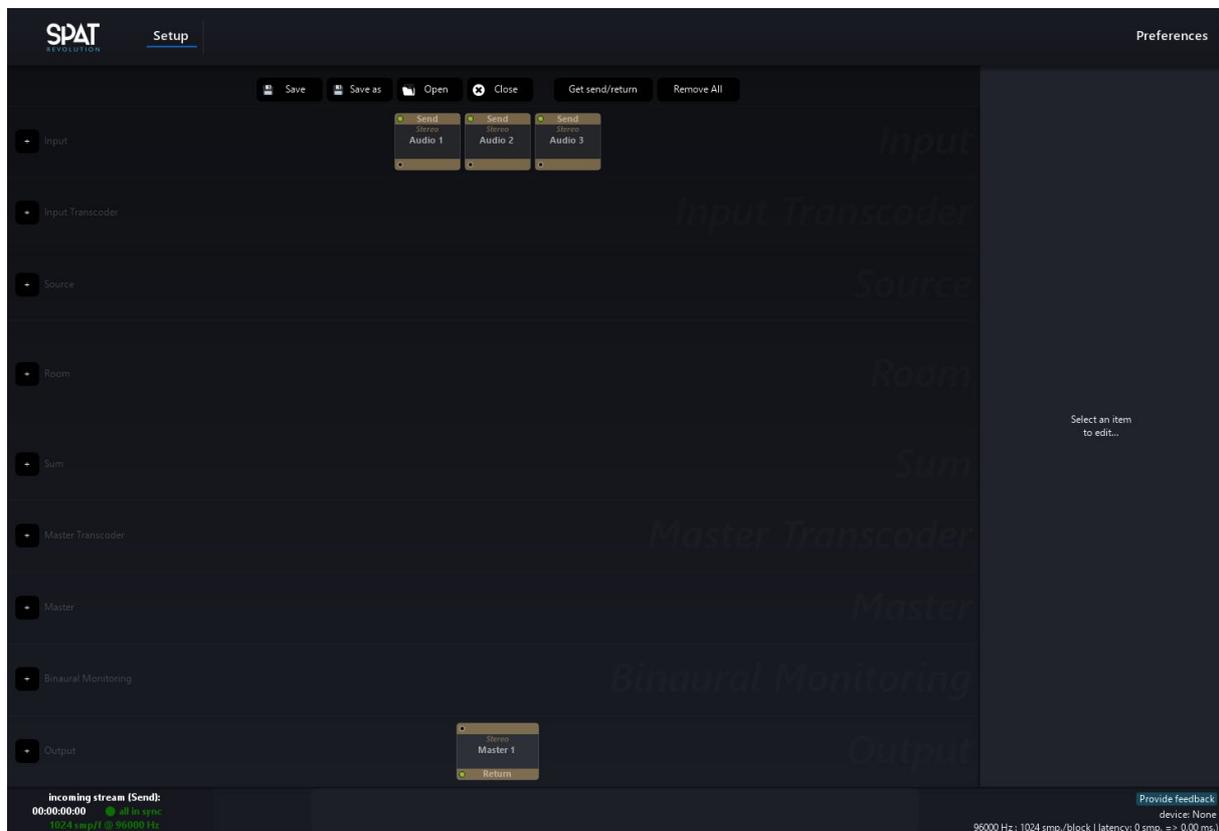
As a first step, please give a look at Appendix B. It describes the best practices for routing in most DAWs to avoid sync issues with Spat Revolution.

To setup a Protools session to work with Spat Revolution using the Local Audio Path mode, is simple. Just add Send and Return plugins and enable the 'Local Audio Path' option for each.

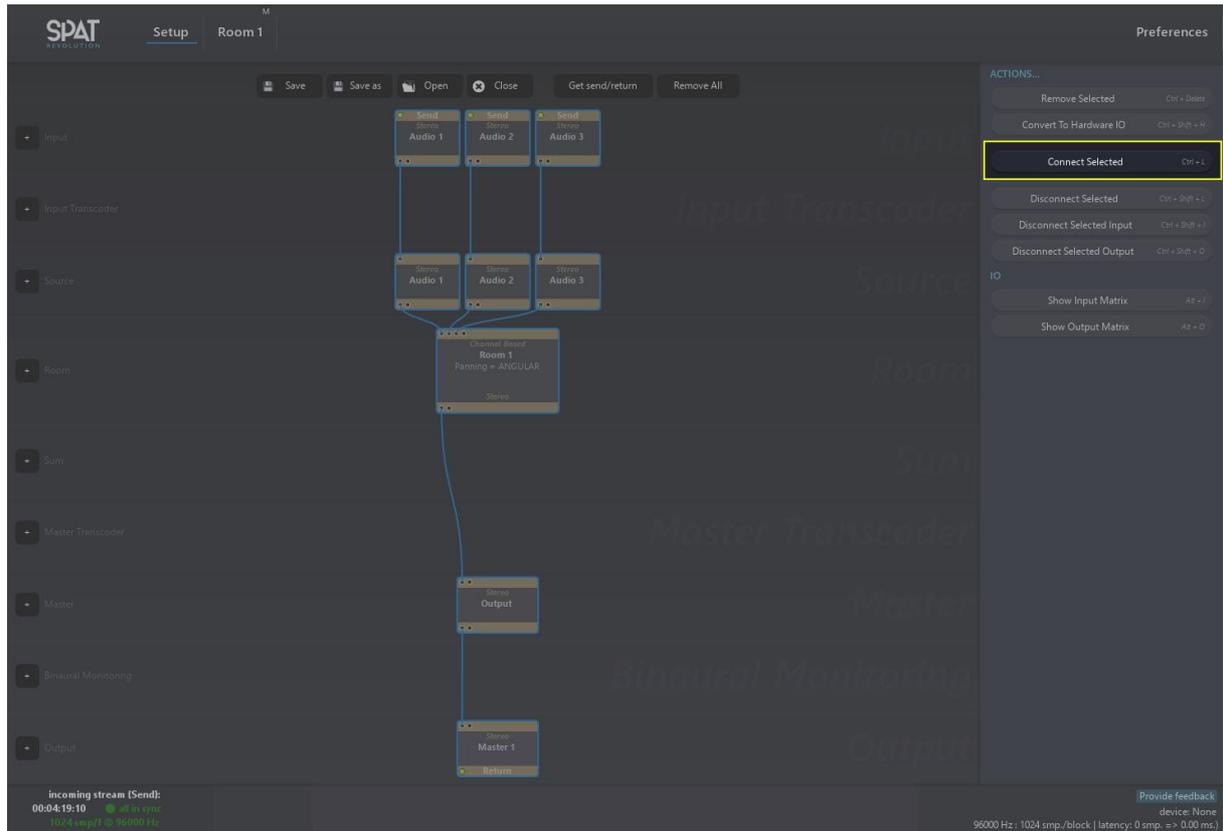
Once you flick on the *Enable* switch, matching Send inputs should show up at the top of the Spat Revolution setup page, and Return outputs at the bottom of the graph.



**⚠ Routes to the Plug Ins will appear in the Signal Flow Graph only when Enable is switched on**



Select all the objects in the 'empty' signal graph using a drag select or Command-click selection, then choose the 'connect selected' action. A default graph will be completed automatically.



Now you should be able to send and receive the spatialised audio streams via Pro-Tools mixer channel strips.

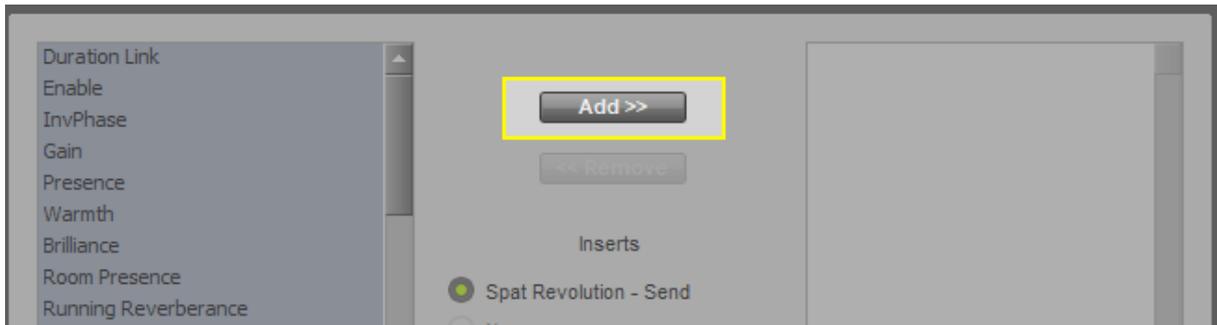


Please note the use of a 'Dummy' bus track, to make the audio tracks easier to set-up. The idea is that you route audio tracks to the 'Dummy' bus using aux send and insert a 'send' plugin with local audio path option enabled.

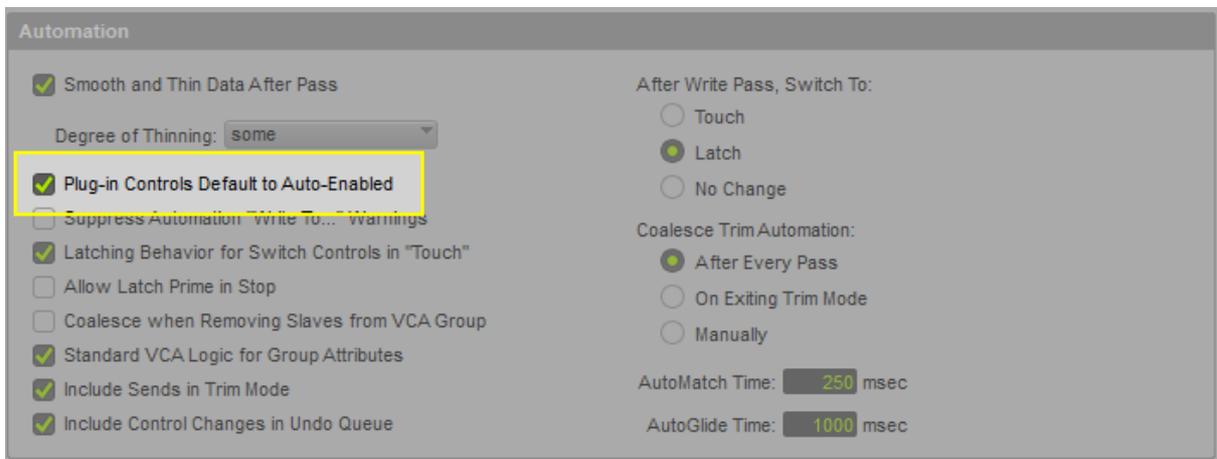
This 'Dummy' bus track has 2 aux send to route the signal to both the 'return' tracks. Simply put an audio file in the send tracks and connect the input and outputs objects in Spat Revolution.

## Spat Source Automation in ProTools

To automate variable parameters in the Spat engine from ProTools automation lanes, it is necessary to activate the parameters for each plugin instance first.



To do this one by one could be laborious. Save some time and clicking by enabling the '*Default to Auto-Enabled*' option in ProTools. When inserting a plugin, all its parameters will then be available for automation automatically.

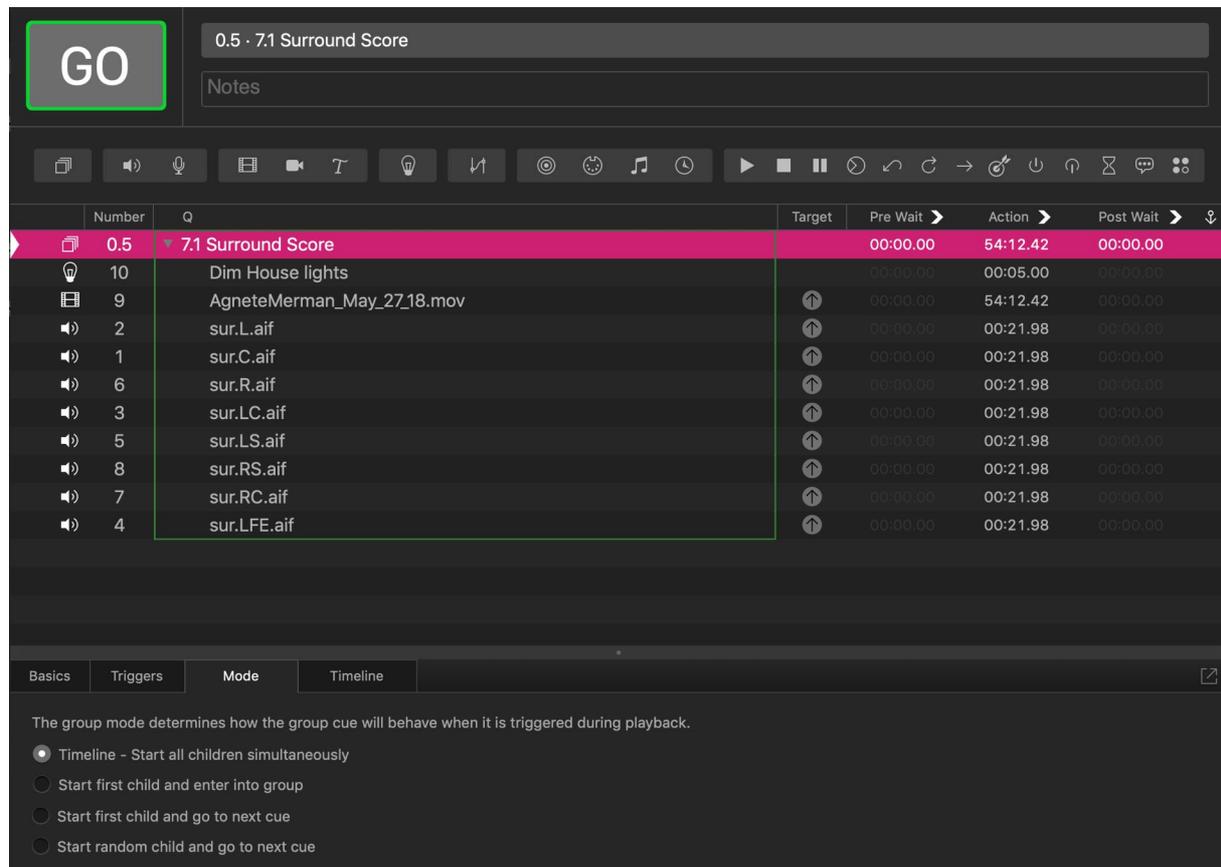


## 12.4 Figure53 QLab

Rock solid and fully featured show control program, based under a 'cue' based paradigm. QLab is great for manually sequencing multiple media and show type events, running video, sound, lights and virtually any type of control scripts through its easy to use interface. Some of the more advanced control features in QLab make the most of its native OSC network integration - which is why it becomes easy to integrate with Spat Revolution.

One of the great features of QLab is that it can handle multi-channel audio stems - at the moment, it is only possible to handle multichannel Send and Returns two channels at a time. The Cue Outputs in WLab will allow to route the pairs of channels to the correct destination. Despite this little workaround needed for multi-channel operation, QLab is still without doubt the most straightforward and reliable way to synchronise a video screening with a pre-rendered multichannel surround soundtrack made in Spat. We will take a look at this simple example set up now.

# Film with Multichannel Audio



If you are making a screening for a client or an audience and do not want to go down the Dolby DCP route, then this is a simple way to do it in QLab. All the stems were rendered to disk from Spat Revolution and are now played perfectly together with the speaker routing handled by the QLab routing system.

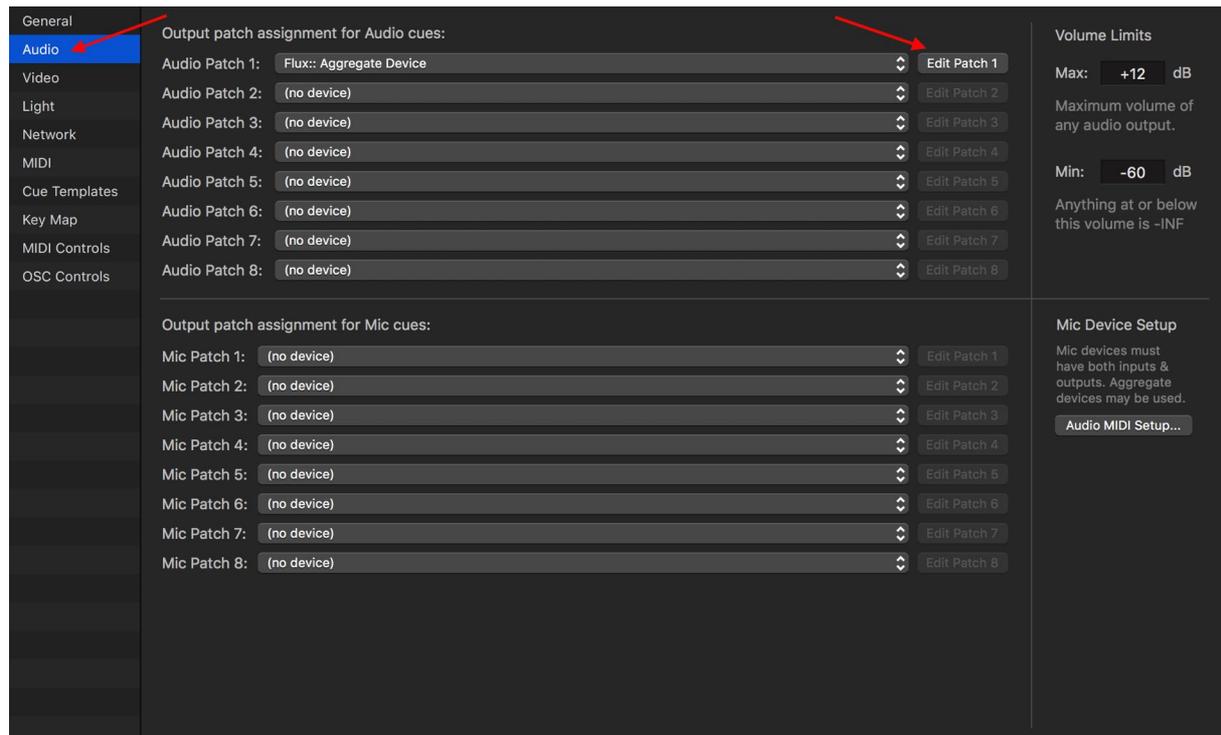
## Local Audio Path in QLab

The Local Audio Path integration between QLab and Spat Revolution is possible using Spat send and Spat Return in AU Plug-in format. In this case both applications are running on the same local computer and audio is being routed to and from QLab and Spat Revolution. In this setup, QLab remains the audio software application managing the inputs and outputs to your core audio device.

The ability to insert the Spat Send and Spat Return Plug-in to the Output patch assignment for Audio cues in Workspace Settings (Command + ,) allows to send

Cue Outputs to Spat Revolution software Inputs in Single (1) or Dual (2) channel arrangements. A total mix of 64 Cue Outputs are available for this thus providing 64 single channel inputs, 32 dual channel inputs or a mix of both.

To do this, first open your QLab project with leaving Spat Revolution application close and go to the Workspace Settings in QLab. Click the Audio Section then choose and Audio Patch to Edit by clicking on the Edit Patch section.

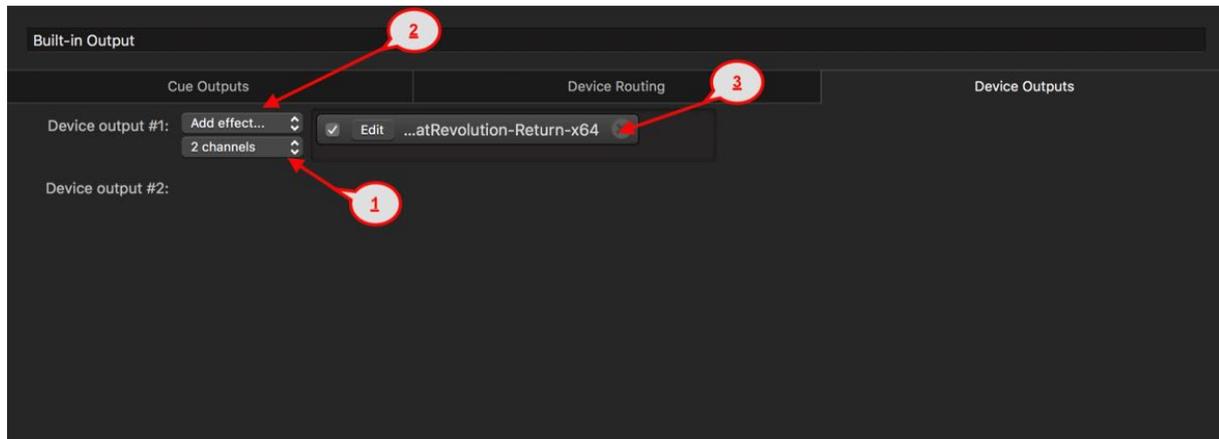


Click the Add effect, Scroll to the Flux category and Choose SpatRevolution-Send-x64. The Spat Revolution Send Plug-In User interface will appear with the default Local audio path disable. Make sure to initialise the audio and automation path by clicking on the enable button.

Continue doing this for the amount of Cue Outputs desired. Next will be to insert the Spat Return Plug-In. You will be able to return to QLab as many Outputs as the Core Audio Device chosen in the Audio Patch and that you are currently editing. Note that Qlab offers single (1) and Dual (2) channel arrangements for Device output plug in insertion.

★ Although the return buses will appear in 2 channels in Spat Revolution they will be able to be fed from larger multichannel room and you will be able to patch the desired output to the desired return buses.

To do this, please press on the Device Outputs tab in QLab.



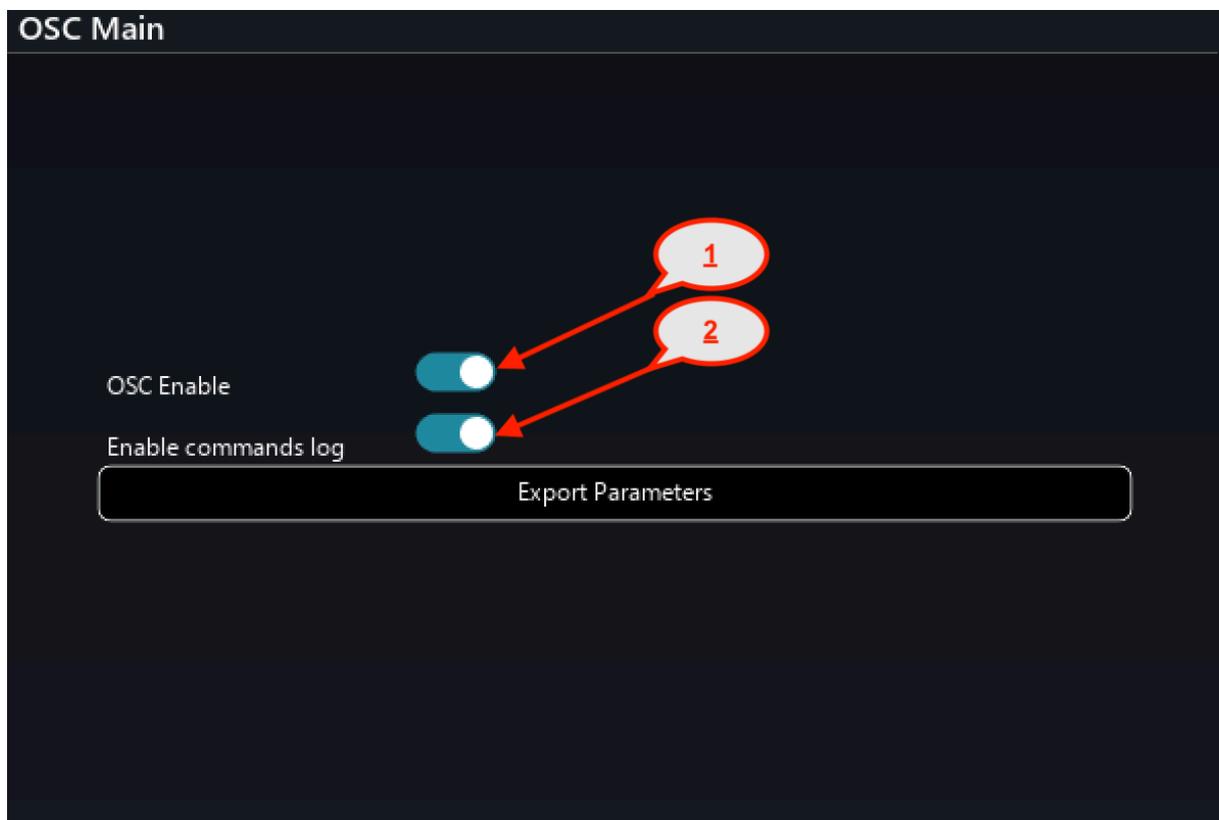
- 1 Choose either 1 or 2 channel arrangement in the pull down menu.
- 2 Click the Add effect
- 3 Scroll to the Flux category and Choose SpatRevolution-Return-x64.

The Spat Revolution Return Plug-In User interface will appear with the default Local audio path disabled. Make sure to Enable the audio path by clicking on the enable button. The Index is automatically generated for you and will be either 1 or the next available index if you have multiple instances of Spat Return in your current workspace. Note that the index represents the output number in Spat and is unique to each Spat Return instance.

## QLab and Spat Revolution via Network Control

In a realtime situation, where performers or sounds are being spatialised live by Spat Revolution, and cues need to be sent in the right running order with the rest of the show then Network OSC type cues can be sent from QLab to Spat to control all aspects of the Spat rendering software. To do this amazing interaction, it is necessary to setup the inter-application OSC communication. It is relatively straightforward.

In the Spat Revolution preferences make sure the OSC Enable is engaged.



★ Enable commands log to view the commands and confirm you are receiving data (Shift + F7 will open the log window)



1 change the pulldown from None to In (meaning you are setting a OSC Input to Spat)

2 Select the network interface you want to be receiving the commands from. Doing a local integration will require you to choose the localhost / loopback address 127.0.0.1

3 Set the Port # to the corresponding QLab Network Patch Output Port. In this case 53000

4 Choose if you want OSC only for Sources or Add Room to permit to change Room Parameters

# QLab Network Cues

General	Network Cue Destination Patches					
Audio		Name	Type	Network	Destination	Passcode
Video	Patch 1:	SPAT	address ↕	Automatic ↕	localhost	53700
Light	Patch 2:	Reaper	address ↕	Automatic ↕	localhost	8300
Network	Patch 3:	OSculator	address ↕	Automatic ↕	localhost	9005
MIDI	Patch 4:	Name	address ↕	Automatic ↕	Address	0
Cue Templates	Patch 5:	Name	address ↕	Automatic ↕	Address	0
Key Map	Patch 6:	Name	address ↕	Automatic ↕	Address	0
MIDI Controls	Patch 7:	Name	address ↕	Automatic ↕	Address	0
OSC Controls						

On the QLab side, use the Network settings to setup multiple OSC destinations. One of them can be SPAT - another could be Reaper or OSCulator or whatever else.

You can now send OSC network cues from QLab to Spat, and control *ALL PARAMETERS* of the virtual sources using Spat's OSC command dictionary. Once you get the hang of it, this is really very straightforward. As of QLab4 there is a time interpolated 2D fade system for creating spatial XY gestures or similar multi parameter control ideas.

**GO** Move XY 2 Meter Square - Source 1 - 2 Sec

Notes

Find: 0 found Done

Number	Q	Target	Pre Wait	Action	Post Wait
	▼ Position				
	Source 1 - Location preset		00:00.00	00:02.00	00:00.00
	Set Source 1 - Azimuth 90, Elevation 0 and Distance 2M		00:00.00	00:00.00	00:00.00
	Set Source 1 - XYZ - X (0), Y (2), Z (0)		00:00.00	00:00.00	00:00.00
	Set Azimuth Source 1 to 30 degree		00:00.00	00:00.00	00:00.00
	Set Distance Source 1 to 2 Meters		00:00.00	00:00.00	00:00.00
	Set Elevation Source 1 to 30 Degree		00:00.00	00:00.00	00:00.00
	Set Yaw Source 1 to.. (-180 to +180 Degree Default 0)		00:00.00	00:00.00	00:00.00
	Set Pitch Source 1 to.. (-90 to +90 Degree Default 0)		00:00.00	00:00.00	00:00.00
	Set Aperture Source 1 to.. (10 to 180 Degree Default 80)		00:00.00	00:00.00	00:00.00
	Set Scale Multi-Channel Source 1 to.. (0.10-100 Default 1)		00:00.00	00:00.00	00:00.00
	Set KNN speaker Neighbors % - Source 1 (0 - 100% Default 0)		00:00.00	00:00.00	00:00.00
	Enable Relative Directivity - Source 1		00:00.00	00:00.00	00:00.00
	▼ Movements				
	<b>Move XY 2 Meter Square - Source 1 - 2 Sec</b>		<b>00:00.00</b>	<b>00:02.00</b>	<b>00:00.00</b>
	Ping Pong Fx Source 1 - 5 Sec		00:00.00	00:05.00	00:00.00
	Circle Azimuth 1D move - 2 Sec		00:00.00	00:02.00	00:00.00
	Circle Azimuth Drawing move - 10 Sec		00:00.00	00:10.00	00:00.00
	Reset XY Source 1		00:00.00		00:00.00
	2D Map Test		00:00.00		00:00.00

Basics Triggers Settings

Destination: 1 - SPAT Type: OSC message Fade: 2D Fade at 30 fps with floats Send

/source/1/xyz #x# #y#



Duration: 00:02.000  
Max X: 2  
Max Y: 2

Enter an OSC address and arguments, e.g.: /path/to/a/method with arguments, insert position with #x# #y# Live Preview Off

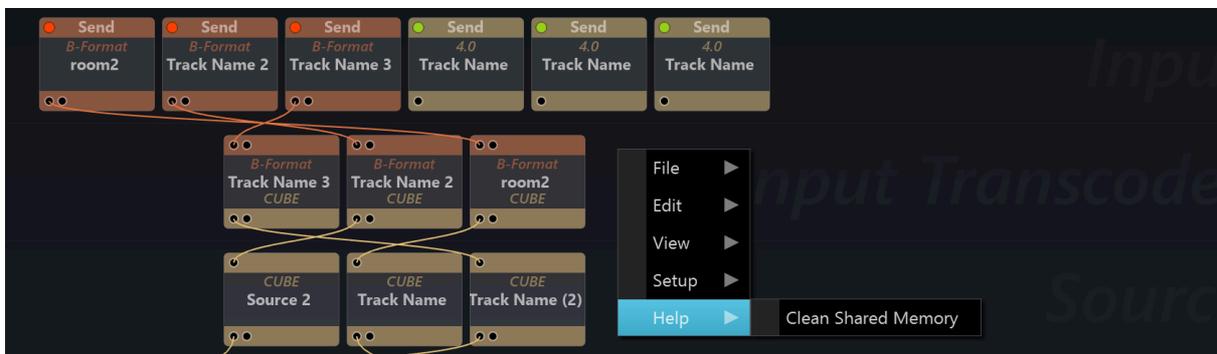
★ Flux team have created a full command template for QLab available on request

## Clearing Shared Memory

Currently with Spat Revolution v1.1 some users have experienced an issue where Spat SEND and RETURN plug-ins are not cleared from RAM after using certain third-party hosts. QLab is one of those.

If this happens, Spat SEND and RETURN modules will appear in the Spat setup graph editor, when there is no host software running in the background. It can cause problems, when a host with plugins is launched and more SEND and RETURN plug-ins appear to be doubled.

The workaround, if this is happening with your particular 3rd party software, is to invoke a special debug action in the Spat setup editor. It is called *Clean Shared Memory*. It is available by right-clicking anywhere in the background of the signal graph editor. A pop-up menu will appear with various options and shortcuts. Scroll to the bottom and choose *Help/Clean Shared Memory*



When this command is executed, **both** Spat and the plug-in host will then need to be restarted.

# 12.4 Avid VENUE S6L

Please contact Flux support to get a copy of the support files for integrating SPAT with this high end live mixing console. This guide outlines the basis of setup.

Currently, SPAT Send is the only available Plug-in from Flux available for the Spat Revolution integration to **Avid VENUE S6L** platform. There is an available file template from Flux included in the Avid VENUE show file with 64 channels (1 - 64) containing the Spat Send Plug-in and doing the integration via either the console AVB audio entity (In 128 Channel AVB Mode, officially qualified for MAC Pro) or MADI integration the with MADI-192 option card. The template show file contains the following criteria;

- Channel 1-64 are by default being fed from Stage 1-64 like normal

(Thus sending Top Pre to AVB 1-64)

- Post Fader pick up of channels 1- 64 are sent to AVB audio 65-128

(Simultaneously to MADI 1-64 for and audio routing option)

- 2 Snapshots. One for an **all source reset** and one for an **all source circle spread**

(Both with a global time value of 2 seconds)

- Spat Send Plug-In Index set to 1-64

(OSC parameters remaining to be set see below)

- OSC integration done over the AVB Network port (Port C) or via ECx port

A Spat Revolution .JSON file is as well provided containing a 64 audio hardware inputs sources feeding a 3rd order ambisonic room and a transcode to be adjusted to a desired channel base speaker setup and as the following criteria;

- Default template built using AVB E6 Engine128 as a Spat audio hardware device
  - Input 1-64 are fed from hardware audio 65-128

(matching the above Avid S6L AVB patch, see input matrix)

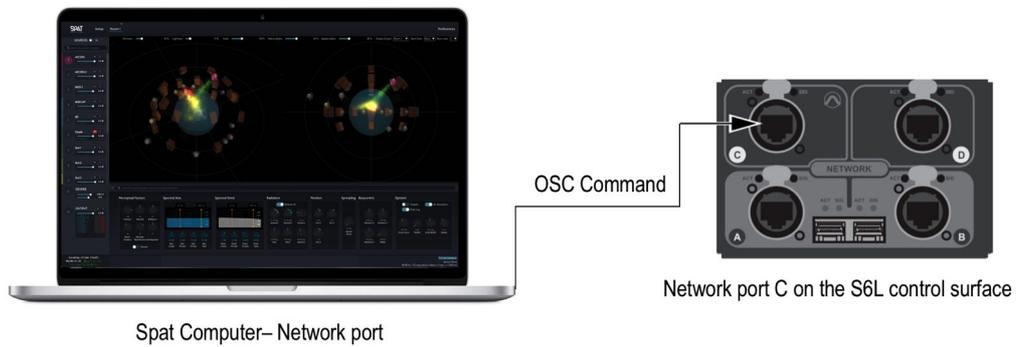
- A transcode module available to bring this to specific channel base speaker setup

The OSC part is the key aspect to this integration as when properly configured, the console will be sending OSC commands to Spat (OSC Out) and receiving OSC commands from the S6L console (OSC In). Thus for a bidirectional configuration. This is what the Spat Send plugin does on the console. Optionally is the ability to set a second OSC output from the console to send to a backup Spat Revolution computer engine

The plug-in offers the ability to control the SPAT object/source parameters and automate them within the VENUE show file and the VENUE Snapshot system. Ultimately having a completely integrated control of the objects in the 3D space. Added to the Spat Revolution Source parameters are global and per parameter time values the user is able to use to set a linear morphing between two parameter states over a period of time to create smooth transitions. Thus creating movements in time with the VENUE Snapshots. Important to note that all parameters (with time value) become available on 4 page tables in the encoder section for the plugin.

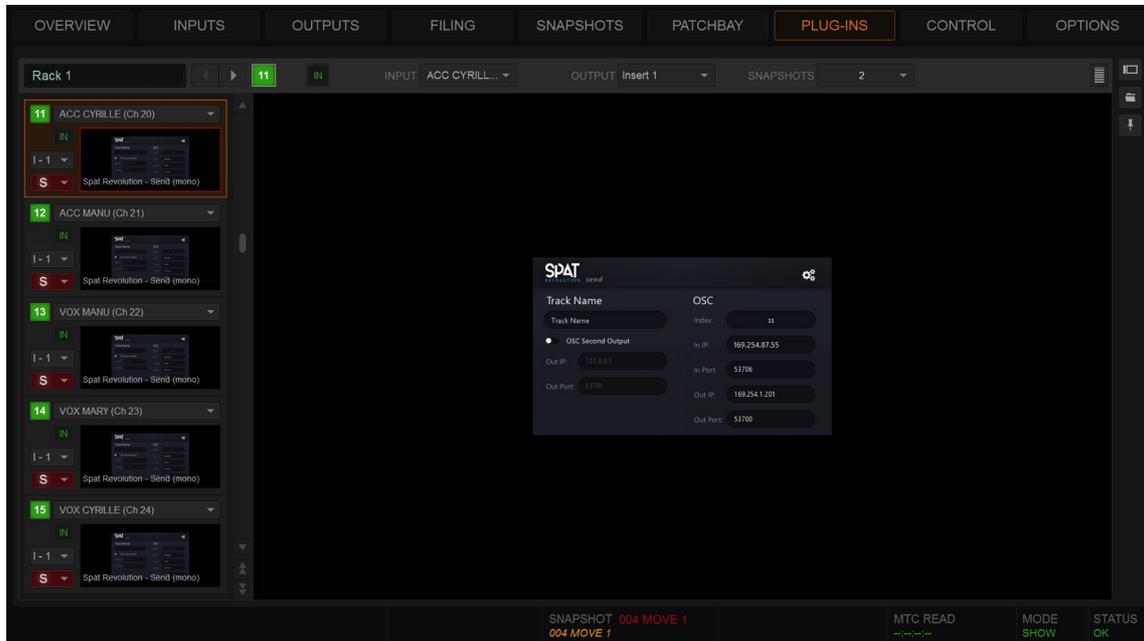


**Spat plugin on S6L CKM module - Page 1 of 4**



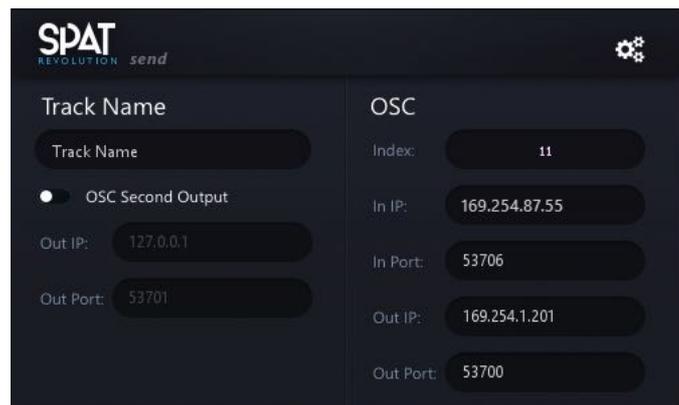
## Spat Revolution computer connected to the AVB Port

Now let's configure the OSC part of this integration. Although provide files are a start template to explore, the OSC settings (plugin and Spat Revolution) are specific to your console and to the Spat Revolution preference



### Spat Send Plugin in the Plug-Ins rack of VENUE

## Spat Send Plugin setup



The PI interface doesn't have much and it's straight forward.

- Index refers to the object number in SPAT application. Automatically generated every time there is a new instance. (Can be changed by snapshots too)

- In IP. Pull down will allow you to choose the ethernet interface you will be listening to OSC commands in the S6L coming from SPAT (you will need this address for SPAT preference setup later). This can be the AVB port 169.254.x.x or your ECx port of the console
- In Port#: It needs to be the corresponding Port# on your Out Port in the SPAT application preference and can be left to the default port (more on this later)
- Out IP. 169.254.x.x or whatever address your SPAT application MAC is at.
- Out Port#: It needs to be the corresponding Port# on your In Port in the SPAT application preference and can be left to the default port (more on this later)
- Second Output checkbox is to send the same commands to a backup SPAT rig! Yes redundancy!

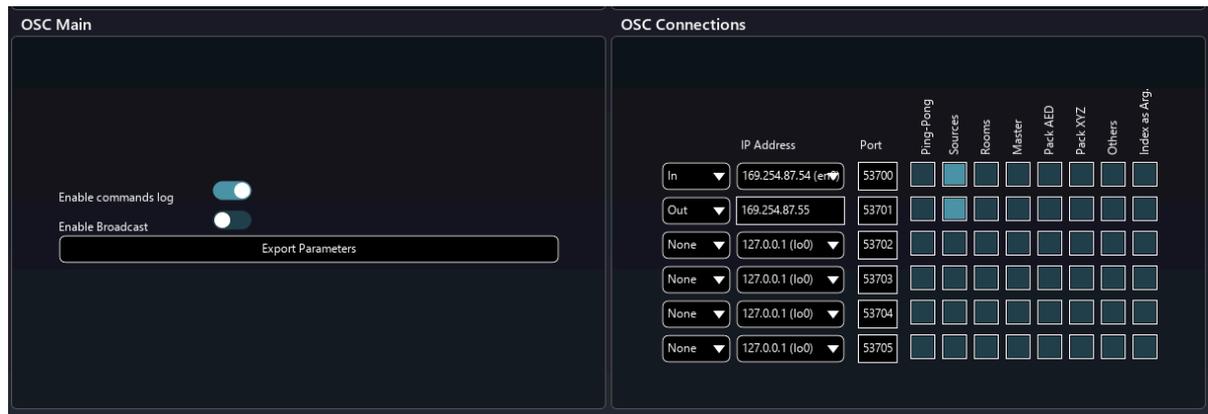
Note that any changes you do properties wise to the Spat Send interface get done across the board (across all PI instances)

## **Setting the preferences of the Spat Revolution Application**

Before to open a template Spat .JSON project file or to start creating your project, let's set up your preferences in Spat Revolution preferences page :

- Hit preference in the right corner. Look for the OSC connection section. You will see 6 OSC slots... Set one to In and one to Out. We will use 2 slots for this. Out will be to go to S6L Spat Send PI. In will be to listen to S6L in SPAT

## Spat OSC Connections preferences



- Set the In IP address by pulling then the menu and choosing the 169.254.x.x of your SPAT application computer (or whatever network interface you are using for this communication)
- Set the In port # to the desired port #. This is the port # that you need to put (or have) in the Out Port # of the S6L Spat Send PI. For Example you can use 53700
- Set the Out IP address by putting the IP address of the S6L Console (you will have found this address in the In IP of the S6L Spat send PI above)
- Set the Out port # to the desired port #. This is the port # that you need (or have) in the In Port # of the S6L PI.
- On the 8 checkboxes (Ping-pong, Source, Room, etc) you will for now just engage Source on both the In and Out (Basically we are only moving source objects on this integration for the moment)

We recommend in the Spat OSC Main section to activate "Enable commands logs" option as this can help to confirm OSC communication .

## Spat Hardware IO (Audio) preferences



- In Devices, please select your core audio device. In this case we are using the AVB Core Audio E6 Engine entity in 128 channel mode but this could be as well your MADI interface of choice if doing this via AVB
- Sample rate should be set at 96000 Hz and buffer can be set to your desired buffer (mind lower buffer will required good computer resource and may not need to be really low unless dealing with IEM mixes for example)

Now in SPAT Revolution Setup page

- Open the Flux Spat Revolution provided template file or created your own project
- Right click on the top right SPAT logo and click show mini terminal (Shift + F7) ... You will see OSC message log window appear which can be practical for this first test

Testing configuration :

In SPAT Revolution Room page, you will see all your virtual sources in the sound field

- You can now move objects from the Plug-in CKM encoder page table of the console and see them move (and confirm OSC messages with your OSC Log terminal)
- In Spat Revolution application, on the left side are all your sources. Double clicking on an object number will open the source properties. Changing the radiation properties for example should as well change the values in the PI

encoder page (Again the terminal can confirm this). You are then equip to start building snapshots containing you plug-in parameters.

- If using the Flux Spat Revolution and S6L template files, you can try using snapshot 1 and 2 already created to show a all source reset and all source circle spread move made in 2 seconds.

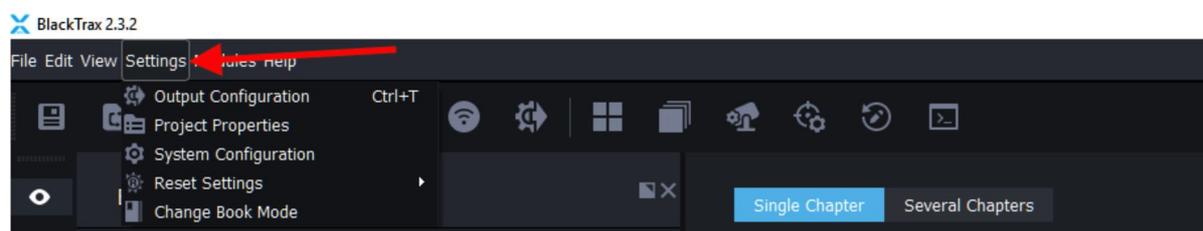
# Appendix A

## Flux Spat Revolution and BlackTrax tracking system integration (RTTrPM)

Starting in Spat version 1.1, a new UI implementation is available in order to integrate BlackTrax RTTrPM tracking protocol in Spat. This provides a way to map BlackTrax tracking beacons with Spat source objects and Spat room listeners.

In order to do this, you will need to configure the output configuration of a BlackTrax system and have your Spat (or multiple Spat) computers networked on the same third party network interface (NIC) of the BlackTrax server hardware.

In the BlackTrax software, press on the output configuration button or access output configuration via the menu bar in the Settings section . Shortcut Ctrl+T



In the Output Configuration window, you will need to create with the + sign 1 (or multiple if running a main and a back up Spat computer engine) output that will be sending tracking RTTrPM data to Spat. Clicking the + sign will give you an edit window

## Output Configuration

Active Outputs

	Label	Type	Output NIC	Communication	Details	Status
1	Lighting	RTTrPL	127.0.0.1	Multicast	238.210.10.3:6456	Disabled
2	Spat Back Up	RTTrPM	127.0.0.1	Unicast	127.0.0.1:24002	Sending
3	Spat Main	RTTrPM	192.168.1.200	Unicast	192.168.1.201:24002	Sending

Enable + -

In the edit window, you will first label you Output to the desired name (Ex: Spat Main). Type of of output will be RTTrPM, the third party motion protocol of BlackTrax.

NIC is the actual network interface that you are doing the integration to Spat. The default NIC of BlackTrax server for RTTrPM (They do have multiple NIC for various use such as lighting and the actual BlackTrax network itself) is 10.133.3.9 - 255.255.255.0. You will want to make sure your Spat is either in the same range as this address. (10.133.3.x, 255..255.255.0) or use your addresses of choice in the same range (Please consult with a certified BlackTrax support representative before to attempt to change IP).

In the below example, addresses of the Blacktrax NIC was changed to 192.168.1.200

Communication should be set to Unicast. Please refrain from using Broadcast unless specific circumstances. Currently Multicast is not implemented with Spat software.

Port number should be left to the default port # 24002 as it is the default port for the BlackTrax system and the default port preconfigured into Spat software.

Press Apply and you are set with this step.

Edit Output

Network **Advanced**

Label: Spat Main

Type: RTTrPM ▼

NIC: 192.168.1.200 ▼

Communication: Unicast ▼

Address: 192.168.1.201

Port: 24002

Apply

Edit Output

Network **Advanced**

Coordinate System: Stage (WYSIWYG) ▼

Integer: Big Endian (Network Order) ▼

Float Point: Little Endian (Intel Order) ▼

Orientation: Euler ▼

Apply

Pressing the Advanced button will give you some advanced option. Your default Coordinate System should be Stage. You can apply these settings.

You are now set. Before BlackTrax will actually output anything to the Spat system, you will need to enable the Beacons present in the system that you actually want to be sending tracking data.

In the Active Output window, select you Spat Main output. On the bottom right, select the Beacon you would like to have data sending and press Enable Selected. You can multi-select when pressing the CTRL key or select all with CTRL+A

Active Outputs

	Label	Type	Output NIC	Communication	Details	Status
1	Lighting	RTTrPL	127.0.0.1	Multicast	238.210.10.3:6456	Disconnected
2	Spat Back Up	RTTrPM	127.0.0.1	Unicast	127.0.0.1:24002	Disconnected
3	Spat Main	RTTrPM	192.168.1.200	Unicast	192.168.1.201:24002	Disconnected

Enable + ❌

Sending Trackables

Beacons Mini Rigid Soft

Enabled	Beacon	Smoothing (-4 to 2 inclusive)
	1	
✓	2	0.00
✓	3	0.00
✓	4	0.00
	5	
✓	6	0.00

Output Options

Trackable Settings Output Modules

Smoothing (-4 to 2 inclusive):  
Smooth

Enable Selected
Disable Selected

When done Like with all operations with BlackTrax, you want to Apply the changes to the system. You can at this point close the Output Configuration window and go back to the main BlackTrax page. You will want to apply the changes to the system

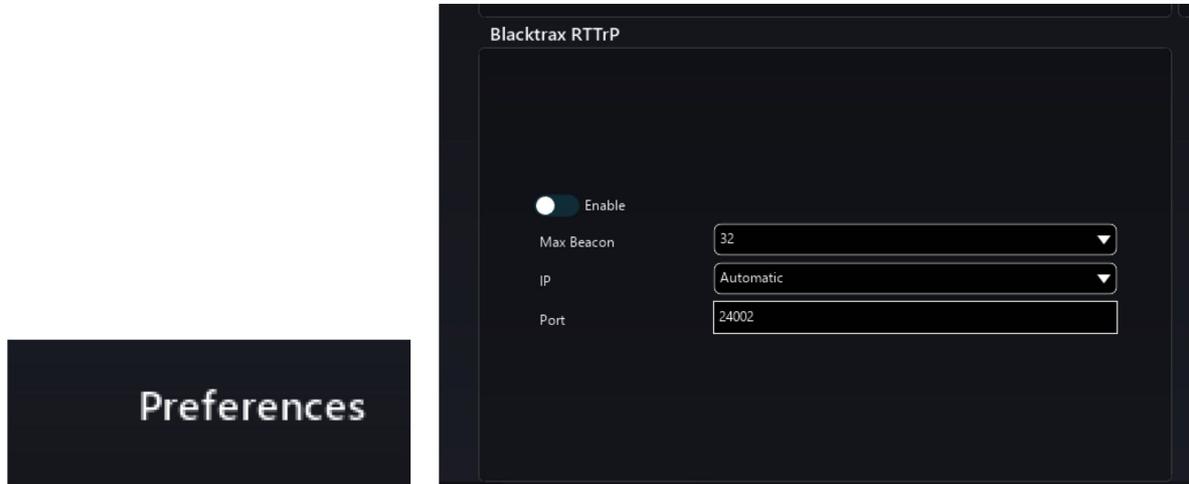
Beacons Fixture Settings Trackable Settings

System Standby MCV: N/A ▶ 👤 ⬇ 💬 ● Changes: Live Apply Changes

You can confirm that you are actually sending data by going back to the Output Configuration page. The status will give you this information

	Label	Type	Output NIC	Communication	Details	Status
1	Lighting	RTTrPL	127.0.0.1	Multicast	238.210.10.3:6456	Disabled
2	Spat Back Up	RTTrPM	127.0.0.1	Unicast	127.0.0.1:24002	Sending
3	Spat Main	RTTrPM	192.168.1.200	Unicast	192.168.1.201:24002	Sending

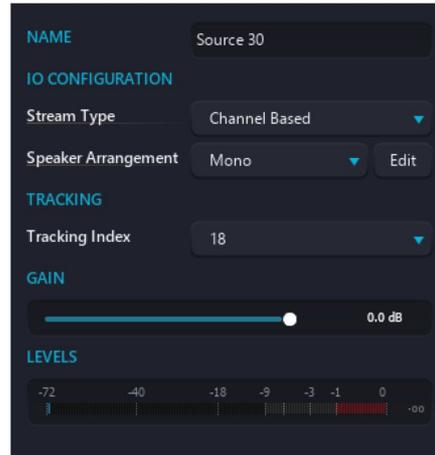
Now let's go and configure the Spat software. In the preference page accessible by clicking on the top right button you can scroll down to the BlackTrax preference section



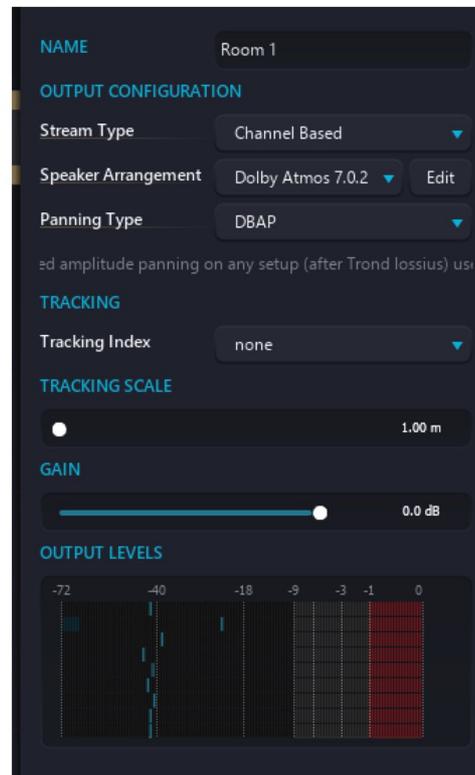
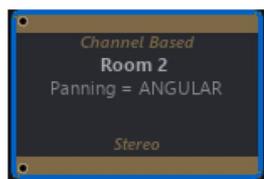
You can from there, Enable BlackTrax RTTrPM tracking, Select the amount of Max Beacon you system may need (1, 2, 4, 8, 16, 24, 32 or more), Select the network interface you are using on your computer for the BlackTrax integration (Or leave to Automatic) and then the port # which is already set at the default port # used by BlackTrax RTTrPM output.

You are set. These new preferences will be store with all your preferences and remain after closing and restarting Spat Revolution software.

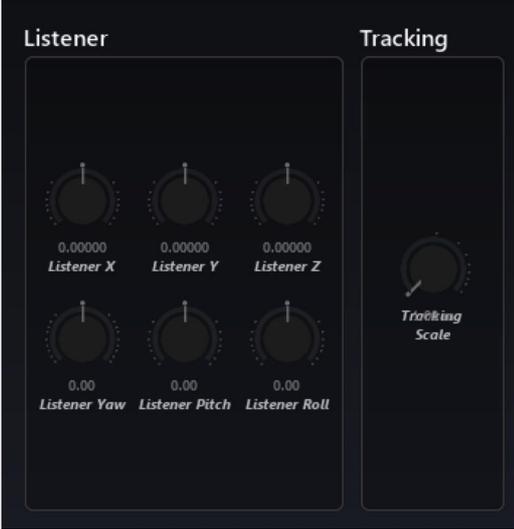
First option is to assign a Beacon ID to an actual audio source object. Selecting a source will give you the source menu options. In the option are now Tracking options and a pull down to choose the Tracking Index (The beacon ID number you want for this audio source). The list will be based on your Max Beacon preference. It is possible to have the same Index for multiple source providing the ability to have multiple sources track a single beacon



Your second option is to assign a Beacon ID to an actual room in order to have the listener position tracked. Selecting a room will give you the room menu options. In the option are now Tracking options with a pull down to choose the Tracking Index (The beacon ID number you want for this room listener). The list will be based on your Max Beacon preference. A Tracking Scale setting is as well provide which would allow to scale the incoming data to adjust to your speaker output setup size if need b



In the actual room output parameters in the room view, you will also see a tracking option become available.



# Appendix B

## Spat Send and Return DAW Troubleshooting

Sync issues encountered in Spat Revolution when using the Local Audio Path workflow, can often be fixed when following some guidelines for routing order inside the DAW.

If you are using the Cockos REAPER DAW take a special look at the options that need to set to avoid sync problems (section 12.1).

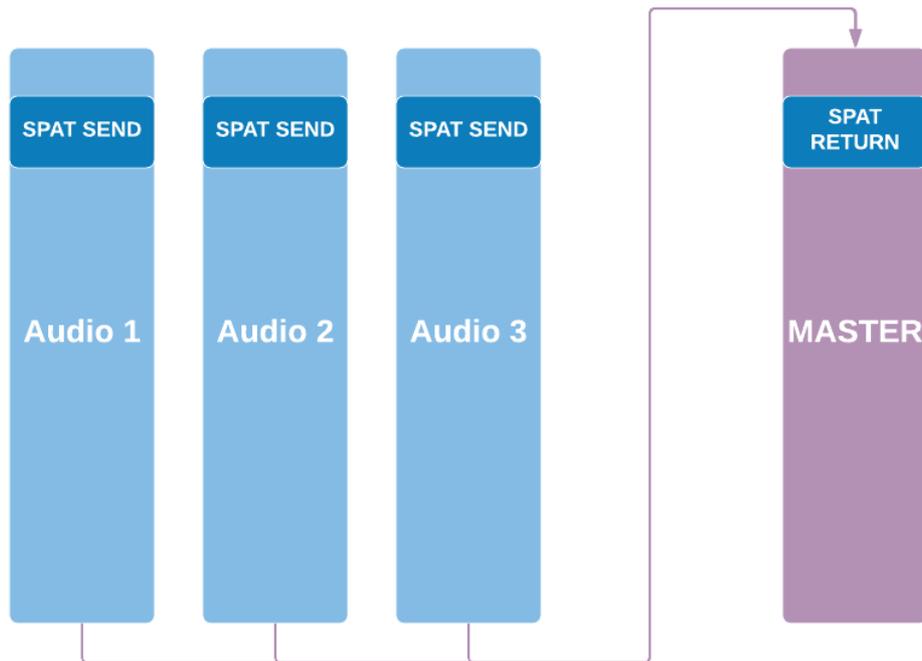
To work properly, the SEND plug-ins instances must be processed by the DAW **before** the RETURN plug-in instances. To force the DAW to process this way, each track with SEND plug-in inserted must be routed (directly or indirectly) to the tracks hosting a RETURN plugin, using DAW internal routing.

Following are 4 examples of recommended practice with DAW routing, which should cover the main use cases. If your problems persist even after implementing these suggestions, don't hesitate to drop us a line at Flux support.

**PLEASE NOTE:** *In the current version 1.1, the mixing of HARDWARE inputs and LOCAL AUDIO PATH Send inputs may report a sync loss, as Spat Revolution cannot guarantee correct sync in this scenario. Proceed with caution if this is unavoidable.*

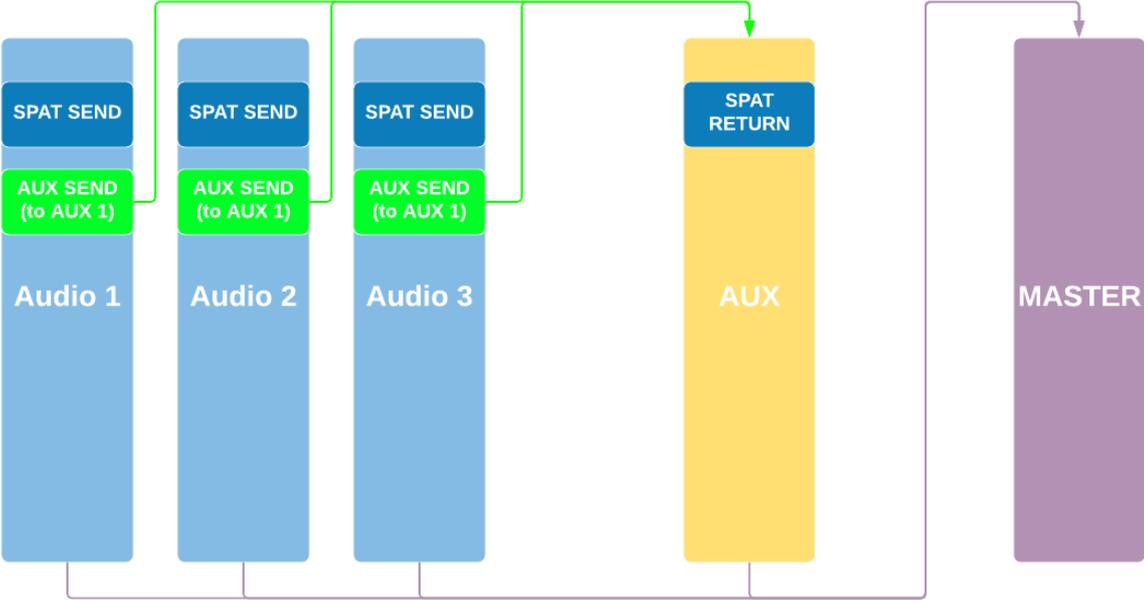
## Return on master track

In simple projects, when having a single return plugin on master track, you should not encounter any sync issue as long as each SEND track is routed to the master.



# Single RETURN on an AUX track

Issues may happen when return is inserted on an AUX track. Make sure that each SEND track is routed to the AUX track (RETURN track) using AUX send (see example below).



★ You might achieve the same by routing the track outputs to a buss

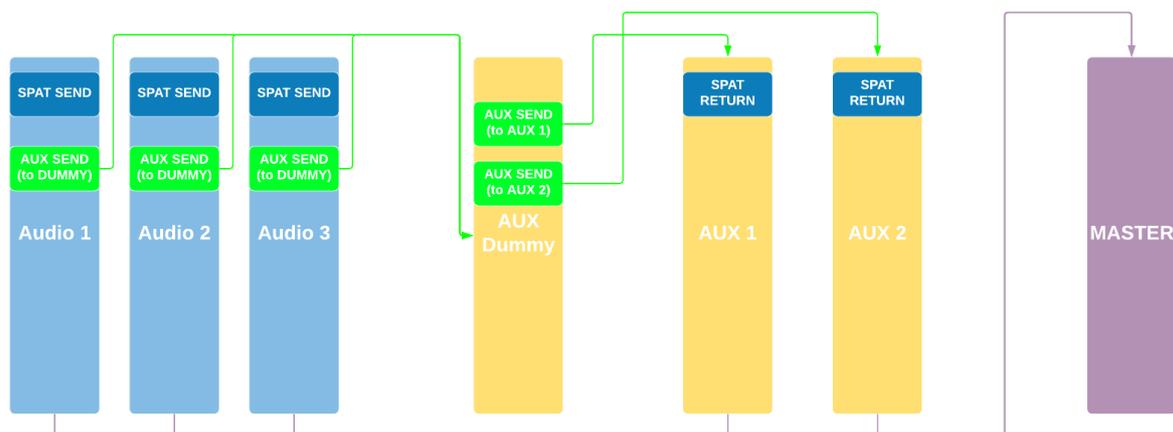
## Several RETURN on AUX tracks

When you need several RETURN tracks (for example several rooms in Spat Revolution, and/or several outputs), you will have to route each SEND track to each RETURN track, using the same technique.

But it can quickly become complicated as the project grows. So, in the following example, we added a 'dummy' track, that avoids using several AUX sends on the 'SEND' tracks. It makes routing more clear and fast to setup on large projects.

The 'dummy' AUX track is routed to all the RETURN tracks (using AUX send)

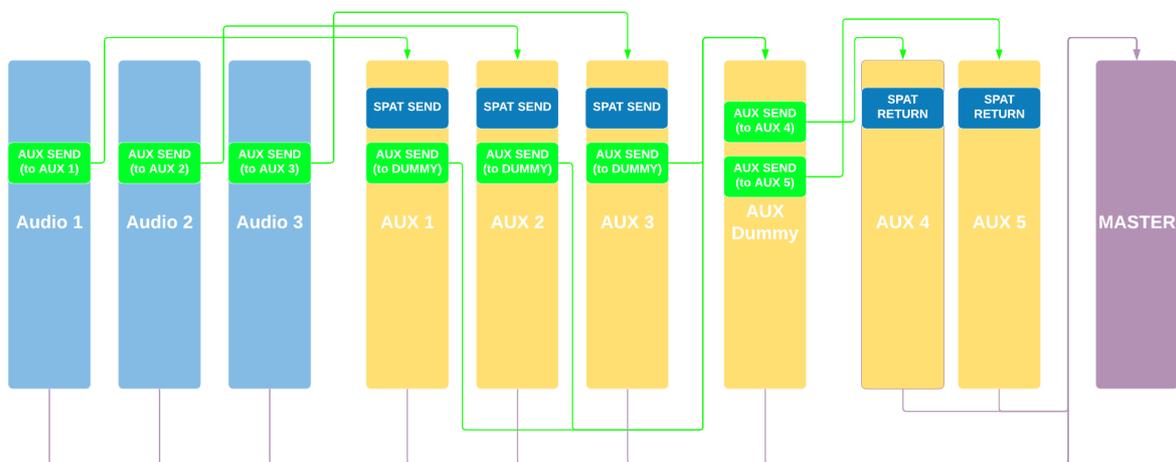
Then, simply route all your SEND tracks to this 'dummy' track (using aux send).



## Independent Dry and Wet Signals

In larger projects, when you need to keep both the dry signal and the Spat RETURN signals independent (when the mix requires to switch easily from one to the other), add one AUX (pre-fader may be useful) per audio track to be sent to Spat, and insert the send plugins on these AUX tracks.

This way, you keep the dry signal on the audio track's output.

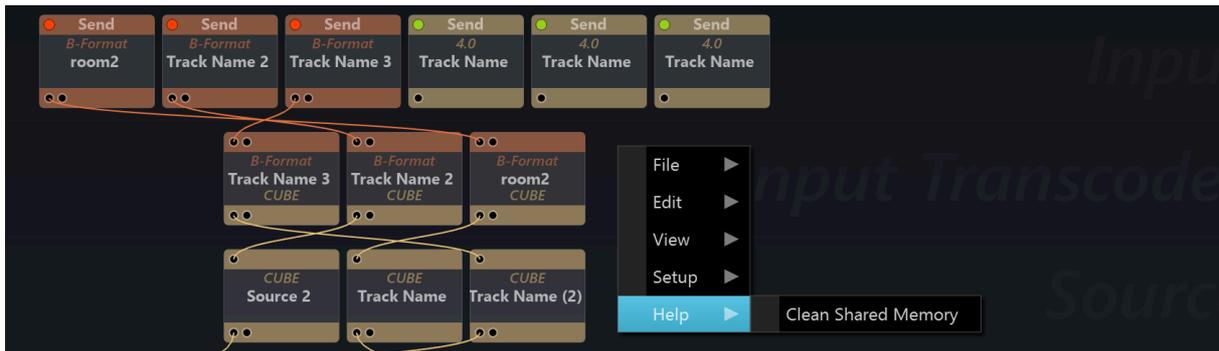


## Clearing Shared Memory

Some users have experienced an issue where Spat SEND and RETURN plug-ins are not cleared from RAM when used with certain third-party hosts.

If this happens, Spat SEND and RETURN modules will appear in the Spat setup graph editor, when there is no host software running in the background. It can cause problems, when a host with plugins is launched and more SEND and RETURN plug-ins appear to be doubled.

The workaround, if this is happening with your particular 3rd party software, is to invoke a special debug action in the Spat setup editor. It is called *Clean Shared Memory*. It is available by right-clicking anywhere in the background of the signal graph editor. A pop-up menu will appear with various options and shortcuts. Scroll to the bottom and choose *Help/Clean Shared Memory*



If this command is executed, Spat and the plug-in host will then need to be restarted.

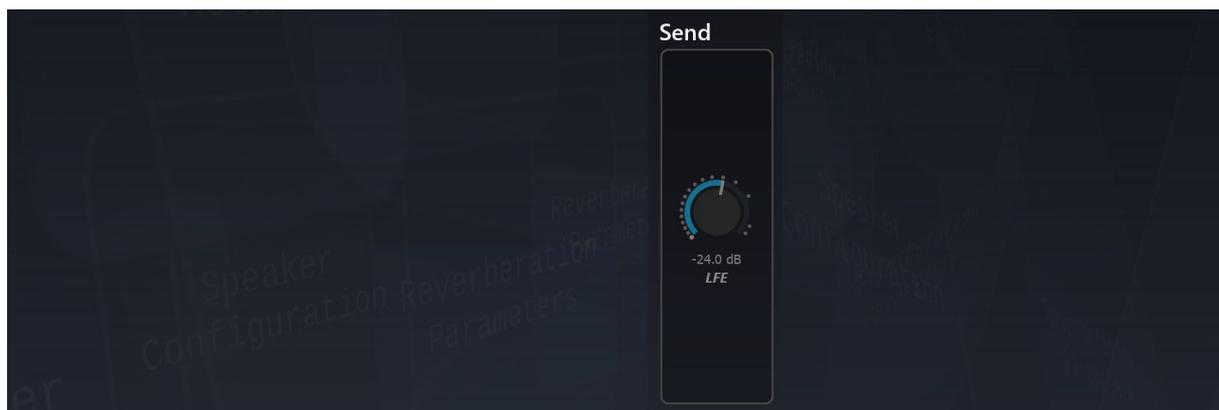
## Appendix C

# Sub and LFE Channels

In some standard surround formats for cinema - often denoted as *n.1 surround* - the *.1* refers to a dedicated channel for extra sub bass effects in cinema. The LFE channel is normally sent directly to the sub-woofer speakers in a real room.

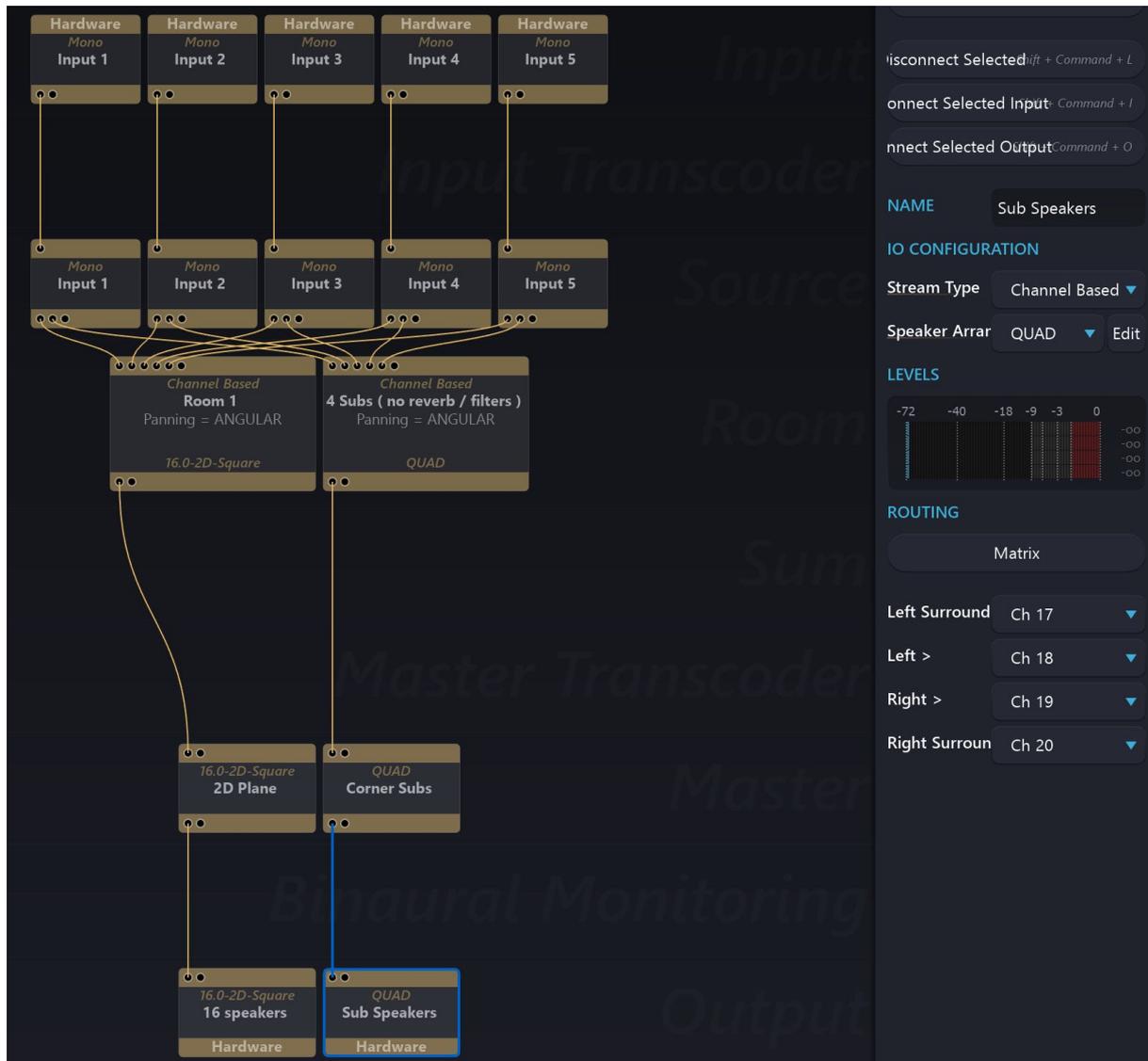
SPAT does not model sub-woofers or spatialise their acoustic information in the Virtual Rooms. But it does handle the channel in a special way.

When working in a purely Channel Based signal graph, with Channel Based rooms outputting to an *n.1 surround* speaker arrangement, each source will have an individual *LFE Send* which controls how much of its signal is sent to the LFE channel at the output.



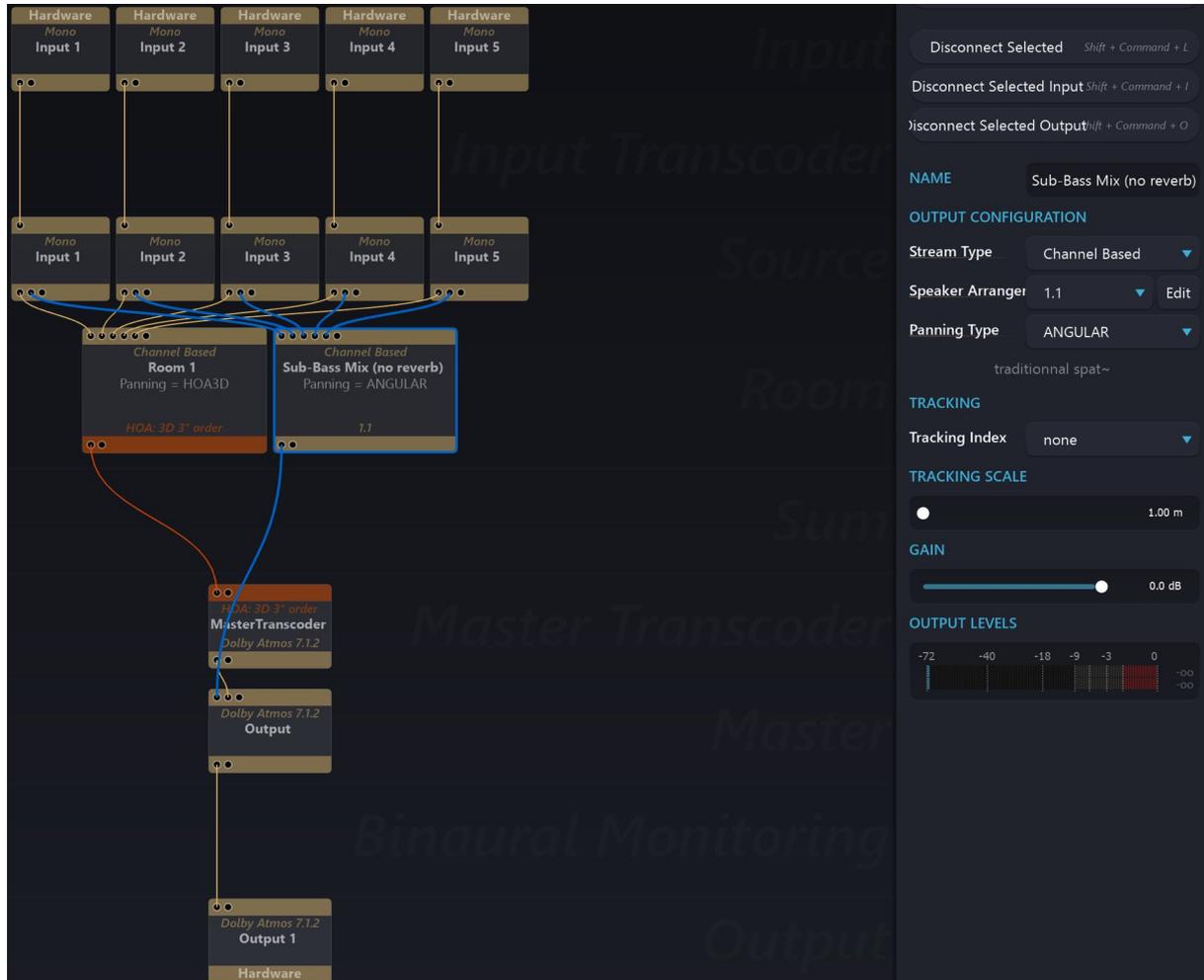
This parameter is available inside the Room editor, but also available directly from a Source module in the *Environment Setup* graph.

# Spatialised Sub-Woofers

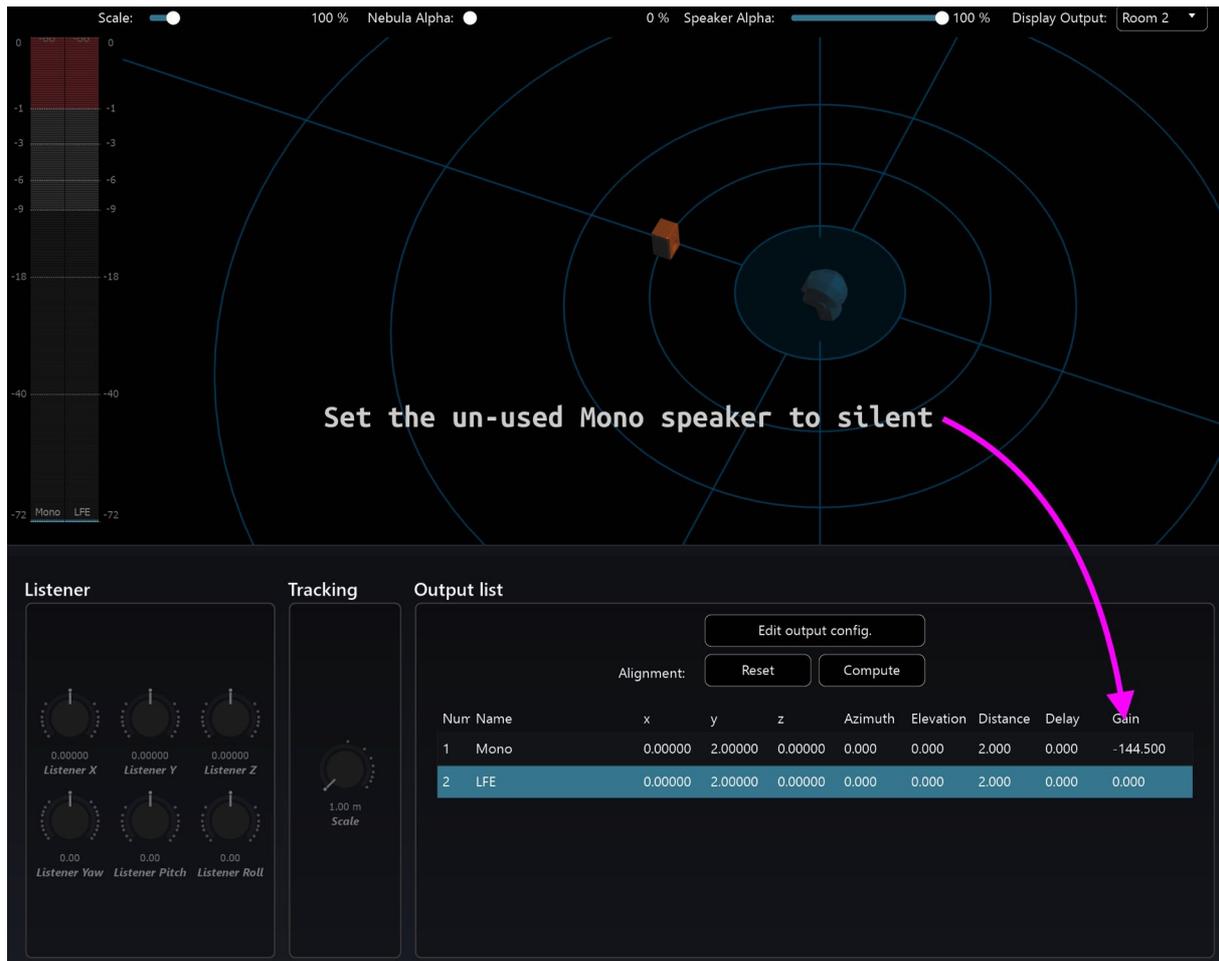


In a venue or a system that has sub-woofers in the corners or at particular locations, sometimes it is a good idea to use them all independently. Here is a suggested set up for spatialisation over a system of 16 satellite speakers in a 2D-Square and 4 sub-woofers placed in the corners of a room in a pseudo Quad configuration.

# LFE Mixing from Ambisonic or Binaural Rooms

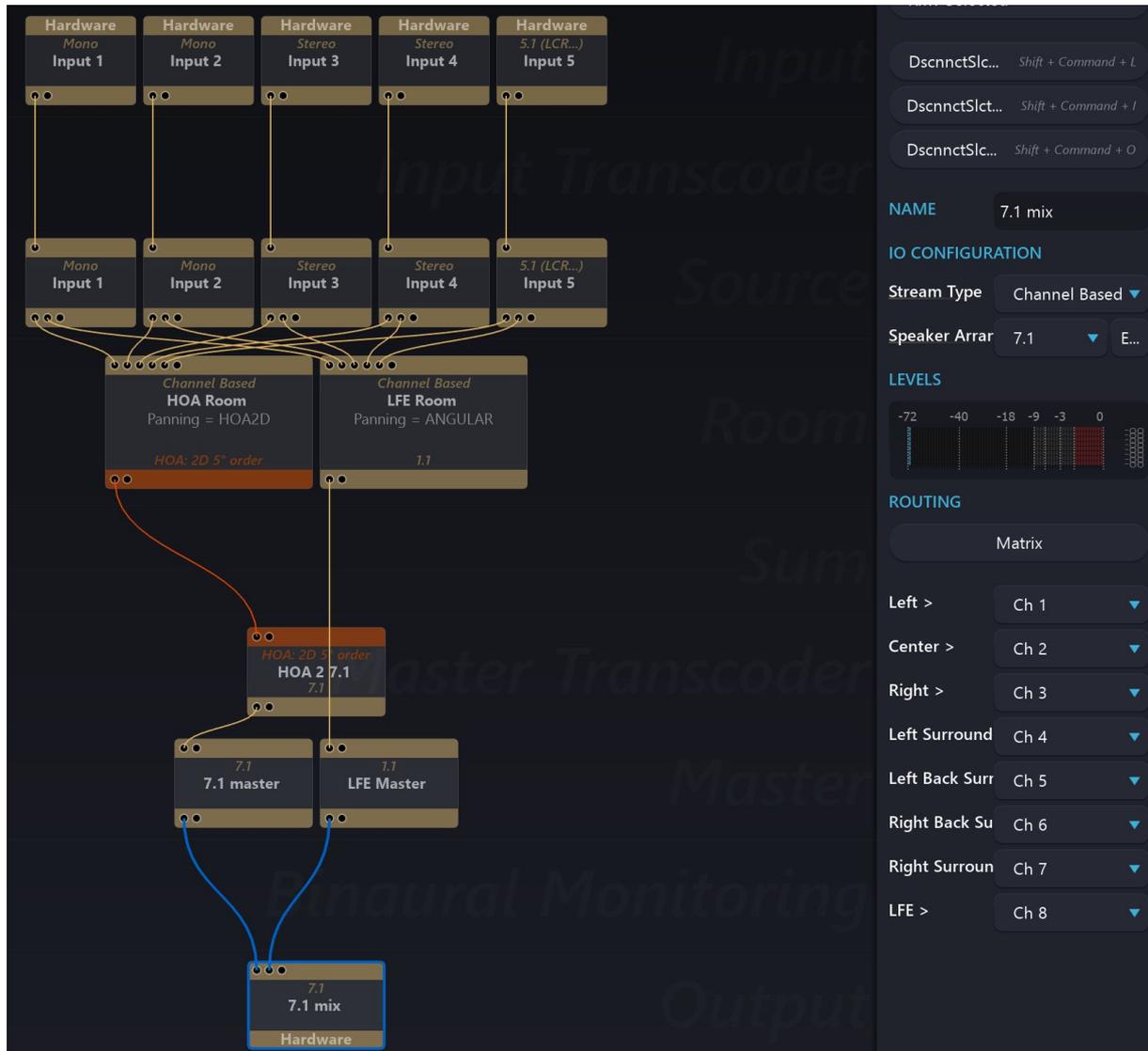


Currently in Spat Revolution version 1.1, it is necessary to create a mono mix for a LFE channel using a workaround method. To get a mono LFE mix that will be summed into the  $n.1$  channel in a surround configuration, create a secondary mix room. It should not have reverb, and is a special channel based configuration called  $1.1$  - a Mono speaker plus LFE room.

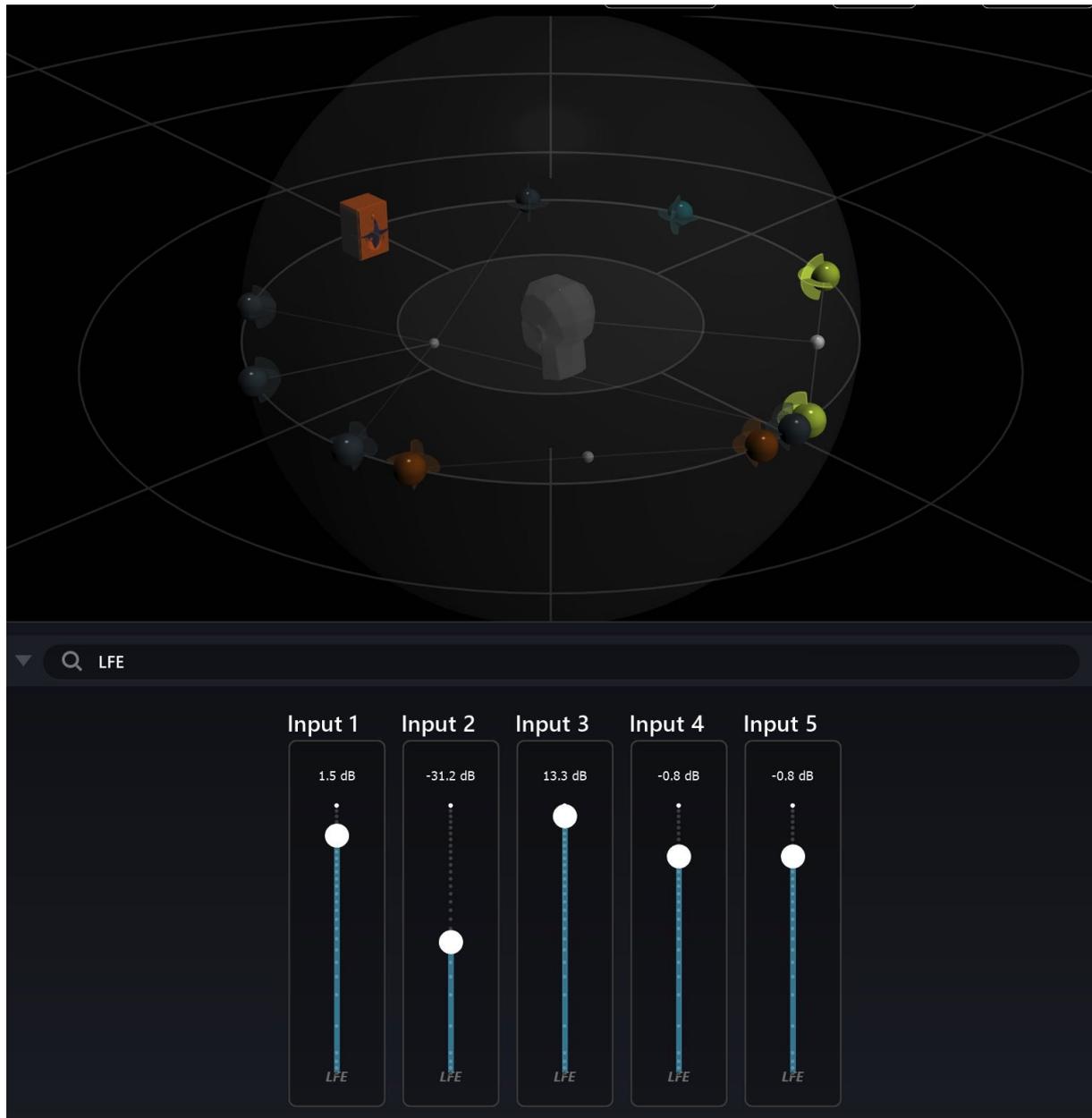


As we are not going to have an actual satellite speaker in this room, we need to set the unwanted virtual mono speaker gain to -144.5 (silent) inside the 1.1 LFE room. Sum the room output in a Master module with a transcoded *n.1* Channel Based stream from the HOA room. The LFE channel from the LFE room will be summed as the *.1* LFE Channel of a surround format at the output module. The overall LFE output level can be controlled using the Room Gain of the Mono room and each individual Source can send to it with a separate level ( which could be automated from the DAW of course).

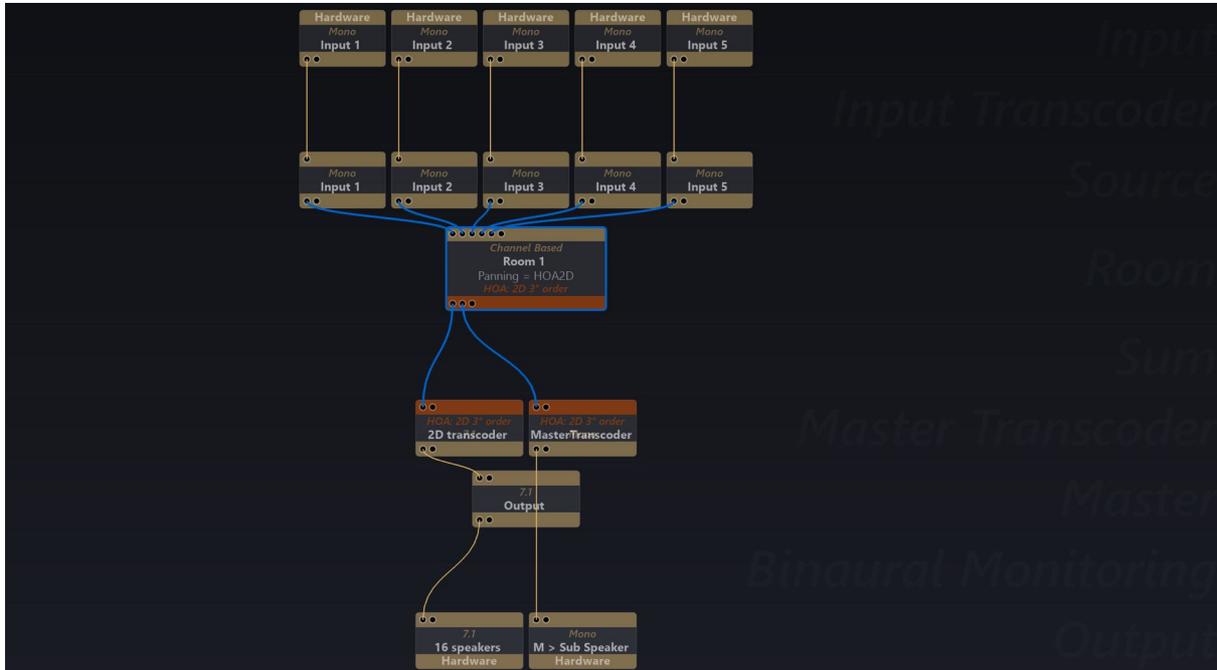
## How to Sum the .1 LFE



## Use the Parameter Filter to create mixer of LFE sends for each source



Here is another 'rough and ready' suggestion that will yield a separate output route for the bass content.



Reduce the output of the HOA room down to MONO using a master transcoder module and send that to the sub speaker. The sub-bass signal will have its own gain control in the Master module. Transcode to any Channel Based format and automatically get a mono mix down of the HOA Room activity for all setups.

## **Appendix D**

# **Spat Revolution Credits**

### **Project Manager:**

Gaël Martinet

Hugo Larin

### **Designer:**

Gaël Martinet

### **Application Development**

Gaël Martinet

Siegfried Hand

Antoine Lorence

Alexis Gentil

### **DSP Development (*for "Spat~" see below*)**

Gaël Martinet

Maxence Grandidier

Lorcan Mc Donagh

### **Graphic design**

Nicolas Philippot

Gaël Martinet

### **User Guide and Documentation**

Cristian Vogel

Hugo Larin

### **Additional contributions:**

Vincent Carlier

Anders Tveit

## **Flux:: Framework development**

Gaël Martinet

Siegfried Hand

Antoine Lorence

Lorcan Mc Donagh

## **Additional contributions:**

Alexis Gentil

Hadrien Marquez

Vincent Carlier

Jean Cruypenynck

Pablo Arias

Samuel Tracol

## **Flux:: Framework graphic engine**

Emmanuel Julien (GS lib)

Gaël Martinet

## **Additional libs:**

NuXPixels (Magnus Lidström)

spline lib (Joachim Klahr)

LibJpeg

Freetype 2

Zlib

Boost

## **Additional Python modules:**

argh (Andrey Mikhaylenko)

Crypto (Dwayne C. Litzberger)

Docopt (Vladimir Keleshev)

ecdsa (Peter Pearson)

enum (Ben Finney)

ipgetter (Fernando Giannasi)

paramiko (Jeff Forcier)

pathtools (Yesudeep Mangalapilly)

psutil (Giampaolo Rodola)  
watchdog (Yesudeep Mangalapilly, Google, Inc)  
yaml (Kirill Simonov)  
winshell (Tim Golden)  
six (Benjamin Peterson)  
Request (Kenneth Reitz)  
RTTrPM (Blacktraxx)

## **"Spat~"**

### **Digital signal processing algorithms and implementation in Max**

Jean-Marc Jot (Espaces Nouveaux / Ircam).

## **"Spat~"**

### **Objective and perceptual characterization of room acoustical quality**

Jean-Pascal Jullien  
Olivier Warusfel  
Eckhard Kahle

### **Additional contributions:**

Gerard Assayag  
Georges Bloch  
Martine Marin  
Véronique Larcher  
Guillaume Vandernoot  
Khoa-Van Nguyen  
Markus Noisternig

## **"Spat~" C++ dsp code**

Thibaut Carpentier  
Remy Muller

### **And**

thanks to all fantastic testers...

## **Flux:: Special Thanks to**

Alain, Yves, Bruno and Claude for helping to shape our minds over the years  
HAL for its support

## **Ircam Special Thanks to**

Xavier Chabot, Eric Daubresse, Gerhard Eckel, Serge Lemouton, Gilbert Nouno, Laurent Pottier, Manuel Poletti, Leslie Stuck, and Zack Settel for instructive discussions and advice.

## **Ircam R&D Director :**

Hugues Vinet

## **IRCAMTOOLS Collection Manager for IRCAM:**

Frédéric Rousseau

## **Collection Manager for Flux::**

Gaël Martinet

**Flux:: sound and picture development**

**Orléans, France, [www.flux.audio](http://www.flux.audio)**

**Copyright 2018 Flux:: sound and picture development, All Rights Reserved.**

IRCAMTOOLS SPAT and "SPAT Revolution" Copyright 2018 Flux:: sound and picture development and IRCAM. All rights reserved.

*The Ambisonic Logo, a trademark formerly owned by Wyastone Estade Ltd, which expired in 2010 is featured on page 30 of this guide*