

SPAT Revolution

FLUX:: Immersive

2023-02-06

Table of contents

FLUX:: Immersive SPAT Revolution 23.08	22
A User Guide	22
I Welcome to SPAT Revolution	24
II SPAT Revolution’s Heritage	27
III About this Guide	29
Getting started	30
1 Users and Applications	31
2 Installation and Activation	33
2.1 How to Install SPAT Revolution?	33
2.2 Create and account	33
2.3 Redeem a License Code from the activation code	33
2.4 iLok User Account	34
2.5 iLok License Manager	35
2.6 Transferring license	35
2.7 FLUX:: Center	36
2.8 Center Preferences	37
2.9 Center Preferences and Options	37
3 Ultimate and Essential licenses	39
3.1 Ultimate and Essential: what’s common?	39
3.2 Ultimate and Essential: differences	39
3.3 Ultimate and Essential sessions compatibility	40
3.3.1 Check Essential compatibility	40
3.3.2 Essential compatibility mode in Ultimate	40

IV	Spatialization Technology	42
4	Listener Position	44
4.1	Listener position and head tracking	46
5	Binaural	47
5.1	Introduction	47
5.2	HRTF	47
5.3	HRTF Profiles	48
5.4	Binaural algorithms	48
5.4.1	Standard binaural mode	48
5.4.2	Advanced algorithms	48
5.4.3	Head scale parameter	49
5.5	Binaural Monitoring Module	49
6	Transaural	52
7	Channel Based Streams	53
8	Panning Algorithms	55
8.1	Stereo-exclusive Panning Laws	56
8.1.1	Stereopan Law	56
8.1.2	XY and AB	56
8.2	Vector Base Amplitude panning (VBAP)	57
8.3	Vector Base Intensity (VBIP)	58
8.4	Dual Band Vector Based Panning (VBP Dual-Band)	58
8.5	Layer-based amplitude panning (LBAP)	58
8.6	Distance Base Angular panning (DBAP)	59
8.7	K-Nearest Neighbor (KNN)	59
8.8	Speaker-Placement Correction Amplitude (SPCAP)	61
8.9	Ambisonic Equivalent Panning (AEP)	61
8.10	Angular and PanR	62
8.11	Continuous Surround Panning (CSP)	62
8.12	Related papers	63
8.12.1	VBAP 2D/3D (Vector Base Amplitude Panning)	63
8.12.2	DBAP	63
8.12.3	AEP	63
8.12.4	PMAP	64
9	Wave Field Synthesis	65
9.1	Introduction	65
9.1.1	Availability - WFS Add-on license	65
9.2	SPAT Revolution implementation of WFS	66
9.2.1	Focus Zone	68

9.2.2	Further readings	69
9.3	WFS Settings	69
10	Scene based streams	72
10.1	Introduction to Ambisonics	72
10.2	Encoded audio	73
10.3	Order	74
10.4	Normalization, sorting and presets	75
10.5	A-Format	77
10.6	History and references	78
11	Ambisonic Transcoding	79
11.1	Default transcoding	79
11.2	Transcoding method	79
11.2.1	Projection	79
11.2.2	Regularized pseudo-inverse	80
11.2.3	Energy preserving	80
11.2.4	AllRAD	80
11.2.5	Improved AllRAD	80
11.3	Transcoding types	81
11.3.1	Basic	81
11.3.2	InPhase	81
11.3.3	MaxRe	81
11.3.4	BasicMaxRe	81
11.3.5	MaxReInPhase	81
11.3.6	InPhaseMaxRe	81
11.4	Basic rules of thumb when transcoding ambisonic to speaker	82
11.4.1	Ambisonic order and number of speakers	82
11.4.2	How to handle multiple renderings from a single ambisonic room	82
V	SPAT Environment	85
12	Files and Folders	88
12.1	Files type	88
12.2	Preferences and resources files	88
12.3	Path for Recorded Audio Files	89
12.4	Paths for python script	89
13	Home Page	91
13.1	Welcome box	92
13.1.1	Create Session	92
13.1.2	New session from empty	92

13.1.3	Open a session	92
13.2	Recent Sessions	93
13.3	Resources	94
14	Global Bars	95
14.1	Navigation bar	95
14.2	Status bar	95
14.2.1	Input stream	95
14.2.2	Hardware device	96
14.2.3	Timecode source	96
14.2.4	Timecode	96
14.2.5	Clock source	96
14.2.6	Support	96
14.3	Snapshot bar	97
14.4	Transport bar	97
14.5	Media stream bar	98
15	Setup Page	99
15.1	Setup Modules	99
15.2	The graph editor	99
15.2.1	Module selection	99
15.2.2	Drag & Drop	99
15.3	The setup wizard	102
15.4	Action menu	103
15.4.1	Remove selected	104
15.4.2	Duplicated selected	104
15.4.3	Connect selected	106
15.4.4	Disconnect selected	106
15.4.5	Disconnect selected inputs	106
15.4.6	Disconnect selected output	107
15.4.7	Disconnect between selected	107
15.4.8	Record arm	107
15.5	Routing Matrix	107
16	Items' page	109
16.1	Items type selection	109
16.2	Parameters	111
16.2.1	Common parameters	111
16.2.2	Input specific parameters	111
16.2.3	Source specific parameters	111
16.2.4	Room specific parameters	112
16.3	Navigation	113

16.4	Filtering and Group by	114
16.4.1	Filtering	114
16.4.2	Group sources by rooms	114
17	Snapshots page	116
17.1	The snapshot list	118
17.1.1	Generality	118
17.1.2	The actions	119
17.1.3	Snapshot Index	119
17.1.4	Recall options	120
17.1.5	Using global options or overriding them	120
17.1.6	Relative recall option	120
17.1.7	Changing snapshots' color	121
17.2	Version history	121
17.3	Inspector	122
17.4	Propagate through snapshots	123
17.4.1	Propagate dialog	123
18	Speaker Arrangement Editor	125
18.1	Rendering to Speakers	126
18.2	Virtual and Real Diffusion	126
18.3	Creating & editing a speaker arrangement	127
18.3.1	Speaker Arrangement section	128
18.3.2	Alignment section	128
18.3.3	Selected or all speakers section	129
18.3.4	Add speaker(s) section	129
18.3.5	The speakers list section	129
18.3.6	Panning Type Tips section	131
18.4	Understanding when a speaker arrangement, or a sound scene, is 2D or 3D	132
18.4.1	Alignment	134
18.4.2	Automatic Scaling	135
18.4.3	Speaker arrangements examples	135
18.4.4	Speaker arrangement Python file	136
18.5	Import Speaker Array	136
18.5.1	How to import a speaker configuration from	137
19	Transformations	140
19.1	Sources Transformations	140
19.2	Speaker Transformation	141
20	Signals visualization and ballistic	144
20.1	Vu-Meters	144
20.2	Nebula Spatial Spectrogram	146

20.3	FLUX:: Analyzer Workflow	147
21	Application preferences	149
21.1	Global	149
21.1.1	Allow sending error data	149
21.1.2	Reopen last session on startup	151
21.1.3	Use Metric System	151
21.1.4	Temperature	151
21.1.5	Input delay - Unit of measure	151
21.1.6	Edit Frame Rate	151
21.1.7	Engine Frame Rate	151
21.1.8	Meter Default Refresh Rate	151
21.1.9	Language	152
21.1.10	Reset to default	152
21.1.11	Import/Export Preferences	152
21.2	Hardware IO	152
21.2.1	Input/Output Devices	152
21.2.2	Sampling Rate	152
21.2.3	Block Size	154
21.2.4	Show input/output settings	154
21.3	User Interface	154
21.3.1	Anti-Alias	155
21.3.2	Setup Wire	155
21.3.3	Tooltips	155
21.3.4	Theme	155
21.3.5	Shortcuts window	155
21.4	OSC Main	155
21.4.1	OSC Enable	155
21.4.2	Enable commands log	157
21.4.3	Show invalid OSC input messages	157
21.4.4	Dump on socket change	157
21.4.5	Send current selection messages	157
21.4.6	Output rate	157
21.4.7	Export Parameters	157
21.5	OSC Connections	157
21.6	Snapshots	160
21.6.1	Recall sources name	160
21.6.2	Number of saved versions	160
21.6.3	Ask for update before recalling	160
21.7	HRTF	160
21.7.1	Default HRTF	160
21.7.2	Manage HRTFs - Ultimate license only	160

21.8	Room	160
21.8.1	Room gain	160
21.8.2	Compute LFE	163
21.8.3	Speaker test post “mute/solo”	163
21.8.4	WFS default truncation distance of Efficiency zone for linear antenna	163
21.9	Signal Generator	163
21.9.1	Gain	163
21.9.2	Signal Type	163
21.9.3	Sinus frequency	163
21.9.4	Sweep start frequency	163
21.9.5	Sweep end frequency	163
21.9.6	Sweep time	165
21.10	Engine	165
21.10.1	Automation	165
21.10.2	Audio Processing	166
21.10.3	BlackTrax RTTrPM - Ultimate license only	166
VI SPAT Setup Modules		167
	One module, one task	168
	Stream types and associated options	169
22 Input Section		170
22.1	Input modules	170
22.1.1	Name	170
22.1.2	Inputs configuration	171
22.1.3	Routing	172
22.1.4	Setup	172
22.2	Input Transcoders	173
22.2.1	Transcoding Matrix	175
22.2.2	When to Transcode Inputs?	175
22.2.3	Aggregation of input	176
23 Sources & Rooms Module		178
23.1	Source module	178
23.1.1	Name	178
23.1.2	IO Configuration	178
23.1.3	Tracking	179
23.1.4	Gain	179
23.1.5	Levels	179
23.2	Room module	179
23.2.1	Name	180
23.2.2	IO Configuration	181

23.2.3	Tracking	181
23.2.4	Gain	181
23.2.5	Levels	181
24	Master Section	182
24.1	Sum modules	182
24.2	Master Transcoder modules	183
24.3	Master modules	183
24.4	Binaural monitoring modules	183
24.5	Output modules	183
25	Modules Channel Count	186
25.1	Mono Input	186
25.2	Two Channel	186
25.3	Four Channel Input	186
25.4	Multi-Channel Based Input	188
26	Session compatibility and modules (de)activation	190
26.1	Create and edit a session in the Essential version	190
26.2	Opening an Ultimate session with the Essential version	191
26.3	Opening an Essential session with the Ultimate version	191
26.4	Check Essential compatibility in SPAT Revolution Ultimate	191
26.5	Essential Compatibility Mode for Ultimate version	192
VII	SPAT Mixing Engine	194
27	Source	196
27.1	Basic interactions	197
27.1.1	Reset to defaults	197
27.1.2	Preset Memories	197
27.2	Smart Property Filter	198
27.3	Barycentric Groups	199
27.3.1	Multiple Source Selection	200
27.4	Groups of sources	201
28	Source Parameters	203
28.1	Room specific gain	203
28.2	Perceptual Factors	204
28.2.1	Presence	205
28.2.2	Warmth	205
28.2.3	Brillance	205
28.2.4	Room Presence	205
28.2.5	Running Reverberance	205

28.2.6	Envelopment	205
28.3	Reverb Options	206
28.3.1	Reverb Enabled	206
28.3.2	Early / Cluster / Tail	206
28.3.3	PanRev	207
28.3.4	Early Width	207
28.4	Axis / Omni Filters	207
28.4.1	Spectral Omni	208
28.4.2	Spectral Axis	208
28.5	Radiation	209
28.5.1	Relative Direction	209
28.5.2	Distance	209
28.5.3	Azimuth	210
28.5.4	Elevation	210
28.5.5	Yaw	210
28.5.6	Pitch	210
28.5.7	Aperture	210
28.6	Send LFE	211
28.7	Barycentric	212
28.7.1	Rotation XYZ	212
28.7.2	Scale	212
28.7.3	Relative Direction	213
28.8	Options	213
28.8.1	Doppler	213
28.8.2	Drop Factor and Drop Log	214
28.8.3	Air Absorption	214
28.8.4	Radius	214
28.8.5	XY translation mode	215
28.8.6	Z translation mode	215
28.9	Spreading	215
28.9.1	Spread Factor	216
28.9.2	Nearest Neighbors	216
29	Room	217
29.1	Room Graphic Engine	218
29.2	Room top menu	219
29.2.1	Display output drop-down	221
29.3	Room output parameters	221
29.3.1	Output list	221
29.3.2	Listener	222
29.3.3	Protection Zone	223
29.3.4	Efficiency Zone	224
29.3.5	Scaling	225

29.3.6	Background Image	226
30	Artificial Reverberation	227
31	Reverb Parameters	229
31.1	Reverb Design Presets	229
31.2	Reverb General	230
31.2.1	Reverb enable	230
31.2.2	Tail	230
31.2.3	Reverb Density	230
31.2.4	Size	231
31.2.5	Reverb Start	231
31.3	Perceptual Factors	232
31.3.1	Reverberance	232
31.3.2	Heaviness	232
31.3.3	Liveness	233
31.4	Room Response Parameters	233
31.4.1	Early Min	234
31.4.2	Early Max	234
31.4.3	Early Dist	234
31.4.4	Early Shape	234
31.4.5	Cluster Min	234
31.4.6	Cluster Max	234
31.4.7	Cluster Distribution	234
31.5	Options	235
31.5.1	Infinite	235
31.5.2	Air Absorb	235
31.5.3	Modal Density	236
31.5.4	Abs. RollOff	236
31.6	Crossover	236
31.6.1	Crossover L	237
31.6.2	Crossover H	237
32	Understanding the mixing zones	238
32.1	The protection zone	239
32.2	The presence of a source	241
32.3	The efficiency zone	242
32.3.1	Behaviour of sources outside the efficiency zone	243
32.3.2	Elevation clamping	247
32.4	Limit cases	248

VIII Ecosystem & integration	249
33 Using Hardware I/O	252
33.1 Single Hardware Workflow	252
33.2 Distributed Hardware Workflow	253
34 Open Sound Control	254
34.1 Introduction to Generic OSC	254
34.2 OSC Connections Menu	255
34.3 OSC Transform	257
35 SPAT PI : Automation	259
35.1 DAW Automation - Manual setup	259
36 SPAT PI : Local audio path	263
36.1 Setting up a local audio path connection	264
36.2 Plug-ins parameters	265
36.3 Implications of using the Local Audio Path	265
36.3.1 DAW routing priority	265
36.3.2 Local audio path and delay compensation	266
36.3.3 Dealing with DAW CPU optimizations	266
36.4 Alternative to local audio path	267
36.4.1 Pros and Cons of the Local Audio Path	267
36.4.2 Virtual Loopback Devices	267
37 Snapshot	269
37.1 Create a snapshot	270
37.2 Recall a snapshot	270
37.3 Snapshots handling	271
38 Handle snapshots with OSC	272
39 ADM-OSC	274
39.1 ADM-OSC in SPAT Revolution	274
39.2 Third party example	277
IX Third Party Integration	281
40 Cockos Reaper	283
40.1 Templates	283
40.2 ReaVolution	284
40.3 Setting Up Sync	284
40.4 Setting Up Tracks	284

40.5	Setting up SPAT SEND	287
40.6	Accessing the latency of the FX chain	288
40.7	Optimization	288
40.8	Source/Object Automation	295
40.9	Setting up controllers and LFO	295
40.10	Setting up SPAT RETURN	297
40.11	Recording Immersive creation.	299
41	ReaVolution	302
41.1	Credits & acknowledgements	302
41.2	Introduction	302
41.3	Table of content	303
41.4	Installation	303
41.5	Presentation	304
41.5.1	Windows arrangement and menus	304
41.5.2	SPAT Revolution library	306
41.5.3	Audio Stream library	306
41.5.4	Various library	306
41.6	Use cases	307
41.6.1	Creating sessions	307
41.6.2	Dealing with routing	308
41.6.3	Sending tracks to SPAT Revolution	310
41.6.4	Receiving audio from SPAT Revolution	310
41.6.5	Editing audio	312
41.6.6	Three and four points editing	317
42	Avid Pro Tools (2022 Update)	319
42.1	Workflow	319
42.2	2022 Technical Articles	319
42.3	Templates	320
42.4	Setting Up Sync when using Local Audio Path (LAP)	322
42.4.1	Latency compensation	324
42.4.2	Setting Up Object Tracks	324
42.4.3	The SPATSync (aka dummy bus) and why	327
42.4.4	Pro Tools license versions	328
42.5	SPAT Source/Object Automation	328
42.5.1	Source/Object Index, moving from LAP to OSC	331
43	Ableton Live	332
43.1	Templates	332
43.2	DAW optimization	333
43.3	Ableton Live routing	335
43.3.1	Audio source tracks	335

43.3.2	Object tracks “Send to SPAT”	335
43.3.3	Return tracks	335
43.4	Recording the immersive creation	339
43.5	Higher number of channels	339
44	Ableton Live Tools for SPAT Revolution	342
44.1	Introduction	342
44.1.1	Why use Ableton Live with SPAT Revolution?	343
44.1.2	Why use SPAT Revolution at all?	343
44.2	Demo	343
44.3	Installation	344
44.3.1	Tools	345
44.3.2	Virtual Audio Driver	346
44.3.3	Useful addition	347
44.4	Optimization	348
44.5	Setup	350
44.5.1	Ableton Live	350
44.5.2	SPAT Revolution	358
44.5.3	What is OSC?	360
44.6	Devices Description	361
44.6.1	SEND	361
44.6.2	ROOM	363
44.6.3	RETURN	365
44.6.4	Automator	366
44.7	Workflow	368
44.8	Printing an Immersive Mix	371
44.9	From Single Machine to Dual Computer Setup	373
44.10	Uninstall	374
44.10.1	With the FLUX Center	374
44.10.2	Manually	374
44.11	Credit	374
44.12	Troubleshooting / Q&A	375
44.12.1	Windows?	375
44.12.2	OSC issues? (Automation not going to SPAT)	375
44.12.3	SPAT Revolution Essential license?	379
44.12.4	Demo Sessions?	379
45	Steinberg Nuendo	380
45.1	Templates	380
45.2	Nuendo and External OSC rendering	380
45.3	Using ADM-OSC in Nuendo / Use cases	382
45.3.1	System schematics - Nuendo and SPAT Revolution	382
45.3.2	Creating an up to 64 objects session in Nuendo	383

45.4 Nuendo v11 - Setting SPAT Revolution as external renderer	386
45.5 Importing ADM File	386
45.6 Nuendo audio connections configured for object	389
45.7 Nuendo object position to SPAT Revolution source objects.	389
45.8 Setting up Nuendo OSC Object Position Tracking	389
45.9 SPAT Revolution ADM-OSC settings for Nuendo	395
45.9.1 ADM-OSC Input setup	395
45.9.2 ADM-OSC XYZ Output to Nuendo	399
46 Bitwig Studio	400
47 Figure53 QLab	406
47.1 QLab 5	406
47.2 QLab 5 Integration template	407
47.3 QLab 4 Integration template (V4 validated)	407
47.4 Generic QLab Templates (V4, compliant with V5)	407
47.5 Configuring QLab -> SPAT for messages via network cues	408
48 Avid VENUE S6L	413
48.1 Configuring OSC	416
48.2 Setting the preferences of the SPAT Revolution Application	418
48.2.1 SPAT OSC Connections preferences	419
48.3 Configuring Audio Connections	422
48.4 Templates	423
49 Setting up DiGiCo console with SPAT Revolution	425
49.1 Configuration and Templates	425
49.2 Audio routing	432
49.3 Useful tips for integration	432
50 Solid State Logic SSL Live Consoles	436
50.1 Template	436
50.2 Setting up SSL Live console	436
50.3 External control fader and switches	436
50.4 Setting up SPAT Revolution	438
51 SPAT Revolution Remote	442
51.1 Installation	442
51.2 Configuration	443
51.3 Autostart	445
51.4 SPAT Revolution configuration	445
51.5 Reaching the Remote interface from the browser	448
51.5.1 Adding the SPAT Revolution Remote as a shortcut(Ipad)	449
51.6 SPAT Revolution Remote page	449

51.7 Release Notes	449
51.7.1 22.10.0.50212	449
52 BlackTrax integration	453
Appendices	460
A SPAT Revolution - Credits	460
A.1 Version 22.09	460
A.2 User Guide and Documentation	460
A.3 Software development	460
A.3.1 Project Manager and Designer:	460
A.3.2 Application Development	460
A.3.3 Flux:: DSP Design and Development	460
A.3.4 IRCAM “Spat~” Design of digital signal processing algorithms and im- plementation in Max	460
A.3.5 IRCAM “Spat~” Objective and perceptual characterization of room acoustical quality	461
A.3.6 IRCAM “Spat~” C++ dsp code	461
A.3.7 Flux:: Framework development	461
A.4 Graphic design	461
A.4.1 Flux:: Framework graphic engine	461
A.5 Translation	461
A.5.1 And	462
A.5.2 FLUX:: Special Thanks to	462
A.6 FLUX::	462
A.7 Additional libs	462
A.7.1 Additional Python modules	463
A.8 Open source libraries	465
A.8.1 RTTrPM SDK (Blacktraxx)	465
A.8.2 ThorVG (replaces crossed-out libraries above) - MIT	465
A.8.3 r8brain free - MIT	466
A.8.4 ooura fft from lmdsp - MIT (or MIT-like)	466
A.8.5 LibJpeg - Custom (BSD-like)	466
A.8.6 libpng - PNG Reference Library License version 2	466
A.8.7 Freetype 2 - FreeType License	466
A.8.8 Zlib - zlib	466
A.8.9 bzip2 - Modified zlib	466
A.8.10 Boost - BSL-1.0	466
A.8.11 mimalloc - MIT	467
A.8.12 ni-media - MIT	467
A.8.13 pfft	467

A.8.14	pybind11	467
A.8.15	rtaudio	467
A.8.16	rtmidi	467
A.8.17	concurrentqueue - Simplified BSD	467
A.8.18	readerwriterqueue - Simplified BSD	468
A.8.19	assimp	469
A.8.20	enet - MIT	469
A.8.21	fmt	469
A.8.22	functions (Erik Rigtorp) - MIT	470
A.8.23	inplace_function - BSL-1.0	470
A.8.24	gtest - "BSD 3-Clause "New" or "Revised" "	470
A.8.25	jsoncpp - MIT	470
A.8.26	libcurl - MIT like	471
A.8.27	magic_enum - MIT	471
A.8.28	nanobench - MIT	472
A.8.29	nlohmann_json - MIT	472
A.8.30	openddl - MIT	473
A.8.31	openvr - "BSD 3-Clause "New" or "Revised" "	473
A.8.32	poly2tri - "BSD 3-Clause "New" or "Revised" "	474
A.8.33	pugixml - MIT	475
A.8.34	rapidjson - MIT	475
A.8.35	shared_recursive_mutex - MIT	476
A.8.36	sentry-native - MIT	476
A.8.37	sqlite3 - Public Domain	477
A.8.38	sqlite_orm - MIT	477
A.8.39	stb - MIT	477
A.8.40	utfcpp - BSL-1.0	478
A.8.41	unqlite - BSD 2-clause	478
A.8.42	argh (Andrey Mikhaylenko) - LGPLv3	479
A.8.43	chardet (Mark Pilgrim, Daniel Blanchard) - LGPL v2.1	479
A.8.44	charset_normalizer - MIT	479
A.8.45	docopt (Vladimir Keleshev) - MIT	480
A.8.46	dotenv - "BSD 3-Clause"New" or "Revised" "	480
A.9	Open source python libraries	481
A.9.1	ecdsa (Peter Pearson) -	481
A.9.2	environs (Steven Loria) - MIT	481
A.9.3	ffmpeg-python - Apache 2	482
A.9.4	ftputil (Stefan Schwarzer) - BSD	482
A.9.5	numpy - "BSD 3-Clause"New" or "Revised" "	483
A.9.6	pendulum - MIT	484
A.9.7	psutil (Giampaolo Rodola) - "BSD 3-Clause"New" or "Revised" "	484
A.9.8	watchdog - Apache 2	485
A.9.9	tgcrypto - LGPL v3	485

A.9.10	idna - BSD 3	486
A.9.11	marshmallow - MIT	486
A.9.12	packaging - BSD	487
A.9.13	paramiko (Jeff Forcier) - LGPL 2.1	488
A.9.14	pathtools - MIT	488
A.9.15	ptvsd (Microsoft) - MIT	488
A.9.16	pyga (Arun K.R.) - BSD	489
A.9.17	pylightxl - MIT	489
A.9.18	pyarsing - MIT	490
A.9.19	pytzdata - MIT	490
A.9.20	winshell (Tim Golden) - MIT	491
A.9.21	six (Copyright (c) 2010-2020 Benjamin Peterson - MIT	491
A.9.22	Request (Kenneth Reitz) - Apache 2	492
A.9.23	sentry_sdk (Copyright (c) 2018 Sentry) - MIT	492
A.9.24	urllib3 - MIT	492
A.9.25	certifi (Kenneth Reitz) - MPL-2.0	493
A.9.26	anytree (c0fec0de) - Apache 2	493
A.9.27	boltons - BSD	493
A.9.28	msgpack (Inada Naoki) - Apache 2	494
A.9.29	dateutil (Gustavo Niemeyer, Paul Ganssle) - Apache 2	494
A.9.30	PyYAML (Kirill Simonov) - MIT	495
B	SPAT Revolution specifications	496
B.1	Availability	496
B.2	Processing	496
B.3	OS Compatibility	496
B.4	Hardware Recommendations and requirements	496
B.4.1	Single Computer (Creative Station with DAW and SPAT Revolution)	496
B.4.2	Hardware for Live Productions (Dedicated SPAT Revolution Computer)	497
B.4.3	Software License Requirements	498
B.5	Certified configurations (SPAT Revolution v.22.02.50151)	498
B.5.1	48000 kHz	499
B.5.2	96000 kHz	500
C	Troubleshooting	501
C.1	Sync issues with LAP	501
C.2	SPAT Send and Return DAW	502
C.3	Clearing Shared Memory	506
C.4	Performance issues	506
C.5	Display Performance (CPU) measurements	507
C.6	Engine Parallel Processing profile	507

D Appendix C - OSC Table	510
D.1 OSC Grammar	510
D.2 OSC Output Options	510
X SPAT Revolution - Release notes	511
Build 23.11.0.50341	513
New Features	513
Improvements	513
Bugs	513
Known Issues	515
Build 23.08.0.50318	516
New Features	516
Improvements	516
Bugs	517
Known Issues	519
Build 22.10.0.50222	520
Bugs fixes	521
Known Issues	522
Build 22.10.0.50219	524
Improvement	524
Bugs fixes	524
Known Issues	524
Build 22.10.0.50213	526
Improvements	526
Bugs	526
Known Issues	526
Build 22.02.0.50206	528
Bug fixed	528
Known Issues	529
Build 22.02.0.50200	530
New Features	530
Improvements	530
Bugs	531
Known Issues	534

Build 22.02.0.50151	535
Improvements	535
Bugs fixes	535
Known Issues	535
Most important	535
Important	536
Build 22.02.0.50148	537
New features	537
Improvements	537
Spatial Processing	537
Speaker Management	538
OSC	538
Setup	538
3D View	539
Workflow	539
Core optimization	539
Various	539
Bugs fixes	539
Spatial Processing	539
OSC	540
3D View	540
Binaural	540
Setup	540
Speaker Editor	541
Workflow	541
Various	541
UI	541
Plugins	542
Known Issues	542
Most important	542
Important	542
Build 21.4.0.50030	543
Improvements	543
Bug fixes	543
Build 21.4.0.50020	544
SPAT Plug-ins suite	544
New feature	544
Bug Fixes	544
SPAT Revolution new features	544

Improvements	545
User Interface	545
Core behaviors	545
Bug fixes	545
Build 20.12.0.49943	547
Improvement	547
Build 20.12.0.49930	548
Bug Fixes	548
Build 20.12.0.49890	549
SPAT Revolution new features	549
OSC - Many OSC optimizations in the core	550
Major optimizations	550
Other improvements	551
Bug Fixes	551
Build 20.12.0.49880	553
SPAT Plug-ins suite	553
Bug fixes	553
SPAT Revolution bug fixes	553
Core	553
UI	554
Various	554
Known Issues	554

FLUX:: Immersive SPAT Revolution 23.08



A User Guide

Written by Cristian Vogel, Jean-Loup Pecquais, Hugo Larin and Nicolas Erard

with contributions from: G ael Martinet, Vincent Carlier, Florie-Anne Lafaye, Thibaut Carpentier, Anders Tveit and the IRCAM team.

Legal Information

Fourth edition of FLUX:: *SPAT Revolution* User Guide published in July 2023.

All writing is protected under copyright   2024 Cristian Vogel / FLUX::.

FLUX:: is a trademark of HARMAN International. All trademarks and the logos are copyright protected.

FLUX:: SPAT Revolution is published by
HARMAN International
8500 Balboa Blvd
Northridge, CA 91329
United States

Spat Revolution Copyright © 2024, HARMAN International. All Rights Reserved.

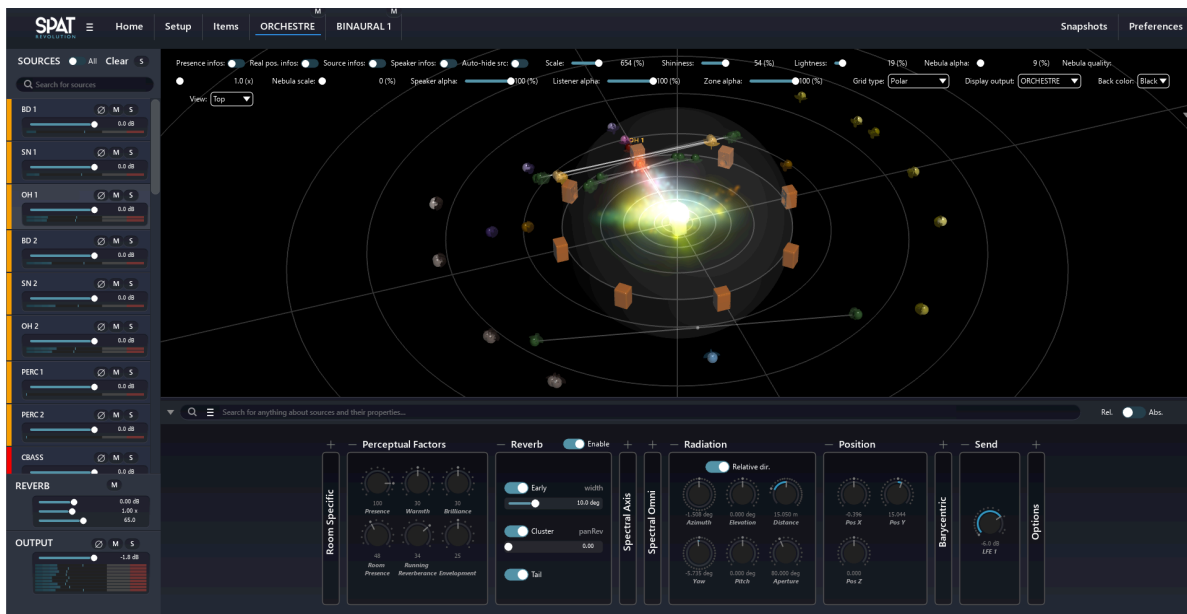
Part I

Welcome to SPAT Revolution

[Product Page](#) | [Shop Page](#)

First of all, thank you for acquiring *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 spatialized audio production powerful and intuitive for beginners and professionals alike. Our real-time audio environment provides easy access to some of the most advanced spatialization algorithms currently available, at the very best sound quality currently possible.

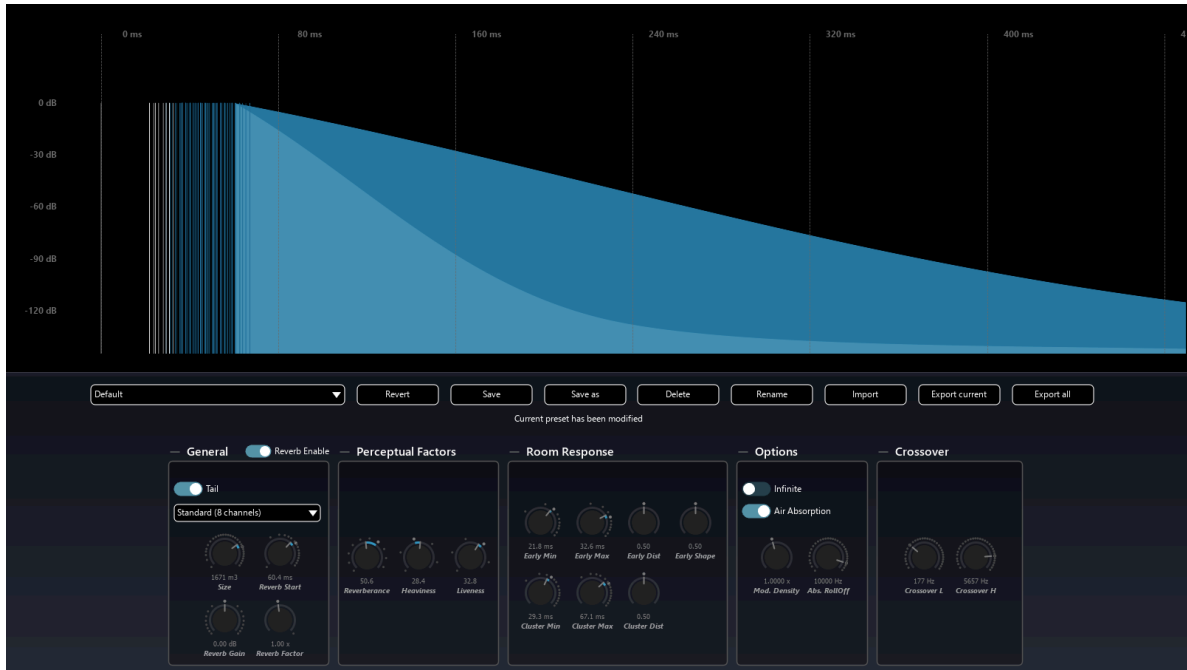
‘*SPAT*’ is short for *Spatialisateur* in French. It is a real-time audio library that allows composers, sound artists, performers and sound engineers to control the localization of sound sources in virtual and real 3D auditory spaces.



SPAT Revolution wraps the ‘Spat’ processing library in a luxurious and characteristic graphic environment to help visualize many aspects of a spatial audio composition in real time. This graphic interface allows sound mixes to be composed as interactive spatial models existing in *Virtual Room*. SPAT contains a powerful multichannel reverberation engine that can be applied to design and add a sense of auditory space in studio mixes and real time on location.

Note

Artificial Reverberation Editor



SPAT Revolution maintains the highest audio quality throughout the entire signal flow.

Part II

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:: SE 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. 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.

Part III

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 strongly suggest spending the time to read through this guide before starting on your first major production and keeping it on hand during the process. Expert support and advice can also be sought from the [FLUX Immersive FB user group](#) and the [Web-based Knowledge base](#).

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

Getting started

In the following section, we will explain how to install the application using the FLUX::Centre and how to redeem and activate your licenses.

We will also explain all the differences between the **Ultimate** and **Essential** licenses.

Furthermore, if you are searching for a quick start guide, check this document : [quick start guide](#).

1 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 at 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 production mixing
- scientific research and development
- sound design for film, music and theatre
- environmental sound

In the contexts of live production 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 (show controllers, trajectory applications, tracking systems, tablet, smartphone, joystick, gestural sensors, etc.) through OSC and RTTrPM interfacing protocols. *RTTrPM is supported by the SPAT Revolution Ultimate license only.*

In the contexts of studio mixing and post-production, a virtual source/object can receive its audio from individual channels on a mixing desk or DAW 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 spatialized 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 are 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.

i Note

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

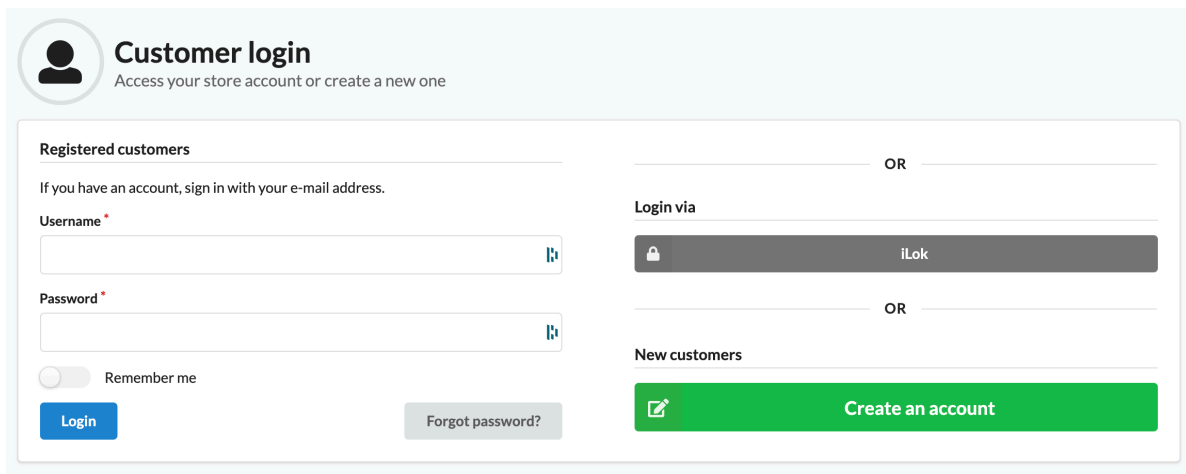
2 Installation and Activation

2.1 How to Install SPAT Revolution?

FOUR (4) STEPS:

1. [Create an account](#) on [FLUX website](#)
2. License code redeem
3. Software license activation
4. Download and installation

2.2 Create and account



The screenshot shows the 'Customer login' page. At the top left is a user icon and the text 'Customer login' with a subtitle 'Access your store account or create a new one'. Below this, there are two main sections: 'Registered customers' and 'New customers'. The 'Registered customers' section includes a text input for 'Username' and a password input for 'Password', both with red asterisks indicating required fields. There is a 'Remember me' checkbox and a 'Login' button. A 'Forgot password?' link is also present. The 'New customers' section features a large green 'Create an account' button with a plus icon. In the center, there is an 'OR' separator and a 'Login via' section with an 'iLok' button.

[Create an account](#) on the FLUX website, clicking on the previous link.

2.3 Redeem a License Code from the activation code

FLUX:: uses the iLok license management system to deliver software licenses to users. If you have received an activation code (such as from a dealer purchase or [30-day trial](#)), you can use the Redeem License Code window to activate your license.

Home / My account / Licence code

Your account

- Dashboard
- Personal information
- Change password
- Address book
- Order history
- Subscriptions
- Manage OAuth links
- iLok licenses
- iLok license - trial request
- Licence code

Licence code

Redeem a licence code

iLok User ID *

Licence code *

[+ Redeem it !](#)

Visit our [License Code Activation page](#) for more information.

2.4 iLok User Account

To activate licenses:



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

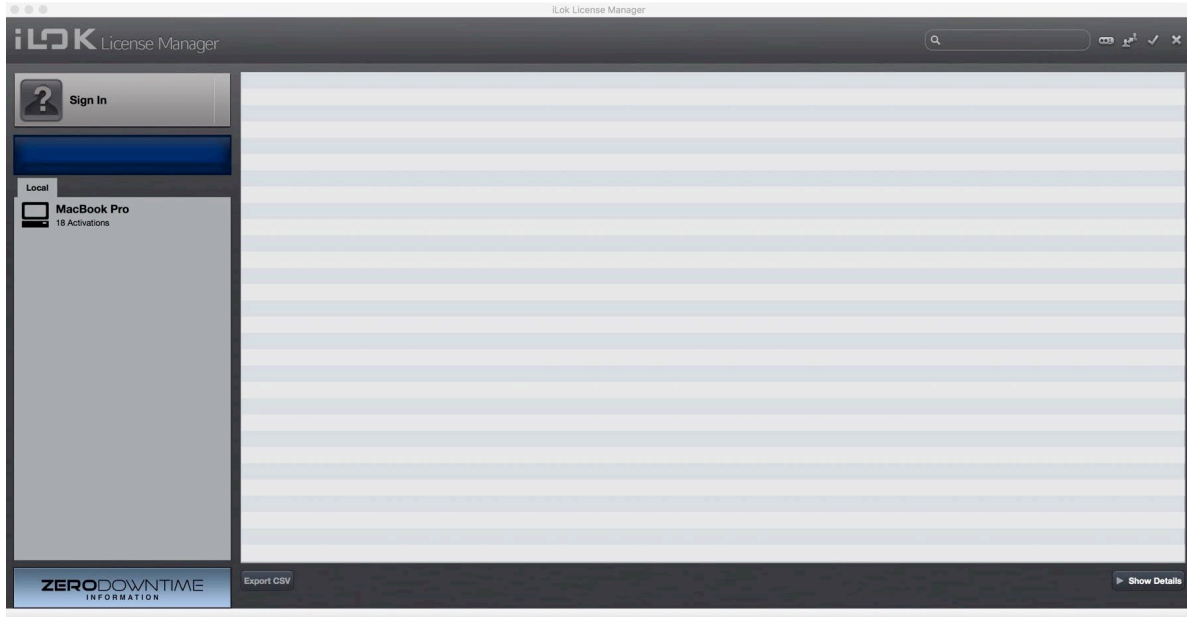
FLUX:: uses the iLok license management system to deliver software licenses to users. If you don't have an iLok account yet, please create a free iLok account at <http://www.ilok.com> and download the iLok license manager. *SPAT Revolution Essential* license comes with one (1) activation whereas the *SPAT Revolution Ultimate* Bundle includes two (2) activation linked to your user account. Having two activations gives you the possibility of a fixed license on one particular machine and a portable license on an iLok USB key if you own one. Other examples are for main and backup SPAT Engine running on a live production or In-studio and on-the-road scenarios.

i Note

Cloud license is currently not supported.

2.5 iLok License Manager

If you have redeemed your software license or completed your purchase process, your license will automatically be delivered to your iLok account.



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 is correctly installed, you can start the license manager software and log in to your iLok user account.

2.6 Transferring license

iLok License Manager

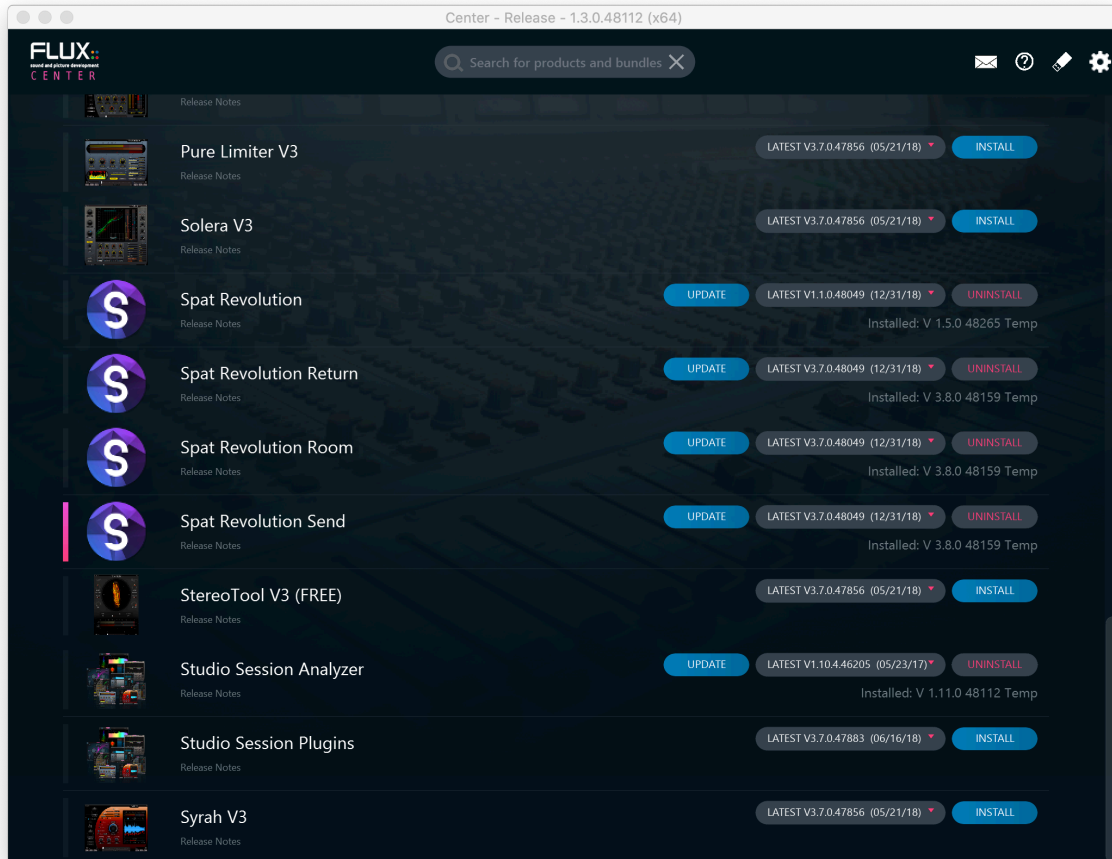
Pressing on the sign-in button will allow you to connect to your account. After Logging in, you are now ready to transfer any licenses to a computer or to any iLok USB key if you happen to have one. The process of transferring a license is as simple as dragging the license from the Available tab to your Local Computer (*or iLok key*) on the left side. *SPAT Revolution Essential* license is a single license where *SPAT Revolution Ultimate* is a bundle containing three (3) licenses, **Essential**, **Ultimate** and **Legacy**.

Simply drag your license to your Local Computer or on an iLok USB key. You are now set!

i Note

If you require further information about iLok and managing licenses please refer to iLok.com website.

2.7 FLUX:: Center



The 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 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](#).

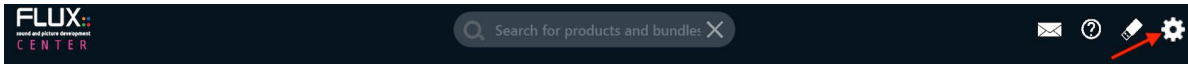
On this page, you will find a macOS, a Windows 64 bits. As well are provided legacy versions for older operating systems. After downloading and installing, you can open the FLUX::

Center applications to begin the process of installing the *SPAT Revolution* software.

Warning

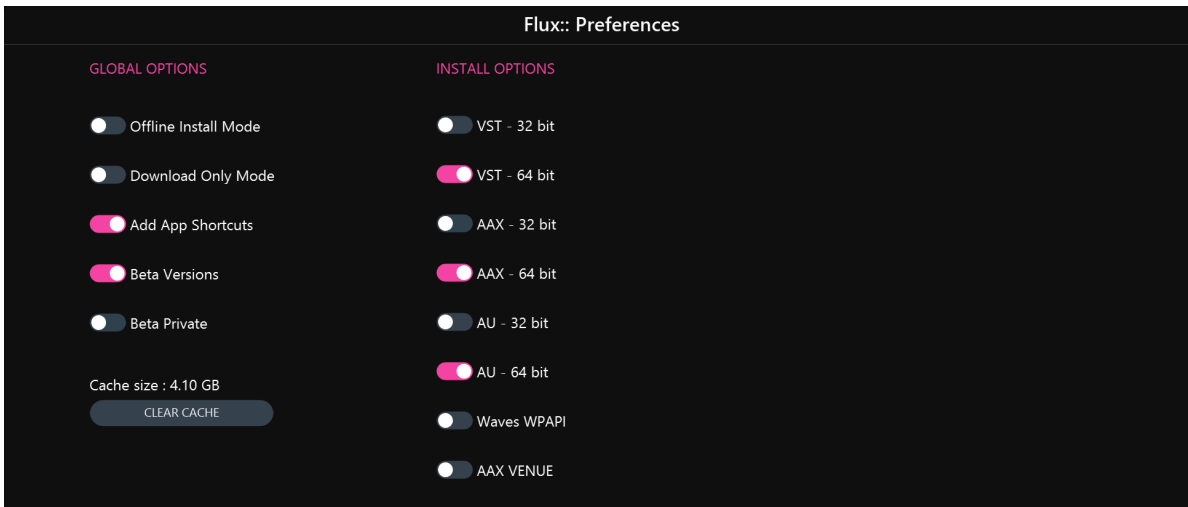
Authentication is required at the launch of FLUX:: Center. This is the login details of your FLUX shop account which allows you to see only your products licensed for (temporary or permanent).

2.8 Center Preferences



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. 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 at the top right of the header area.

2.9 Center Preferences and Options



i Note

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

This preference 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.

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 installers. Note that these versions start with a B whereas official releases start with a V.

3 Ultimate and Essential licenses

SPAT Revolution comes in different flavors: an **Essential** and an **Ultimate** licenses. Both share the same audio engine and powerful 3D audio capabilities.

The **Essential** license aims to be a limited version of **Ultimate**: the same workflow but limited capacities at a reduced price.

3.1 Ultimate and Essential: what's common?

The *SPAT Revolution* software supports the same installer (binary) for both license options and has the common features listed below:

- An easy-to-understand setup page, showing a clear representation of the signal path.
- A powerful mixing environment, using a 3D view and perceptive factor parameters.
- An oriented-object audio engine allowing the user to render either channel-based, Higher Order Ambisonic (HOA), or binaural audio streams.
- Support for mono, stereo, and multichannel inputs streams.
- A flexible speaker array editor allowing going beyond the usual standards.
- A snapshot system to easily recall scenes on the fly.
- An audio pipe technology allowing receiving and sending audio from all the major DAW to and from *SPAT Revolution*.
- An exhaustive list of OSC commands to allow a deep remote control of *SPAT Revolution*.

3.2 Ultimate and Essential: differences

The main limitations of the **Essential** license are:

- The number of cumulated input channels is limited to 32.
- The number of cumulated output channels is limited to 18.
- The number of rooms is limited to 1: no simultaneous rendering.
- HOA order is limited to 3rd order.
- Tracking data cannot be modified.
- RTTrp is not allowed.

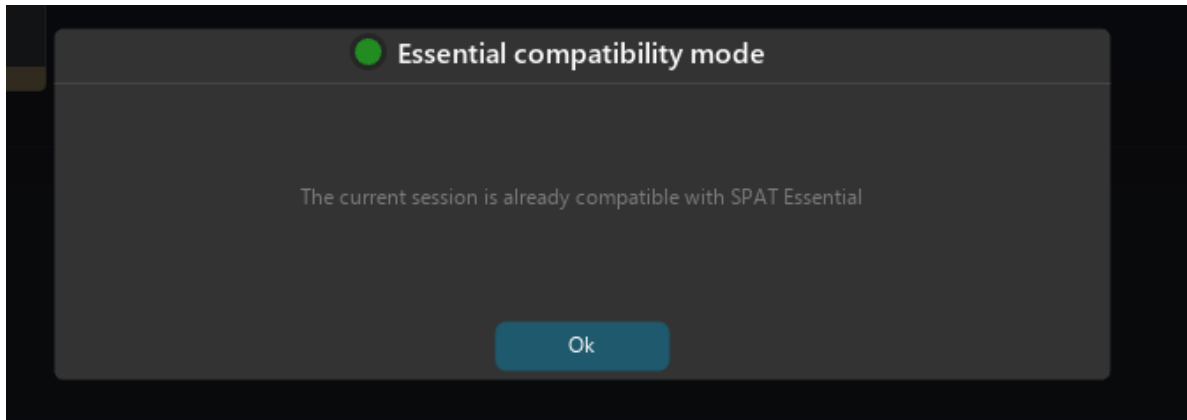
Complete specification is available [here](#).

3.3 Ultimate and Essential sessions compatibility

When creating or opening a session that contains elements non-compatible with the **Essential** license, those elements are simply deactivated (not processed). Thus, an **Ultimate** session can be opened with an **Essential** license and vice-versa. See [Modules (de)activation and sessions compatibility -@module-de-activation].

3.3.1 Check Essential compatibility

In the top bar menu, click on **File > Check Essential Compatibility** to check if the current session is compatible with the SPAT Essential restrictions.



If the session is not compatible, it can be opened with an **Essential** license, the restrictions due to license limitations will automatically deactivate the non-authorized objects.

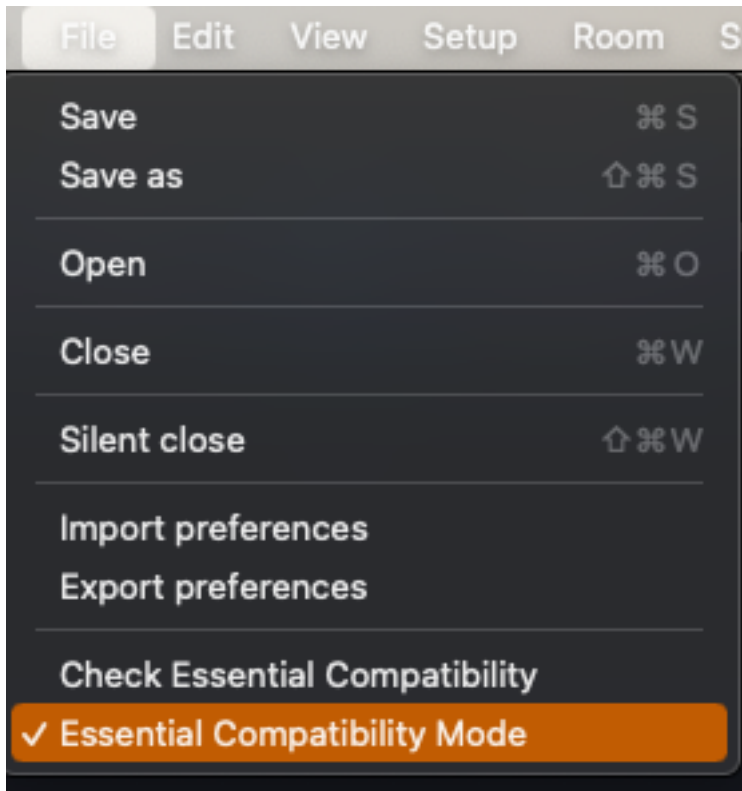
i Note

If a session contains elements non-compatible with an **Essential** license, they can be manually deactivated for the session to fit the **Essential** license restrictions. If all the non-authorized elements are inactive, the session is considered as compatible with **Essential** license.

See [Modules \(de\)activation 26](#) for more information about automatic and manual (de)activation.

3.3.2 Essential compatibility mode in Ultimate

The **Ultimate** license offers an *Essential Compatibility Mode*, for switching between **Essential** and **Ultimate** behaviors without any change in your license authorization.



Part IV

Spatialization Technology

In this section, we will discuss the various spatialization tool used by *SPAT Revolution*.

Here are some straight-to-the-point key points:

- *SPAT Revolution* is an object-based audio engine. The mixes you will create will consist of sets of data for each sound object. These data sets will then be decoded for a certain type of spatialization technique.

i Note

With *SPAT Revolution Ultimate* license, several renders can be done at the same time (multi-room).

- Thanks to the object-based engine, you can create sound stages in all spatialized audio formats possible: channel-based, ambisonics (up to seventh order), binaural and transaural. If these names do not ring a bell to you, do not fear, we will cover these different techniques later in this section.
- On top of the object-based engine, *SPAT Revolution* also covers the acoustic simulation side of things. The built-in reverberation will adapt to your mixes, and, more importantly, to your desired output format.
- *SPAT Revolution* aims at many use cases, many of them being for live performances. It was designed to not add any more latency than necessary on the signal path.

Let's start with an important notion of spatial audio: the listener's position.

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 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 coordinates 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 Revolution*, the *Listener Position* is 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 too. 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 *Listener Position*. The internal distance perceptual cues, such as air absorption, doppler and gain drops are all calculated in relation to the *Listener Position*.

i Note

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

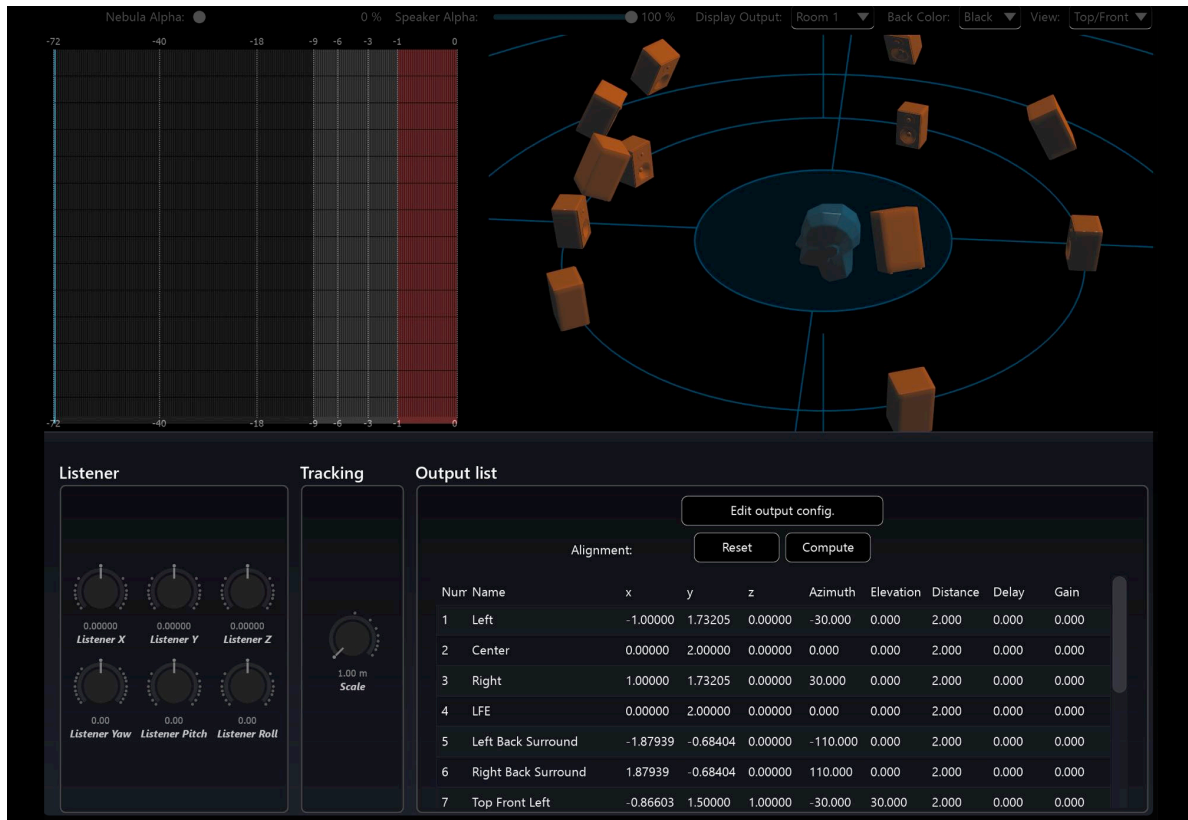
It is worth noting that some panning algorithms in *SPAT Revolution* that are **not** *Sweet Spot* dependent. These are more suitable for use on arbitrarily positioned speaker distributions. We will get to them shortly.

⚠ Warning

The dummy head is only visible for *Sweet Spot* dependent panning types.

i Note

In a *SPAT Revolution* room, the Listener Position can be moved.



4.1 Listener position and head tracking

In certain advanced situations which might combine position tracking systems with real-time binaural audio, it is even possible to transform the *Listener Position* in *SPAT Revolution*. 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. Given the camera position perspective to the listener is as well possible by mapping the camera position to the *Listener Position* of *SPAT Revolution*.

5 Binaural

5.1 Introduction

The word “binaural” 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 (like ‘**Kemar**’ or ‘**KU100**’) and when played back over headphones such recordings can produce an authentic immersive auditory experience with enhanced spatial aspects. Relatively recent progress in signal processing technology has made it possible to synthesize binaural signals without the need for microphones.

Using binaural synthesis, a sound can be arbitrarily positioned around a listener synthesizing the sensory experience of an extended spatialization. Like some other two-channel formats such as Mid-Side Stereo, binaural-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.

Warning

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.*

5.2 HRTF

HRTF is an abbreviation for *Head Related Transfer Function*. This 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 localize sounds around us and is unique to every individual. HRTF is different between the two ears, so we always talk about pairs of HRTFs.

When synthesizing binaural monitoring, a perfect result could be attained by rendering through the exact HRTFs that match the body filtering effect of an individual. In practice this is not easily done, so *SPAT Revolution* offers many choices of pre-analyzed HRTFs profiles that you can apply for monitoring and encoding binaural audio. You can manage the selection of HRTFs profiles in the *SPAT Revolution* Preferences where you will find several profiles including the

option to load your own HRTFs. The default HRTF is the Kemar dummy head model, which is often used as an all-around generic head and shoulder filter.

5.3 HRTF Profiles

The included HRTFs profiles in *SPAT Revolution* are taken from several large-scale laboratory research projects where measurements were taken on many individuals*. 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 HRTFs profile that suits them best. This can be fun! If you are not comfortable listening through someone else ears—which is understandable—you could look into creating a personalized HRTFs profile from your own head and upper torso measurements. There already exist a number of services that can create HRTFs profiles taken from laboratory measurements. If you decide to do this, for yourself or someone else, then you can add the personalized profile to the list in the HRTFs Manager. You can import any HRTFs in SOFA format to the *SPAT Revolution* binaural encoding list, making *SPAT Revolution* a very flexible solution for binaural monitoring and rendering.

Warning

An imported HRTFs profile should be in SOFA format and should match the sample rate of your project. It is preferable to use a “SimpleFreeFieldSOS” IIR type of HRTFs.

* The profiles come from the “LISTEN”, “CROSSMOD” and “BiLi” databases.

5.4 Binaural algorithms

5.4.1 Standard binaural mode

This mode uses the selected HRTFs profile to recreate the sound field.

5.4.2 Advanced algorithms

Near-field binaural

HRTFs are generally measured at one or two meters from the listener. In reality, the HRTFs are changing with the distance between the source and the listener, especially when sources are close.

The Near Field Binaural tends to recreate close HRTFs, with an additional filter set applied.

Spherical head model

This binaural synthesis simulates the head like a rigid sphere, instead of using HRTFs.

As this model does not use HRTFs, this synthesis consumes less CPU. However, the localization is clearly less precise.

Snowman model

This synthesis is the improvement of the Spherical Head Model. Besides the head reflections, torso ones are simulated, like two spheres one on the top of the other. Hence, its name, “snowman model.”

5.4.3 Head scale parameter

The parameter “head scale”, available for the four binaural algorithms, allows adapting the head size to the listener. This will adjust the interaural time and level differences. It is available on the room parameter on the **Setup** page, but also on the output panel of the room.

5.5 Binaural Monitoring Module

In the *Setup* page 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 spatialization 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 Monitor row towards the bottom of the *Setup* page graph. The module is very simple to use. It will automatically detect the type of channel-based audio stream you connect to it.

The binaural monitoring module works by virtualizing 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 center 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 center point either.

To listen to the binaural stream on headphones, you should select the HRTFs profile 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.

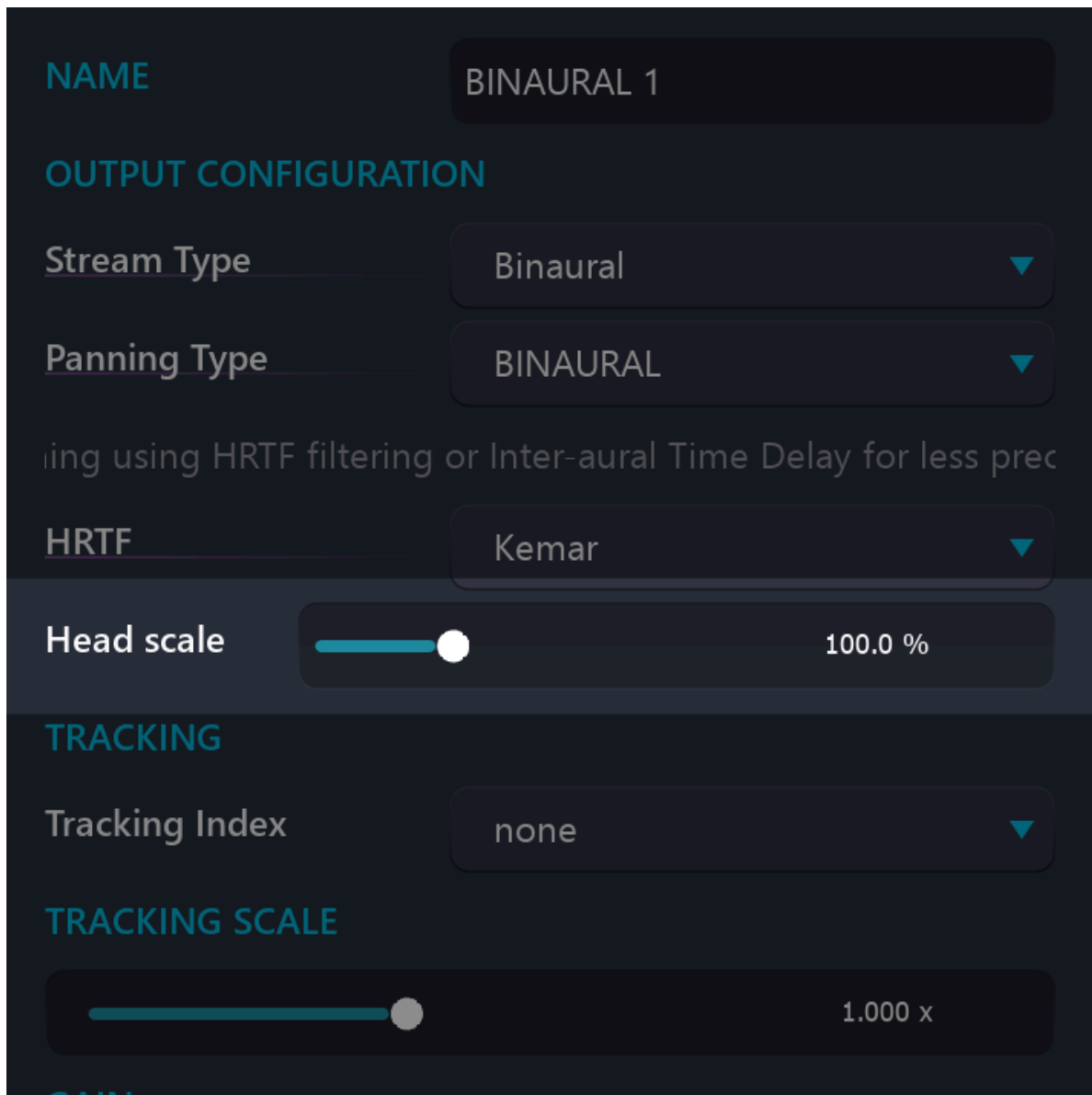


Figure 5.1: Headscale parameter

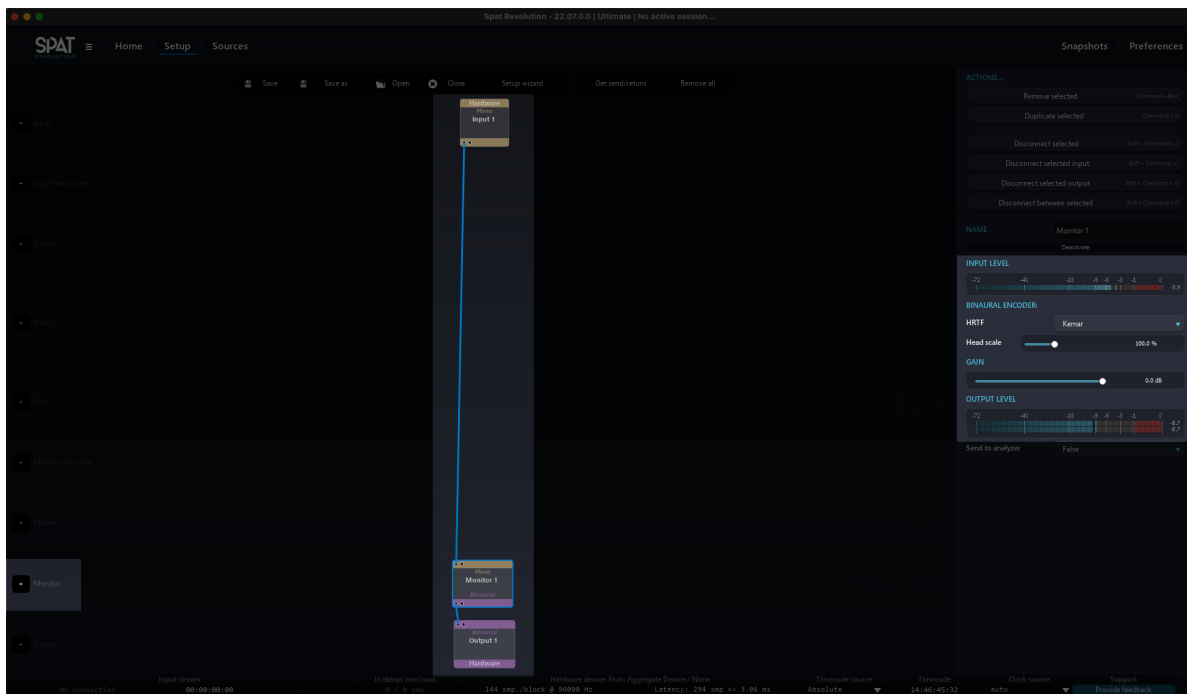


Figure 5.2: Binaural Monitor

6 Transaural

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

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

Warning

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. The loudspeaker pair need to be positioned as a regular stereo setup, Left at -30 degrees and Right at +30 degrees.

An optimum listening position such as that needed for Transaural decoding, is also known as the *Sweet Spot* (see Listener Position 4). It is a fundamental concept to be aware of as we move into the next section: *Channel Based* streams and panning algorithms.

7 Channel Based Streams

We have already covered the two-channel binaural stream type for monitoring and final encoding into a 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 (*). These streams will flow as a perfectly synchronized group through the signal graph defined in the *Setup* page. 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 Revolution*, they automatically inherit the speaker arrangement and channel count from the stream type at their connected input.

Warning

- The **Essential** license limits the total number of channels to 32 input channels and 18 output channels.

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 Revolution* is expecting the loudspeaker 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, the spatial sound image will then be compromised.

Warning

Each output channel must be routed to the correct speaker.

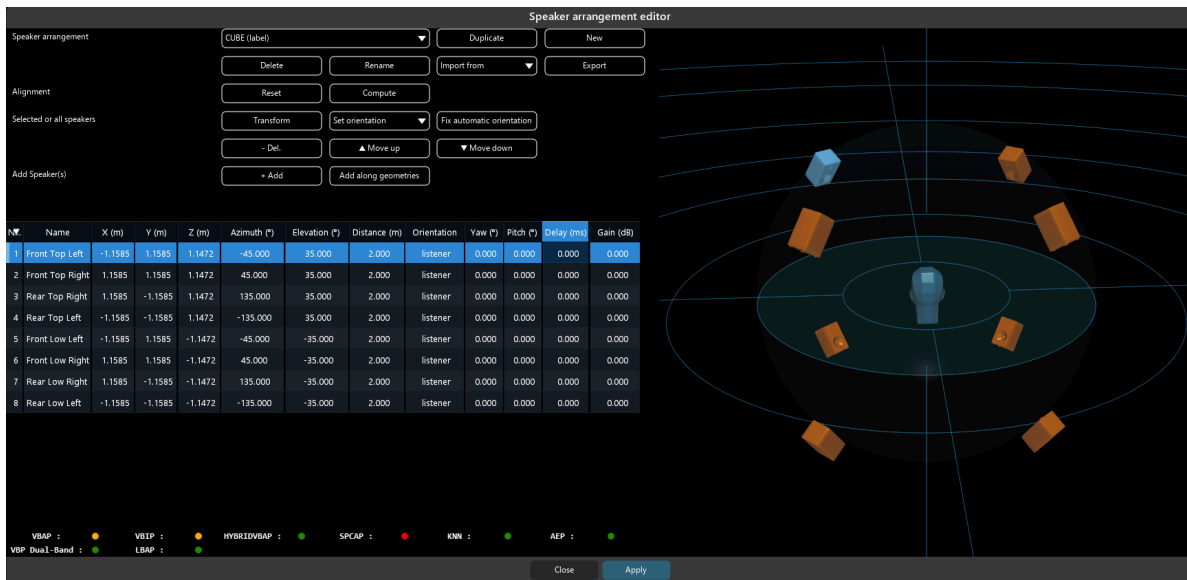
The golden rule for spatialization using channel-based audio is that each rendered channel must be connected only to its corresponding sound emitter in the destination system. An exact correlation from the model to the physical system is assumed by the calculations inside each of the panning algorithms. If the installation is not right for some reason, a listener may experience something with spatial aspects, but not with the intended quality. Correctly installing and tuning 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 is hard to tell by ears.

In order to try and simplify this process in practice, certain labeling 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 Arrangement 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

i Note

Some common speaker channel naming conventions.



The process of matching the correct channel to the correct speaker is vital to the successful rendering of the spatial composition from a *SPAT Revolution 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 us take a look at that topic before going further.

8 Panning Algorithms

There have been a lot of scientific and artistic works 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 make 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 the 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 that has become quite standardized over the years. Things get more complicated when panning over 2D or 3D speaker setups and methods for spatializing on multichannel systems are not standardized 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 hyperrealistic or totally synthesized. Many practitioners will have their preferred methods, but really it is up to the spatial sound designer to choose methods that seem to give the best experience for the listener on any given type of setup. There’s never been just one way to spatialize 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 Revolution*, you will be able to explore some of the best panning algorithms for multi-speaker setups. You can apply them in real time and identify their characteristic differences by ear. Trying them out in real time 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.

i Note

Try using more than one *SPAT Revolution* virtual Room to use different panning algorithms for sound material that has different sonic qualities (**Ultimate** license only).

i Note

Some academic papers about some following panning laws could be found here at the end of this section.

8.1 Stereo-exclusive Panning Laws

8.1.1 Stereopan Law

This mode reproduces the basic experience of a pan pot. It comes with some options:

- The pan law can follow a **sin/cos** approach or a **square root**. This produces a subtle difference in how the sound travel between the left and right speaker.
- It is possible to change the **center attenuation**. By default, a sound placed at the center of the two speakers is played back with an attenuation of 3dB (considering acoustic summing). Other possible attenuation is: -4.5dB, -6dB.
- The **PMAP**, Perceptually Motivated Amplitude Panning, aims at improving sound localization on stereo systems and on any pair speaker system with an arbitrary base angle.

8.1.2 XY and AB

These two Panning Types will only become available when a *Virtual Room* is set to be virtualizing a **stereo** speaker arrangement. 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 ± 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.

i Note

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

8.2 Vector Base Amplitude panning (VBAP)

Vector Base Amplitude Panning has become one of the more standardized methods for multi-channel spatialization. It can reproduce on a 2D or 3D configuration. Its sound is characterized by clearly localizable 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?

Traditional VBAP works by manipulating the gain of the signals being routed to the two (in 2D), or three (in 3D), the 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*.

i Note

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

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*.
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.

i Note

See the Alignment [18.4.1](#) section: the speaker alignment feature provides the impression that the actual speakers are equidistant even when they are not.

8.3 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 localized virtual sound sources.

How does it work?

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

The same four 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.

8.4 Dual Band Vector Based Panning (VBP Dual-Band)

Both Intensity and Amplitude Vector Based panning have an ideal frequency range of action:

- Localization of low frequencies is better with Amplitude Panning.
- Localization of high frequencies is better with Intensity Panning.

A hybrid approach of vector-based panning has been developed in this way: the Dual Band Vector Based Panning. This panning type merges the two approaches in order to combine the best of both worlds and to reach a better localization. Amplitude panning is applied below the crossover frequency, while intensity panning is applied above. The crossover frequency has been defined to 700 Hz by default.

The dependencies mentioned in the VBAP section also apply to Dual Band Vector Based Panning.

8.5 Layer-based amplitude panning (LBAP)

Layer-based amplitude panning can be explained as **multiple 2D VBAP layers**: The speaker setup is split into several layers, depending on the speaker elevation. The panning used between speakers on the same layer is the VBAP 2D. Between these layers, a crossfade is applied between the two nearest layers.

i Note

The difference between VBAP 3D and LBAP is the number of speakers that will be active between the layers: three in VBAP versus four in LBAP.

8.6 Distance Base Angular panning (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.

DBAP was constructed with two assumptions: - All speakers are always active, independently of the source position. - The resulting level is independent of the source position.

Only level differences are used with this panning method.

How does it work?

DBAP localizes 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 more utilized when working with DBAP.

i Note

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

8.7 K-Nearest Neighbor (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-neighbor' interpolation algorithm. This family of algorithms is 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 parameter of KNN is that the user gets manual control over one of the main coefficients in the underlying algorithm. The parameter is called *Nearest Neighbor spreading*. It sets a maximum limit to speakers' number that the algorithm can use as neighbors. The parameter becomes available as a continuously variable percentage *for each virtual source* in a *SPAT Revolution* 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 Neighbors*



Figure 8.1: Source Spreading

Spread parameter is set at a low percentage, while another sound source could seem to diffuse over the entire sound system because its spread variable is set to 100%. For example, on a 10-speaker arrangement: 1-10% will use 1 speaker, 11% to 20% 2, and so on.

i Note

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

8.8 Speaker-Placement Correction Amplitude (SPCAP)

SPCAP is a 3D panning algorithm that 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 loudspeakers 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 in being able to render wider sound sources by smartly using more speakers.

How does it work?

The result will still be *Sweet Spot* dependent, although it will be a wider listening area. SPCAP inherits some 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.

i Note

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

8.9 Ambisonic Equivalent Panning (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 rely 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 three 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 generalized channel-based type of panning. That is the AEP panning type in a nutshell.

How does it work?

AEP 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 pure Ambisonic rooms (more about that in the later section) or to use AEP 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.

8.10 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 backward 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.

8.11 Continuous Surround Panning (CSP)

This Panning Type is available in *Virtual Room* with 5.0 speakers arrangements. It optimizes the render into this arrangement, using circular harmonics. This leads to a continuous law, independent of the angle.

You can find more explanation about it in the [relative paper](#).

8.12 Related papers

8.12.1 VBAP 2D/3D (Vector Base Amplitude Panning)

“[...] Using the method, vector base amplitude panning (VBAP), it is possible to create two- or three-dimensional sound fields where any number of loudspeakers can be placed arbitrarily. The method produces virtual sound sources that are as sharp as is possible with current loudspeaker configuration and amplitude panning methods.”

“[...] the approach enables the use of an unlimited number of loudspeakers in an arbitrary two- or three-dimensional placement around the listener. The loudspeakers are required to be nearly equidistant from the listener, and the listening room is assumed to be not very reverberant. Multiple moving or stationary sounds can be positioned in any direction in the sound field spanned by the loudspeaker.”

8.12.2 DBAP

“[...] Most common techniques for spatialization require the listener to be positioned at a *Sweet Spot* surrounded by loudspeakers. For practical concert, stage, and installation applications such layouts may not be desirable. Distance-based amplitude panning (DBAP) offers an alternative panning-based spatialization method where no assumptions are made concerning the layout of the speaker array nor the position of the listener.”

“[...] Distance-based amplitude panning (DBAP) is a matrix-based spatialization technique that takes the actual positions of the speakers in space as the point of departure while making no assumptions as to where the listeners are situated. This makes DBAP useful for a number of real-world situations such as concerts, stage productions, installations, and museum sound design where predefined geometric speaker layouts may not apply”

8.12.3 AEP

“[...] A further advantage of AEP is the possibility to use an arbitrary order of directivity for each individual sound source. It becomes possible to mix pre-recorded low order ambisonic B-format, medium order ambient sounds, high order precise localizable sound and sounds with changing localizability. How the individual sounds are perceived if different orders are used at the same time is an open question that can be answered only by experience.”

8.12.4 PMAP

“This paper proposes and evaluates a new constant-power amplitude-panning law named ‘Perceptually Motivated Amplitude Panning (PMAP)’. The method is based on novel image shift functions that were derived from previous psychoacoustic experiments. The PMAP is also optimised for a loudspeaker setup with an arbitrary base angle using a novel phantom image localisation model. Listening tests conducted using various sound sources suggest that, for the 60° base angle, the PMAP provides a significantly better panning accuracy than the tangent law. For the 90° base angle, on the other hand, both panning methods perform equally good. The PMAP is considered to be useful for intelligent sound engineering applications, where an accurate matching between the target and perceived positions is important.”

9 Wave Field Synthesis

9.1 Introduction

Wave Field Synthesis, also known as WFS, is a technique for spatial audio reproduction. Unlike most of the traditional spatialization techniques, this technique **does not** depend on a phantom sound source (ie: the stereophonic principles allowing the creation of an acoustical illusion) to recreate an acoustic space.

WFS aims to reproduce the true physical attributes of a given sound field (the waves front) over an extended area of the listening room. **This approach relies on delay and amplitude differences.**

The goal of WFS is to recreate the wavefront of a primary (or virtual) source synthesized by a large number of individually driven loudspeakers. Thus, this strategy implies a sampling of the original wavefront. This is a direct application of the Huygens' principle, which states that every point on a wavefront is itself the source of spherical wavelets, and the secondary wavelets emanating from different points mutually interfere. The sum of these spherical wavelets forms the wavefront.

Under certain strict conditions, WFS can create the impression of a source positioned in front of the speaker system. This zone in front of a speaker line is referred to as the Focus Zone. That being said, the criteria are quite drastic to get this effect. Ircam states at least 32 speakers should be used with 15 to 20 cm spacing.

9.1.1 Availability - WFS Add-on license

i Note

Referred to as a panning type in *SPAT Revolution*, the WFS spatial audio reproduction technique is available with **the WFS Add-on license option**, available with Ultimate and Essential versions.

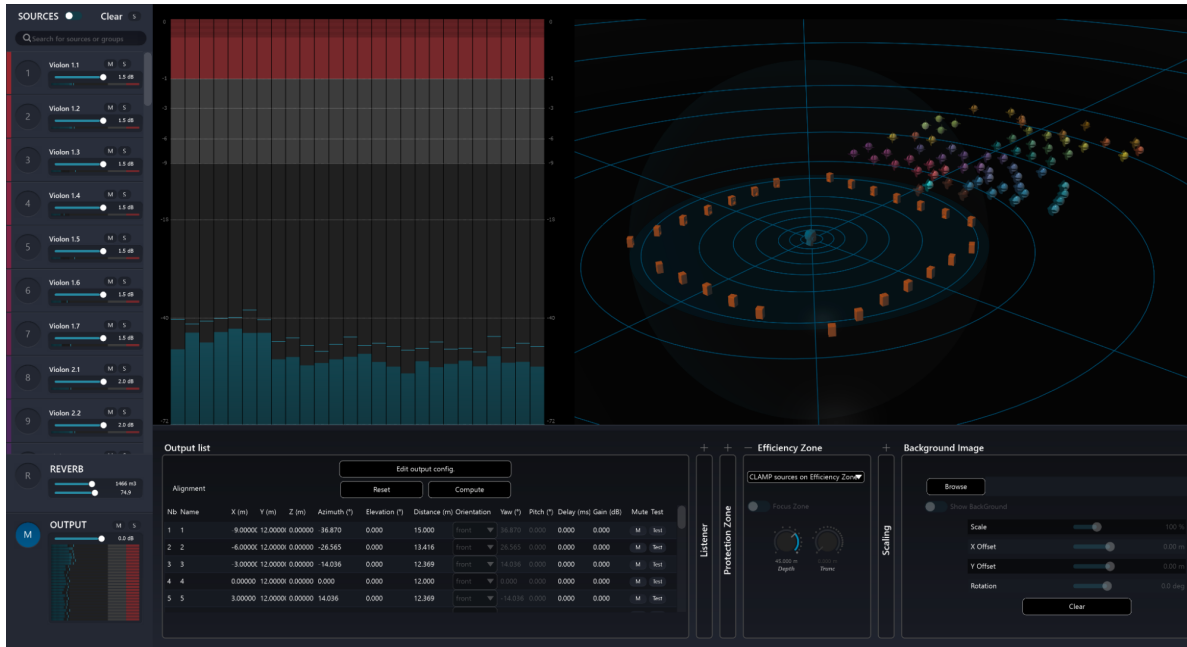


Figure 9.1: WFS

9.2 SPAT Revolution implementation of WFS

Where traditionally WFS is used on a collinear array of very tightly spaced loudspeakers, SPAT Revolution allows for the use of the spatial audio reproduction technique on systems with greater spacing, in smaller loudspeaker quantity and potentially handling any type of 2D or 3D speaker arrangements.

The panning method will become available to the room inspector as soon as a minimum of 5 loudspeakers are available.

⚠ Warning

Even if a large spacing between speakers is allowed, keep in mind that the greater the spacing you have, the less coherent the results will be.

The chosen WFS implementation is a minimum latency approach. This simply means that the closest speaker to the source has no delay. In the focus zone, the same principle is used to keep the latency at its minimum.

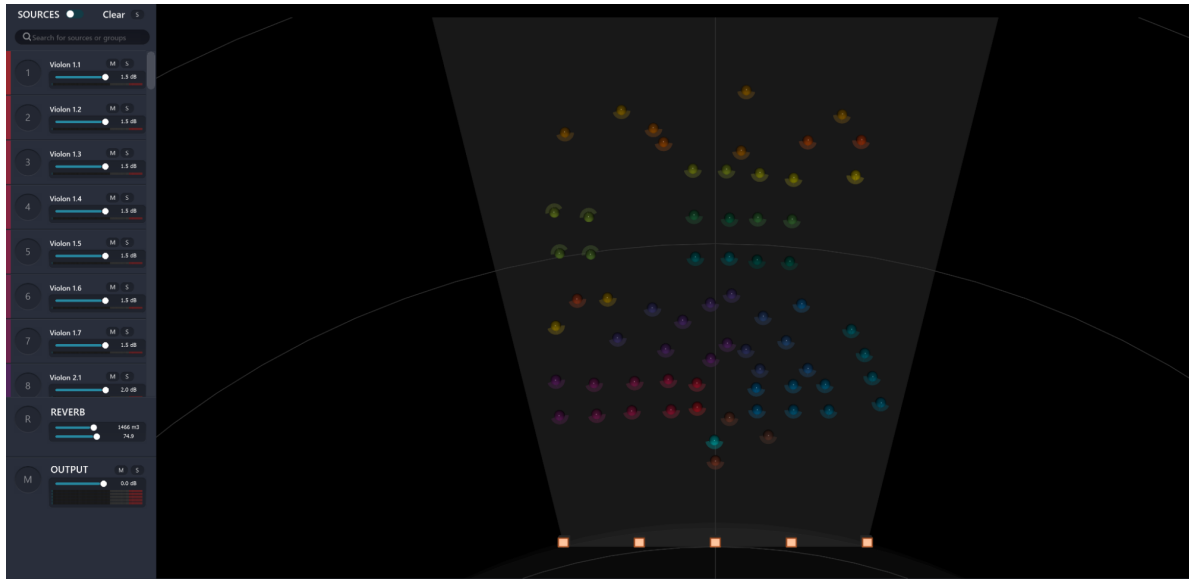


Figure 9.2: WFS



Figure 9.3: WFS Panning type

9.2.1 Focus Zone

When using the WFS panning technique on a collinear frontal line arrangement, the Efficiency zone section in the room output section provides an option to activate the focus zone, i.e. the zone in front of a line of speakers. It is deactivated by default as it is recommended that a certain density and spacing exist for the speaker arrangement.

Warning

Enabling this option may trigger an Efficiency zone warning if the system does not respect the 0.4-meter spacing between speaker source elements recommendation.

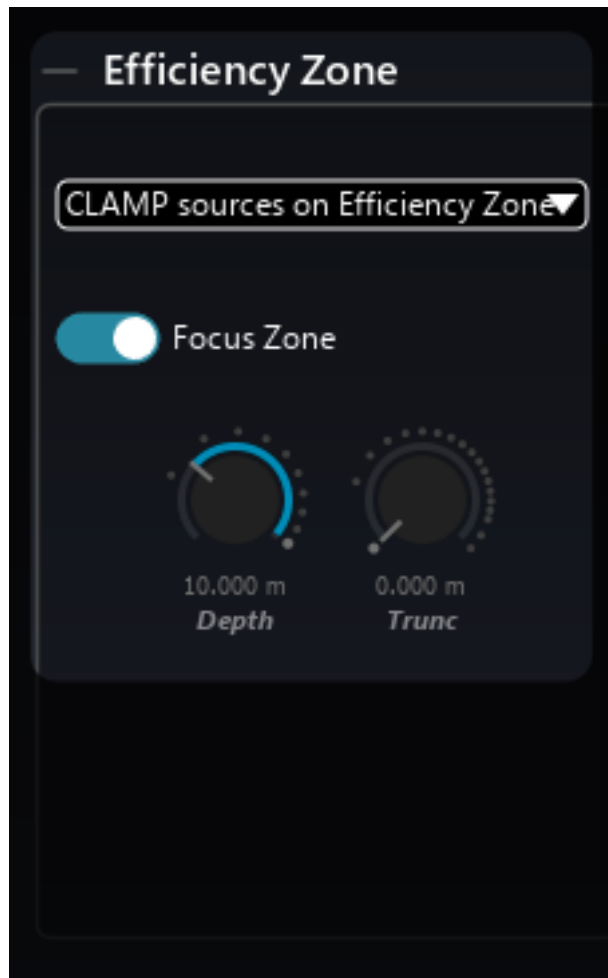


Figure 9.4: Focus zone Enable

After activating, it will extend the efficiency zone with a focus area starting at the listening

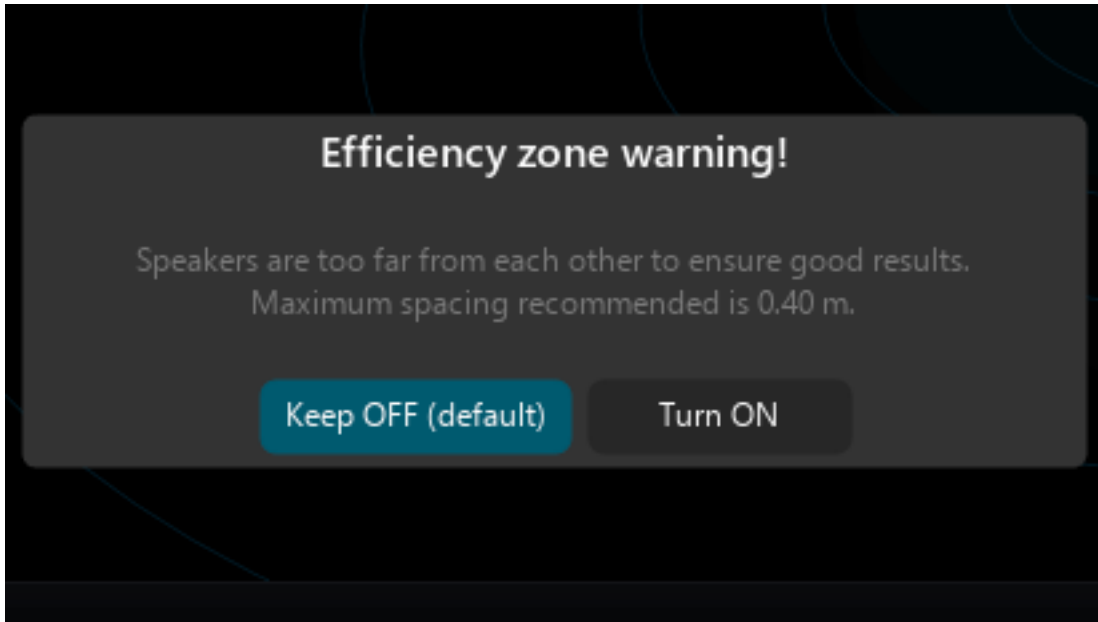


Figure 9.5: Focus zone warning

reference thus allowing sources to freely move inside the listening area.

9.2.2 Further readings

- [Good introduction to WFS](#)
- [IRCAM website on WFS](#)
- [Sound Scene Creation and Manipulation using Wave Field Synthesis](#)

9.3 WFS Settings

FLUX:: WFS implementation uses an interpolation strategy using a very precise calculation mode for delay values. This is to allow for a smooth transition with source movement rather than delay steps (inherent in digital audio). **6 parameters** are available in the room inspector of the setup page, and can allow fine-tuning the WFS render.

Delay interp. time: The delay interpolation time is the time it will take to go from one delay value to another delay value. In WFS, interpolation time allows a smooth transition when a source is moved. Depending on your source content, and your scene movements, you may find this setting useful to modify.

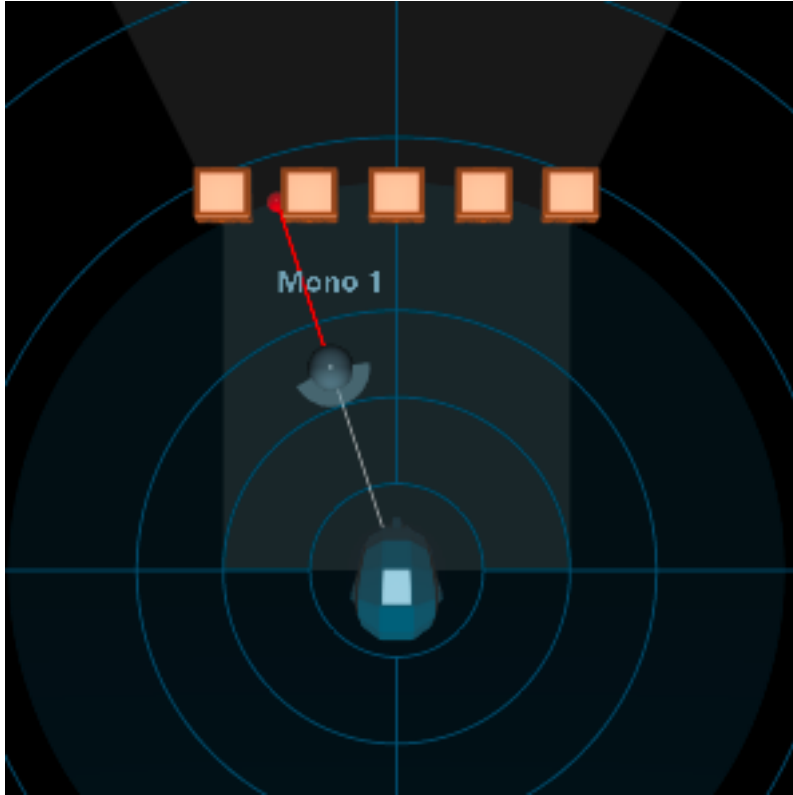


Figure 9.6: Focus zone

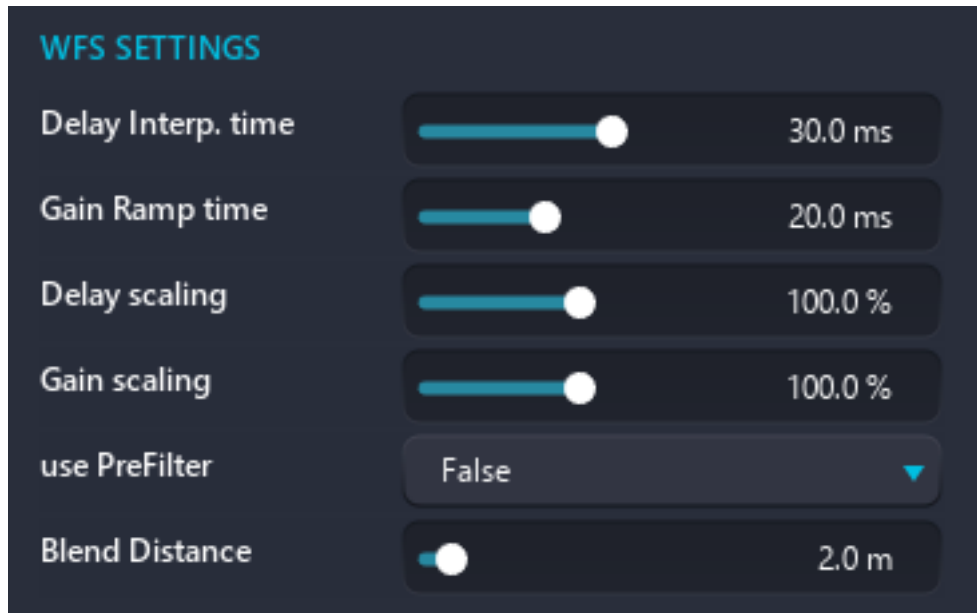


Figure 9.7: WFS Settings

Gain ramp time: Similar to the concept of delay interpolation time, the gain ramp time is the time it will take to transition from one set of gain values to another.

Delay scaling: This parameter allows scaling the computed delays for all sources. It allows shaping the wavefront. Setting the delay scaling at 0% will set all delays to 0 ms, only the amplitude differences will remain.

Gain scaling: This parameter allows scaling of the computed gains for all sources. It is really useful on close WFS line, in order to spread the audio content all along the line or speaker setup, for front-fills for example. For example, you can decrease the percentage in order to activate more loudspeakers serving well audience sitting close to the loudspeaker line.

use PreFilter: A spectral correction in WFS systems called Pre-equalization. This parameter enables/disables a low-shelf filter in order to compensate for the low frequency boost created by the speaker antenna.

Blend distance: A 2.0-meter distance behind the speaker line is defined by default to provide a transition zone as sources get closed to speakers and can help prevent some artifacts.

10 Scene based streams

10.1 Introduction to Ambisonics

Simply put: Ambisonics 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-Format microphones such as the: Soundfield SPS200, Røde NT-SF1, Sennheiser Ambeo, Coresound TetraMic and more. Alternatively, a sound field can be synthesized from any mono, stereo and multichannel sources to Ambisonics, constructing a virtual 3D sound environment by placing the sources at locations in a virtual 3-dimensional field.

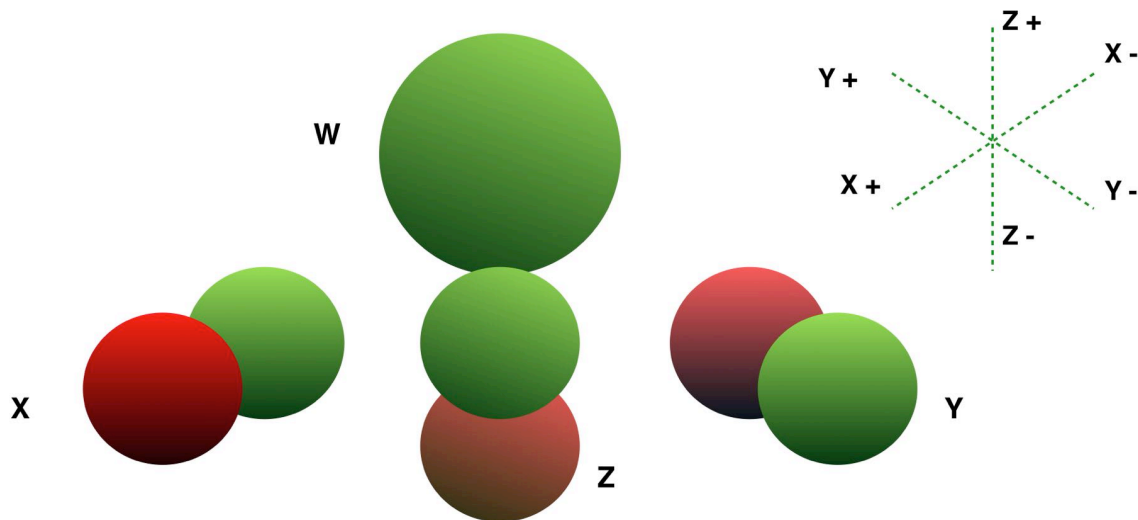
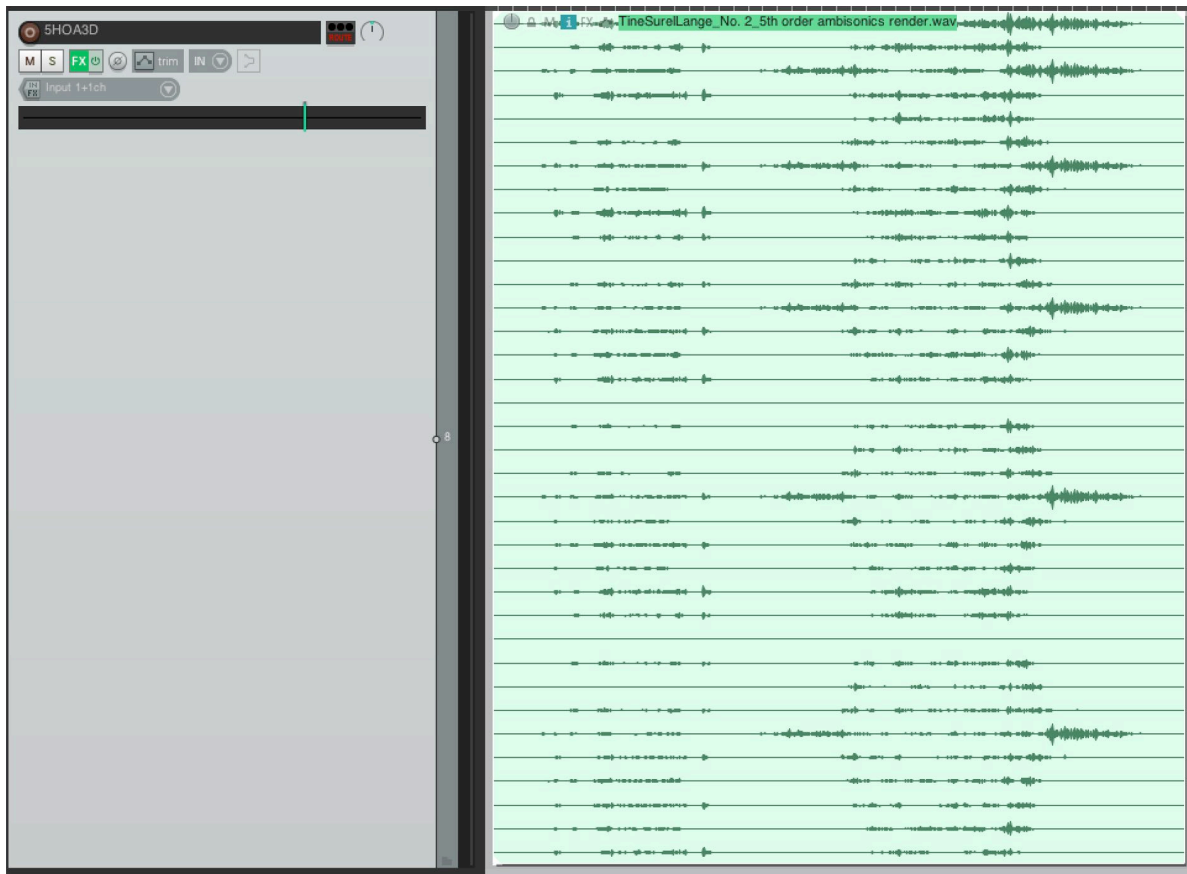


Fig.1. Ambisonic 1st. order WXYZ spherical components. Pressure and gradient polar patterns

10.2 Encoded audio

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 **decoded** to listen on speakers. This encoding / decoding scheme has the advantage of being very portable and flexible since one is not tied 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 speakers, etc. based on the chosen decoder.

When Ambisonics 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.



i Note

Overview of a fifth-order HOA 3D Ambisonic File created by Tine Surell Lange.

10.3 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, order 1, and the simplest form of 3D Ambisonics require 4 channels, conventionally named as:

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

The 4 channels or spherical components W, X, Y and Z can also be described as 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 the above figure.

High Order Ambisonics (HOA) needs more channels to contain the complex interleaved Ambisonics domain information. The components' count increases with the order: WXYZU-VSTRPQNOLMK... 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 increases quickly.

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

Ambisonics 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 $2n+1$ (where n still is the Ambisonic order).

- 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

 Warning

With **Essential** license, 1st, 2nd and 3rd orders are only available.

10.4 Normalization, sorting and presets

These different components are organized according to different standards, known as sorting. The three most used are available in *SPAT Revolution*: - ACN: Ambisonic Channel Number, WYXVTRSUQOMKLN... - SID: Single Index Designation, WXYZUVSTRPQNOLMK... - FMH: Furse-Malham Harmonics, WXYZRSTUVKLMNOPQ...

Different normalizations exist also with ambisonics. This normalization defines the relative level of the omni component compared to the other channels. It differs according to the dimension of the Ambisonics: - SN2D/SN3D: Schmidt-Seminormalized. - N2D/N3D: Fully-Normalized. - FuMa: Furse-Malham normalization.

To help with these different standards, we have created Ambisonics presets to simplify the use of it. These setups the normalization and sorting with common standards: - AmbiX: used for example by Youtube and Facebook 360. The normalization is SN2D / SN3D and the sorting ACN. - B-Format: the normalization is Fuma, and the sorting FMH. - SPAT Room: the normalization is N2D / N3D and the sorting ACN.

 Warning

Do not forget to transcode the ambisonic input if the format is different from N2D/N3D and ACN.

10.5 A-Format

ACTIONS...

Remove Selected *Command + Back*

Duplicate Selected *Command + D*

NAME Input 1

INPUT CONFIGURATION

Input Type Live input ▼

Stream Type A-Format ▼

ROUTING

Matrix

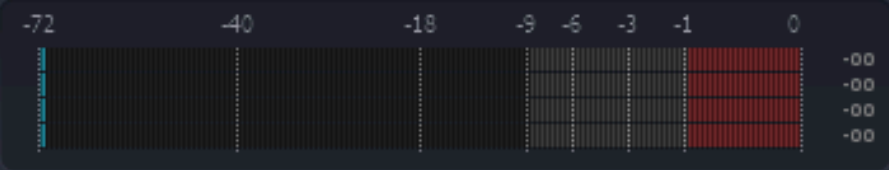
> Ch1 Ch1 ▼

> Ch2 Ch2 ▼

> Ch3 Ch3 ▼

> Ch4 Ch4 ▼

LEVELS



-72 -40 -18 -9 -6 -3 -1 0 -∞ -∞ -∞ -∞

Send to analyzer False ▼

A-format is a 4-channel audio stream. It is the RAW output of a first-order ambisonic microphone, so it has not been encoded to ambisonic yet. So we need to transcode an A-Format input to an HOA or a channel-based stream. Because each manufacturer has their own strategy to create such microphone, we have almost as many transcoders as A-format microphone builders. *SPAT Revolution* comes with a comprehensive list of A-Format transcoders, including Sennheiser Ambeo, Soundfield, DPA, Oktava etc.

10.6 History and references

Ambisonics was originally developed by the late British mathematician and sound engineer Michael 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, Ambisonics is getting a new renaissance being a perfect format for such applications.

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

- https://www.researchgate.net/publication/280010078_Introduction_to_Ambisonics
- [Wikipedia article about normalization and sorting](#)
- [Frank Zotter' book about Ambisonics](#)

11 Ambisonic Transcoding

11.1 Default transcoding

When patching an HOA or B-Format input to a source, or an HOA room to a channel-based output, a transcoder will be automatically inserted. This transcoder will, by default, be set to the Ring or Sloane speaker arrangement corresponding to the HOA order, and, will select an AllRad decoder.

i Note

Ring and Sloane are uniform setups that are recommended for optimized transcoding. It will guarantee precise localization, in particular when transcoding an ambisonic input in order to use it in a room.

11.2 Transcoding method

11.2.1 Projection

Projection decoding is also sometimes called “sampling ambisonic decoding” (SAD). It is the simplest form of ambisonic decoding. It samples the virtual panning function at the loudspeaker directions. SAD is optimal for loudspeakers arranged as t-design layouts, with t $(2N+1)$ (N being the Ambisonics order). Typically, the SAD should only be used for 2D loudspeaker layouts, i.e., regularly arranged in a circle. Avoids this decoder for 3D setups.

What is a T-Design Layout?

To keep it really simple, t-design is a mathematical way of constructing sphere perimeters or circle surfaces with an array of points that is homogenous. In 2D the point simply put on a circle and are evenly spaced. In 3D, things get much more complicated many T-design point layouts exist. In *SPAT Revolution*, we chose to use the method used by the mathematician Sloane for our speaker layouts.

11.2.2 Regularized pseudo-inverse

The pseudo-inverse decoder, or “mode-matching decoder” (MMAD), is suitable for both 2D and 3D. It is based on a pseudo-inverse of the re-encoding matrix. MMAD is well-behaved for regular loudspeaker arrangements. It can also give good results with slightly irregular setups. However, it can become unstable with strongly irregular setups, i.e., it can completely blow up the speaker feeds. So, be careful.

The regularized pseudo-inverse decoder or “regularized-mode-matching decoder” (RMMAD) is somehow similar to MMAD. However, it uses a regularization factor for the stabilization of the pseudo-inverse. This regularization factor (alpha) varies from 0% to 100%. A value of 0% provides results similar to MMAD. A value of 100% generates even energy distribution, i.e., results similar to EPAD. Intermediate values of alpha allow to “blend” MMAD and EPAD.

11.2.3 Energy preserving

EPAD (energy-preserving ambisonic decoding) and AllRAD (All-round Ambisonic decoding) are other HOA decoding methods suitable for 2D and 3D HOA, and they can cope with any kind of loudspeaker arrangement. These decoding methods always work, as soon as there are enough loudspeakers; they are always feasible and by nature numerically stable. EPAD uses a regularized matrix inversion such that the decoded energy is preserved even with non-uniformly arranged arrays (and even for directions with only sparse loudspeaker coverage). EPAD is the default method in spat5 (and the one we usually recommend).

11.2.4 AllRAD

“All-round Ambisonic decoding” (AllRAD) is designed in two steps. First, an optimal virtual loudspeaker layout using T-design arrangement is considered (for which the SAD is optimal). Secondly, the signals of these virtual loudspeakers are mapped to the real loudspeakers via VBAP.

11.2.5 Improved AllRAD

“Improved All-Round Ambisonic Decoding” (AllRAD+) combines AllRAD and SAD. Constant energy that is achieved for the idealized virtual loudspeaker setup in AllRAD is corrupted by the VBAP stage as, per loudspeaker pair, all virtual sources are superimposed linearly instead of energetically. The prevailing linear superposition increases the energy wherever the loudspeaker spacing is large. Roughly, at such directions AllRAD doubles the energy, whereas it is halved at directions with dense loudspeaker spacing. Conversely, SAD might lose all energy where the loudspeaker spacing is large and roughly doubles it where the loudspeaker spacing is dense. AllRAD+ tries to solve this issue by combining (i.e., mixing) SAD and AllRAD. The

loudness variation of AllRAD+ is competitive with EPAD and its angular mapping resembles AllRAD.

11.3 Transcoding types

To improve the ambisonic render, there are some strategies that can be applied at the decoding stage. The idea is to optimize the phase or the energy to improve sound localization.

11.3.1 Basic

This is the standard way to decode ambisonic, and no optimization is applied.

11.3.2 InPhase

The audio content will be optimized in phase, for the all spectrum.

11.3.3 MaxRe

The energy of the audio content will be optimized, for the all spectrum.

11.3.4 BasicMaxRe

The low end of the audio content is not optimized, but a MaxRe method is applied to the high end. The crossover frequency is by default set to 700 Hz.

11.3.5 MaxReInPhase

The low end of the audio content is optimized in energy (MaxRe), but a in-phase method is applied to the high end. The crossover frequency is by default set to 700 Hz.

11.3.6 InPhaseMaxRe

As phase optimization is more efficient in the low frequencies, and energy optimization is prominent in the high frequencies, this method takes this phenomenon to its advantage by splitting the signal into two frequency bands. The crossover frequency is by default set to 700 Hz and can be adjusted.

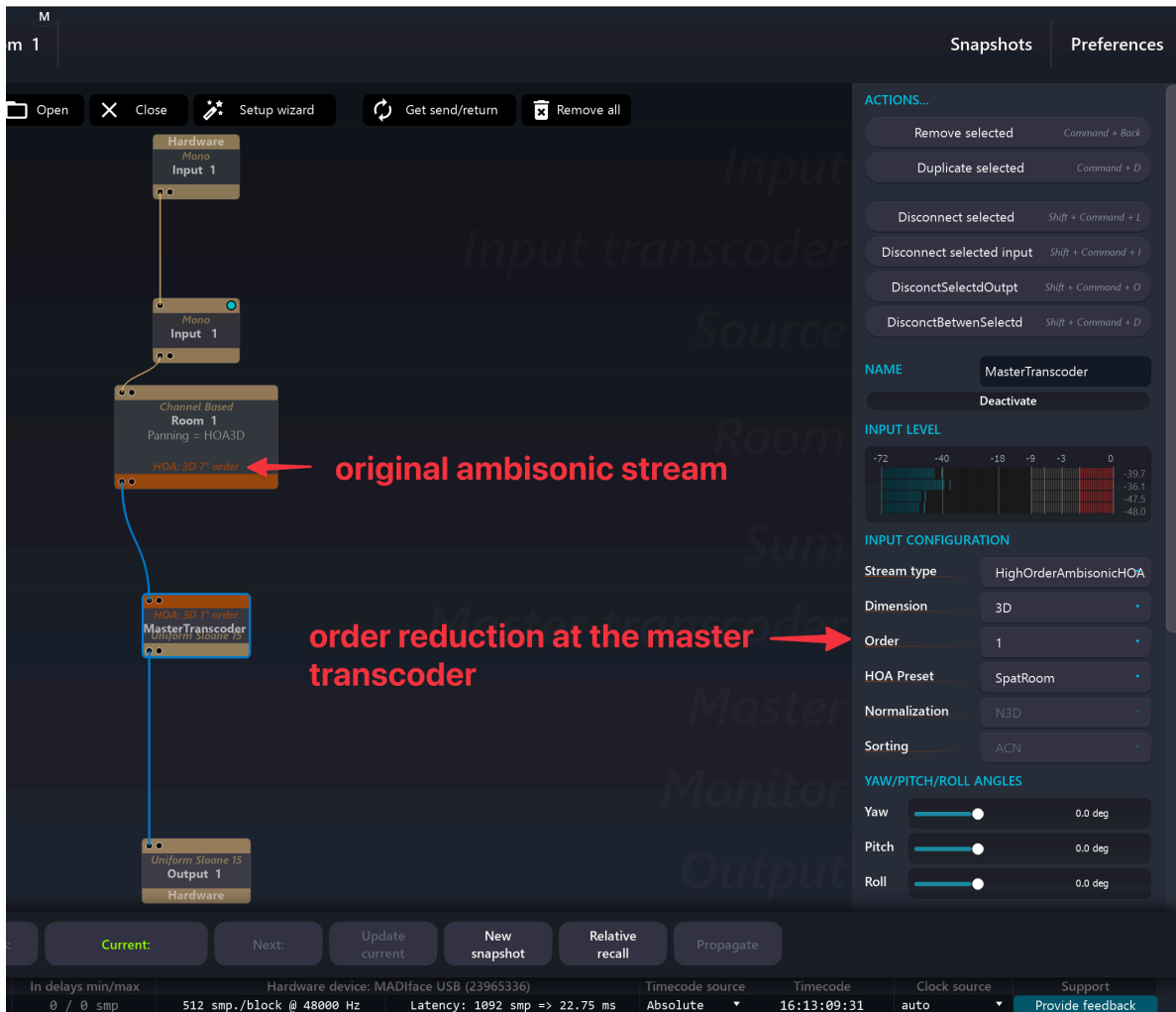
11.4 Basic rules of thumb when transcoding ambisonic to speaker

11.4.1 Ambisonic order and number of speakers

There is a direct relationship between the ambisonic order and the number of loudspeakers we want to use. **There must always be at least as many loudspeakers as there are ambisonic channels.** For example, first-order ambisonic encapsulates 4 channels and second-order ambisonic encapsulates 9 channels. So we should use first-order ambisonic for a 3D array of up to 8 loudspeakers, then move on to second-order ambisonic. We could continue to use second order until our loudspeaker array grows to use more than 15 loudspeakers. We can continue this way of thinking up to 7th order ambisonic.

11.4.2 How to handle multiple renderings from a single ambisonic room

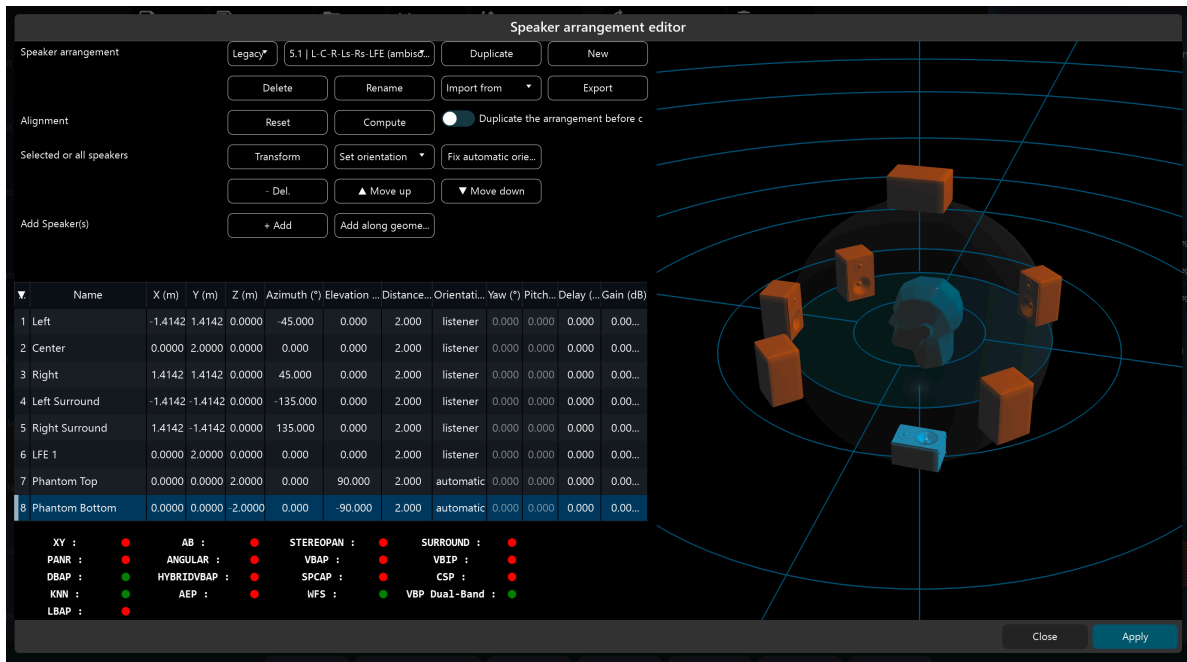
An example of an ambisonic workflow would be to use a 7th-order room and then decode it to the desired loudspeaker array. As mentioned before, we need to adjust the ambisonic order depending on the number of loudspeakers. It is possible to reduce the ambisonic order by changing the order of the input stream in the master transcoder. When this is done, higher-order channels will not affect the rendering.



If we want to transcode a 3D ambisonic stream to a 2D speaker layout, one solution is to use two phantom speakers, one at the top and one at the bottom. For example, we could use the speaker layout shown below to transcode a 3D ambisonic scene into a 5.1 layout. It's important to note that this method will **fade out** the element that had elevation.

! Important

Phantom speakers must be taken into account when determining the appropriate ambisonic order.

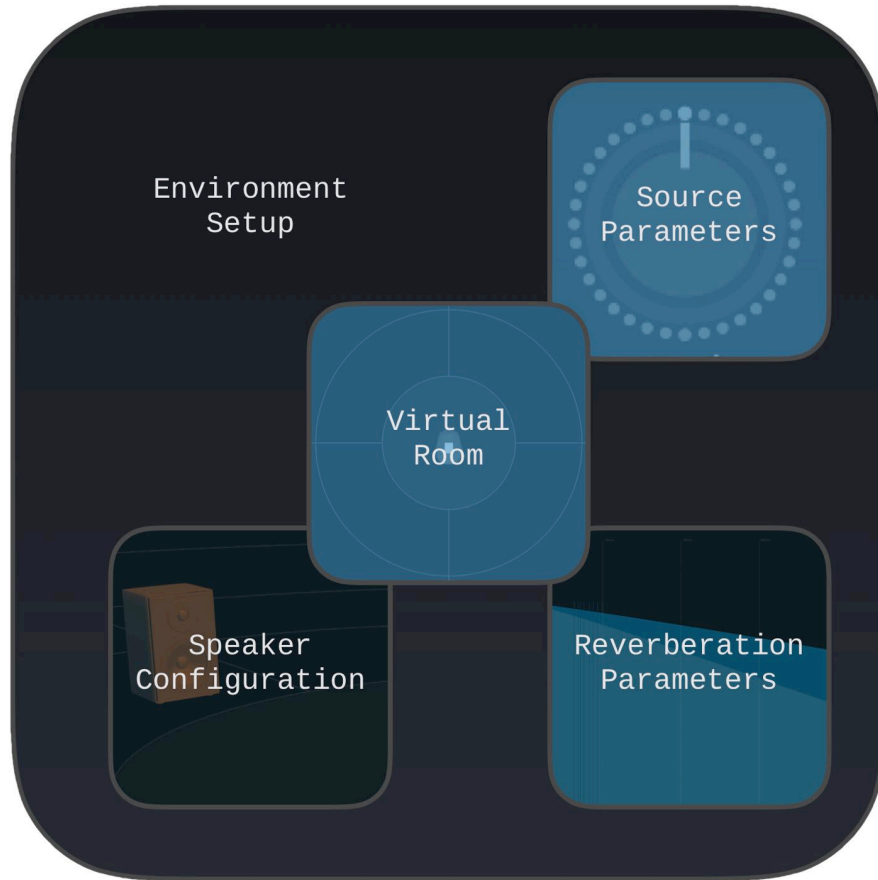


We can then have as many transcoders as we like, each dedicated to a specific loudspeaker layout and each adjusting the ambisonic stream accordingly.

Part V

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.



In a nutshell, the **Setup Page** is the brain of your SPAT Revolution session. It's where you route signals and address your mix to the loudspeaker array you're using.

The **Home Page** is a simple hub that displays your previously opened sessions and gives you a user-friendly way to get started.

The **Snapshot Page** is the interface to the very powerful Snapshot engine. Here you can create, re-order, retrieve, update snapshots and much more!

The **Item Page** is another way of visualizing information in your session. It is very powerful in terms of setup or OSC integration to quickly access the information that matters.

The **Speaker Arrangement Editor** is a simple yet comprehensive tool for creating and managing speaker layouts. Remember, this is a critical step in creating a convincing spatialization experience.

The **Preferences Page** is where you will find many of the application settings that allow you to tailor SPAT Revolution to your needs.

While we cover these main menus, we will also discuss the files and folder structure of the application, as well as discussing signal ballistics and a few other topics.

12 Files and Folders

12.1 Files type

SPAT uses 3 different file types:

- .json
- .ioconfig
- .reverbPresets

.json are the main files of *SPAT Revolution*: sessions are saved in this file type.

To save a session, click on the “Save session” button on the Setup page, or use the shortcut **Ctrl + S** on Windows, or **Cmd + S** on Mac, on any page of *SPAT Revolution*.

User-created custom speaker arrangement(s) can be exported or imported as *.ioconfig* files in the edit speaker config window.

Reverb preset can be stored and exported as *.reverbPresets*. They can later be imported back into a session.

12.2 Preferences and resources files

Users/.../Document/FLUX SE

Users/.../Document/FLUX SE - IRCAM

Users/.../Document/Ircam

The FLUX SE Folder contains a subfolder named *Config* which has 3 files:

- *.ioconfig* contains your added speaker arrangements to *SPAT Revolution*
- *.presets* contains your reverb presets
- *.theme* contains your theme (Dark or Light mode)

A subfolder named *Preferences* containing:

- *hrtf.json* file which includes your HRTFs files location
- *users.json* contains your saved software preferences

- Preferences.xml saves some paths
- UI.xml saves your user interface preferences
- Property Memory subfolder contains the memory slots saved by parameters

A subfolder named *Shell* containing:

- history.txt - a history of the terminal commands

The FLUX - IRCAM Folder contains preferences and presets of the three *SPAT Revolution* plug-ins.

The Ircam Folder contains a subfolder called *sofa* which contains the sofa.catalog.xml file. The HRTF catalog.

i Note

When backing up a system, make sure to copy all these folders to secure the complete software configuration.

12.3 Path for Recorded Audio Files

Audio files are recorded in the directory of your session, in a folder named “Session-Name_media”, SessionName replaced by the name of the SPAT session name. To be noted, if the session is in a folder of the same name, the folder for the audio is “Media” (no need to be documented, only for internal information). Audio is recorded in wave Riff64, 32-bit float, interleaved. All inputs will be recorded in one unique file. For the other blocks, a single file will be created for each block, with the following name convention: “Blocktype blockname (format) - recordNumber.wav”. For example, for a room named “WFS” in the format “Frontal Line 5”, it will be “Room WFS (Frontal Line 5) - 0001.wav” This is not a broadcast wave for now, the recording has no timestamp.

i Note

A first file is created when arming a file to prevent CPU spike at the recording start. It will be deleted when de-arming the blocks or closing the session.

12.4 Paths for python script

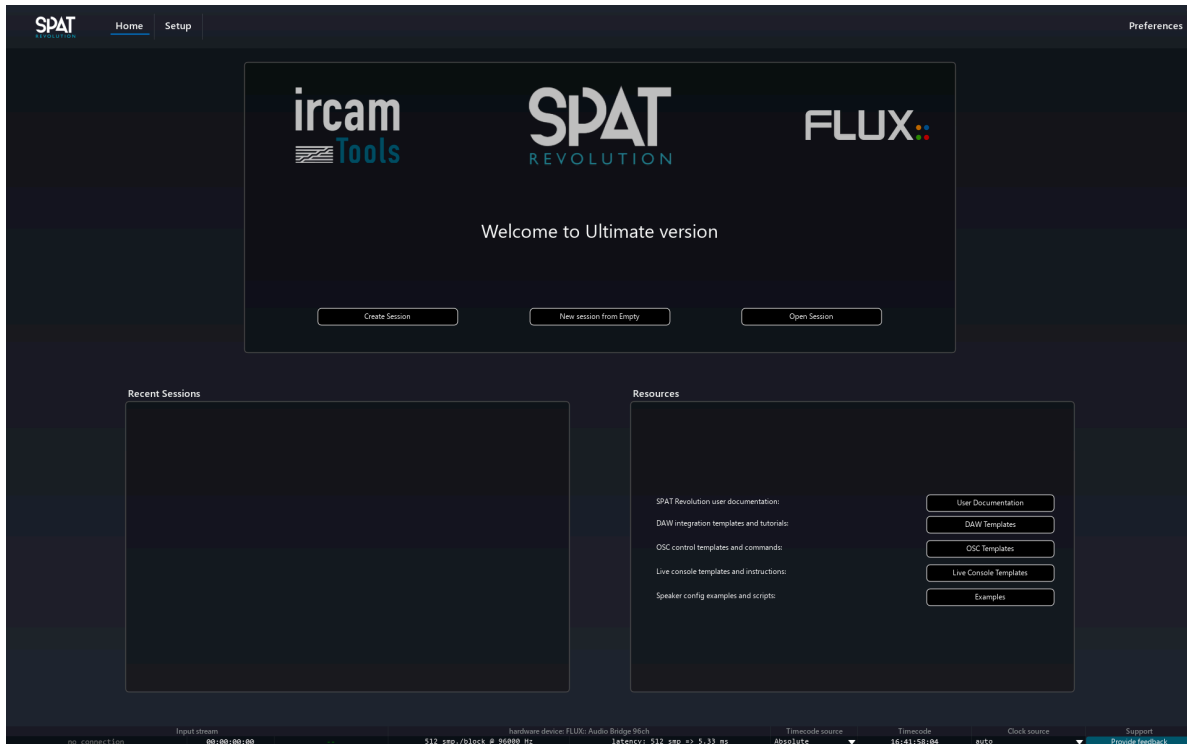
Users/.../Desktop

Users/.../Document/FLUX/Spat Revolution

i Note

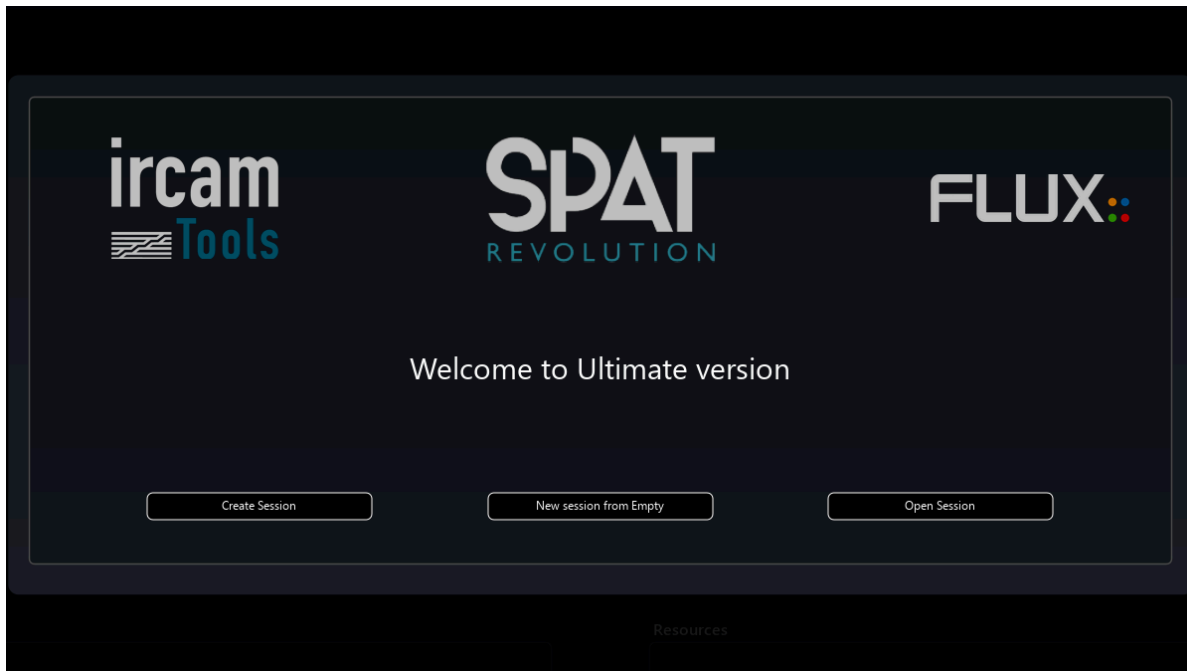
An example of a script file is the [customSpeakerArrangement](#) that can be used as a method to add arrangements to SPAT Revolution.

13 Home Page



This page is the welcome page of *SPAT Revolution*. Here you can create sessions, open ones from disk or from a list of recent ones or access to the resources of *SPAT Revolution*.

13.1 Welcome box



This simple box lets you access to basic action of session management.

13.1.1 Create Session

Create a session with a wizard to set up the routing.

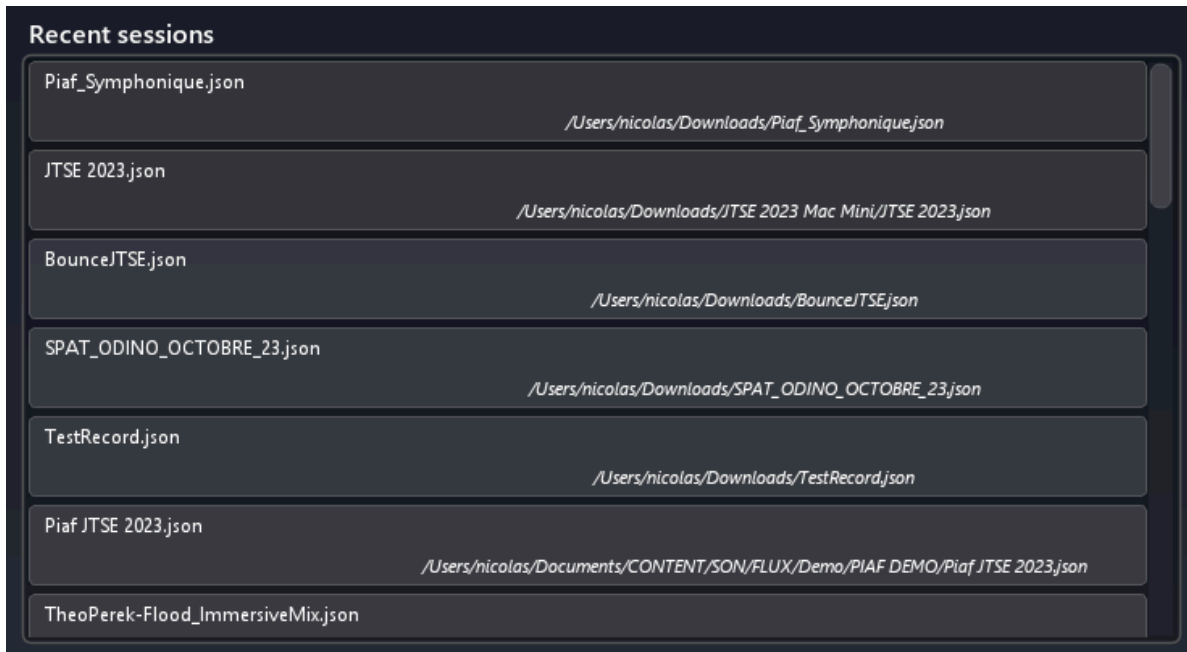
13.1.2 New session from empty

Create an empty session, without calling the wizard.

13.1.3 Open a session

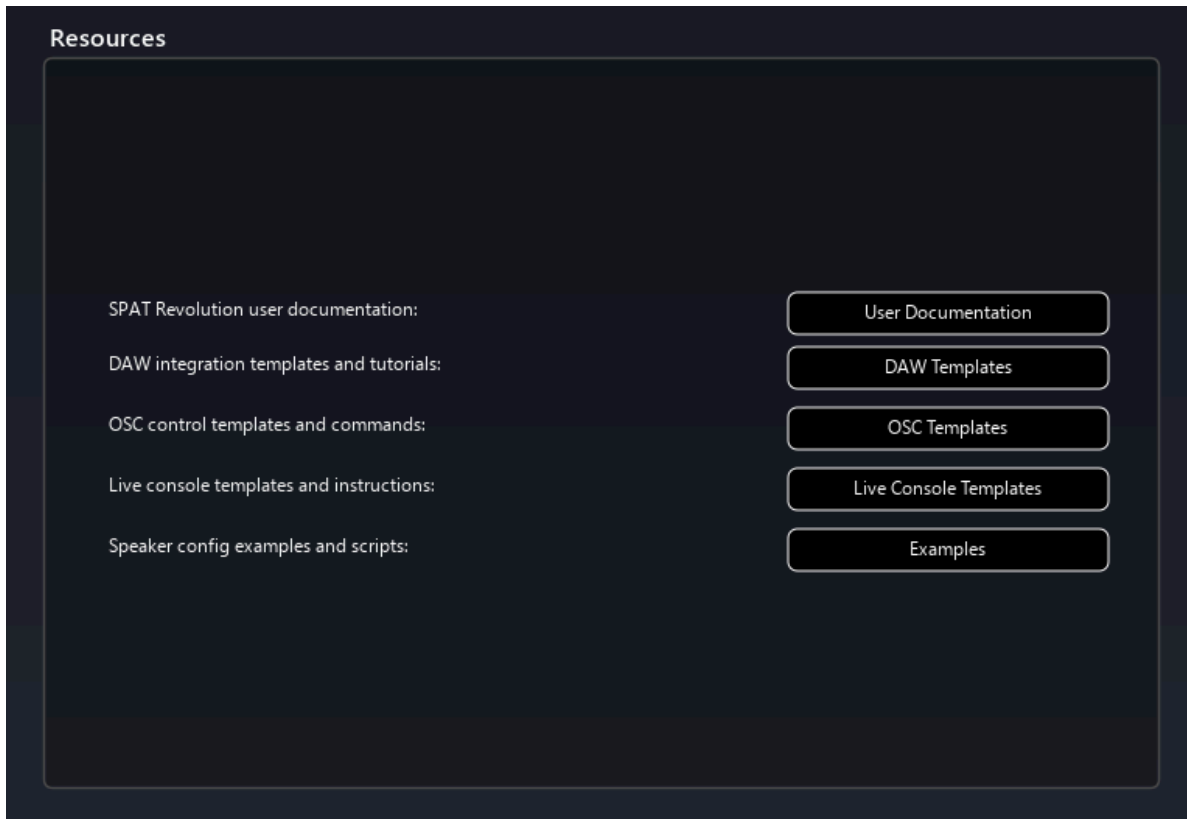
Let you open a session from a disk.

13.2 Recent Sessions



This box lists the latest open session in *SPAT Revolution*. You can open one by double-clicking on it.

13.3 Resources



This box gives you direct access to many resources related to *SPAT Revolution*:

- This documentation
- DAW templates for using LAP: Ableton Live, Apple Logic Pro X, Avid ProTools, Steinberg Nuendo, Merging Technologies Pyramix
- OSC templates, which include a lemur template and a QLab example session
- Live console templates: S6L, Digico and SSL
- Examples of various scripts.

14 Global Bars

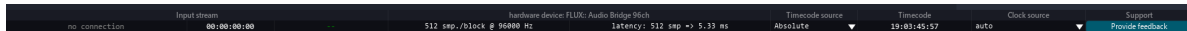
14.1 Navigation bar



The Navigation bar appears at the top of all views. As well as links to different editor views and to the preference, it also offers you the possibility to mute and unmute rooms.

The status and help bars appear 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 Revolution* graphical user interface. The *Provide feedback* button sends a message directly to FLUX:: Immersive support which automatically includes your system information for our support team.

14.2 Status bar



The status bar helps you easily monitor many critical information about your hardware and your incoming or outgoing audio stream.

14.2.1 Input stream

Allows monitoring of the good reception of audio through LAP. + When “No connection” is displayed, this means that no SPAT Send plug-in with LAP enabled is seen by *SPAT Revolution*. + When a connection is validated, it should provide a timecode, corresponding to the DAW transport position. + If the sample rate or block size is unmatched between *SPAT Revolution* and the DAW used, the “all in sync” mention will turn red and count the sync errors.

14.2.2 Hardware device

This section helps you monitor your audio hardware and how it interacts with *SPAT Revolution*.

- You can see the name of your input & output device.
- The block size and sample rate of *SPAT Revolution*.
- The whole latency of the system. This depends on the block size and on the selected audio interfaces. Using different audio interfaces can result in higher latency.

14.2.3 Timecode source

Here, you can choose which timecode *SPAT Revolution* should look at.

- By default, it is set to “Absolute”. It refers to the clock of your computer, no matter the connection with an audio interface or with LAP streams.
- “LAP Send” looks at the clock provided by SPAT Sends plug-ins.
- “LAP Return” looks at the clock provided by SPAT Returns plug-ins.
- “Audio hardware” looks at the clock provided by the selected audio interfaces.

14.2.4 Timecode

Let you monitor the timecode seen by *SPAT Revolution*. For proper operation of snapshots and automations, it should be always running.

14.2.5 Clock source

Choose what clock SPAT should follow. + Internal. + Hardware. + Auto: use hardware if one is connected.

14.2.6 Support

Allows to send feedback to the FLUX:: Immersive support team.

i Note

With Send/Return plug-ins, if the sampling rate and block size between DAW and *SPAT Revolution* are different, the status bar will be red. Double-click on this bar to automatically change them into *SPAT Revolution*.

14.3 Snapshot bar

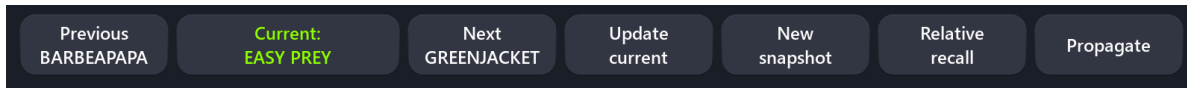
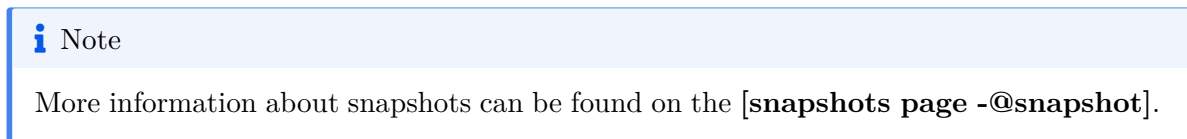


Figure 14.1: Snapshot toolbar

This toolbar has been designed to help to handle snapshots without navigating to the snapshots page. Some of the most important snapshot actions will be found on it: * Recall the **Previous** snapshot. * The name of the **Current** snapshot. Clicking on it will display the snapshot list, enabling to recall any snapshot of the list. * Recall the **Next** snapshot. * **Update** the current snapshot. * Enable or disable the **Relative Recall**. * **Propagate** values between snapshots.

*This bar can be hidden on the snapshot panel of the **Preference** page.*



14.4 Transport bar

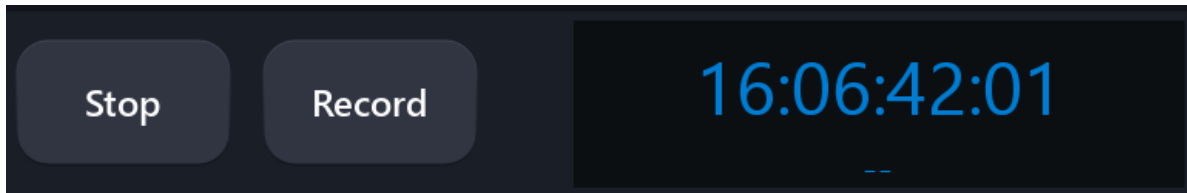


Figure 14.2: Transport bar

The transport bar allows to access to the recording feature of SPAT Revolution. It is possible to write the audio stream going through any blocs (except for the sources) to audio files.

It is possible to launch the recording by clicking on the “*Record*” button in the transport bar, located at the bottom of the user interface, or using the shortcut “Cmd/Ctrl+Shift+R”. The session must have been saved to enable the recording. Otherwise, a popup will appear asking to save the session. Closing the session, deleting a block, or dearming a block will automatically stopped the recording.



Figure 14.3: Media bar

14.5 Media stream bar

This bar provides information about the recording blocks, the file names and path, the elapsed time of recording, the number of channels recorded, and more. It will be automatically displayed when arming a block. Also, it can be shown/hidden via the menu “View/Show media stream”, or via the shortcut “Cmd/Ctrl+Alt+Shift+O”.

15 Setup Page

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

15.1 Setup Modules

The *Environment Setup* editor is a 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.

15.2 The graph editor

The modular graph editor handles basic mouse editing.

15.2.1 Module selection

Modules can be selected by clicking on them. To select several ones, hold `Ctrl/Cmd` while clicking on additional modules. It is also possible to use a marquee selection by left-clicking, holding the button, and dragging the mouse over the modules you want to select.

15.2.2 Drag & Drop

The drag-and-drop feature allows an easy and ergonomic way to connect and reorganize modules on the setup page.

Connect modules

To create a connection between two modules, simply drag one on the other. SPAT will automatically connect the two modules. If it is necessary, SPAT will also create supplementary modules if needed. For example, if we drag and drop an input on a room, SPAT will automatically create a “source” block between them.

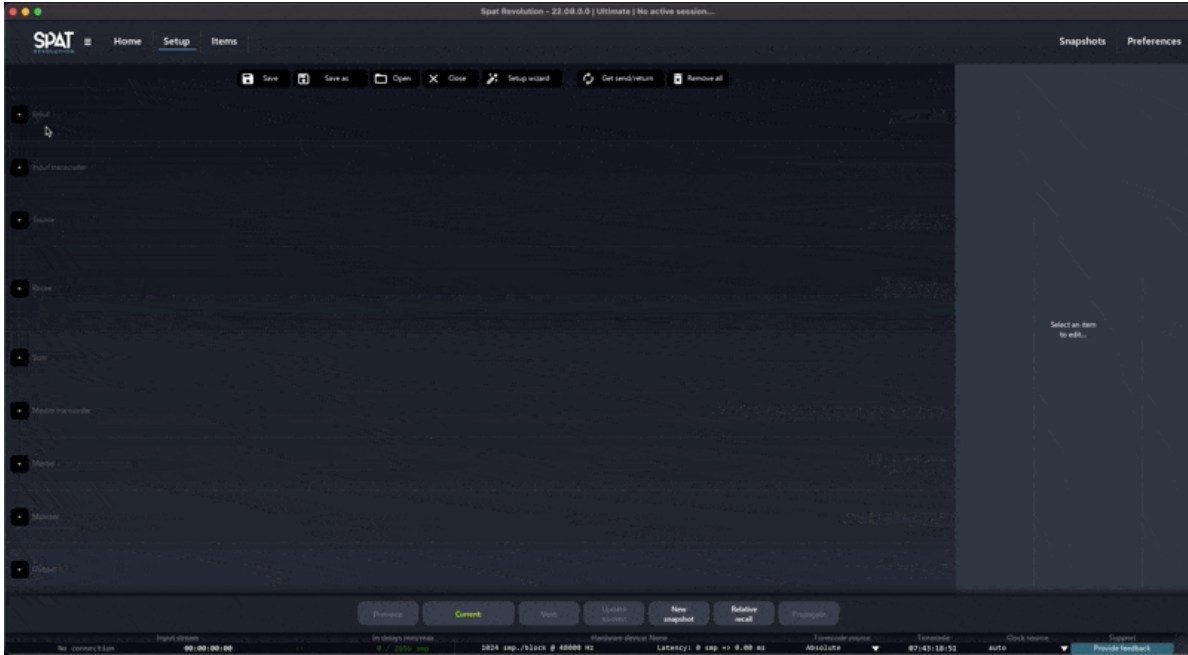


Figure 15.1: Connect blocs with drag & drop feature

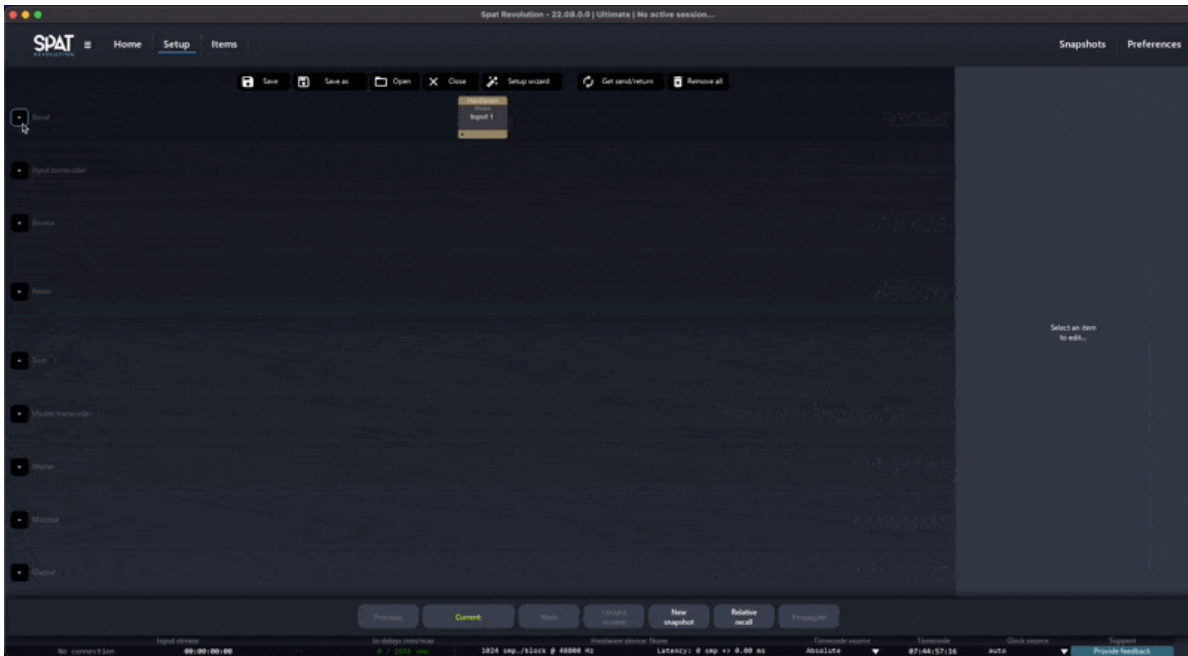


Figure 15.2: Auto-connect blocs with drag & drop

This feature also works on a selection of multiple modules of the same type. For example, if we wished to connect 5 inputs to 1 output, we can select our inputs and drag them on the output. All the input modules will be patched to a room block through sources, and the room is a patch to the output through a master block, with the default stereo room that can be changed later.

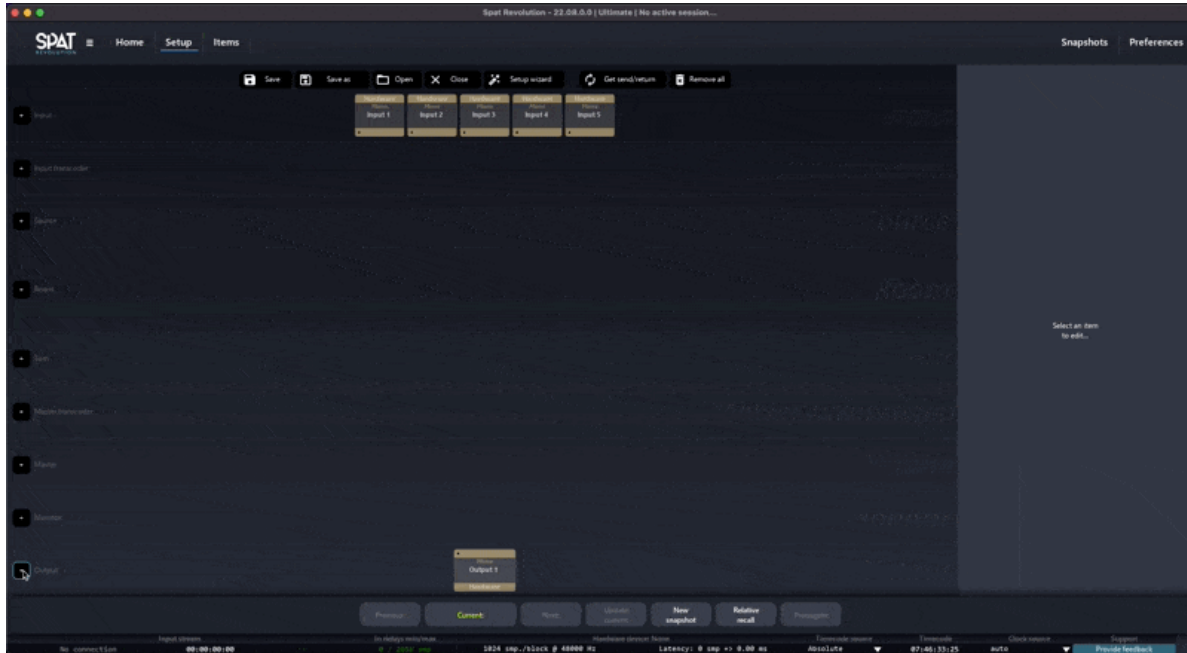


Figure 15.3: Quick setup using drag & drop

Reorganize modules

The drag-and-drop feature also allows reorganizing the modules of the same type. This means that you can now change the order of already created modules. This gives to the setup page a more ergonomic and flexible feel.

Warning

Important to note that this will be changing the index number of the source. So be careful with automation already created. This is specific to OSC like using the plugins with OSC where the index is important. Not the case with software sources/inputs that use a different ID system.

15.3 The setup wizard

In our effort to make *SPAT Revolution* easier to use, we created a small utility to help you set up new SPAT sessions. This is used mainly when dealing with hardware I/O.

To open it, you can either:

- Click on the **Setup Wizard** button in the top menu of the Setup page;
- Go to the main menu, into setup, then “Setup Wizard”;
- Or, use **ALT + W** (the shortcut is not working in Windows).

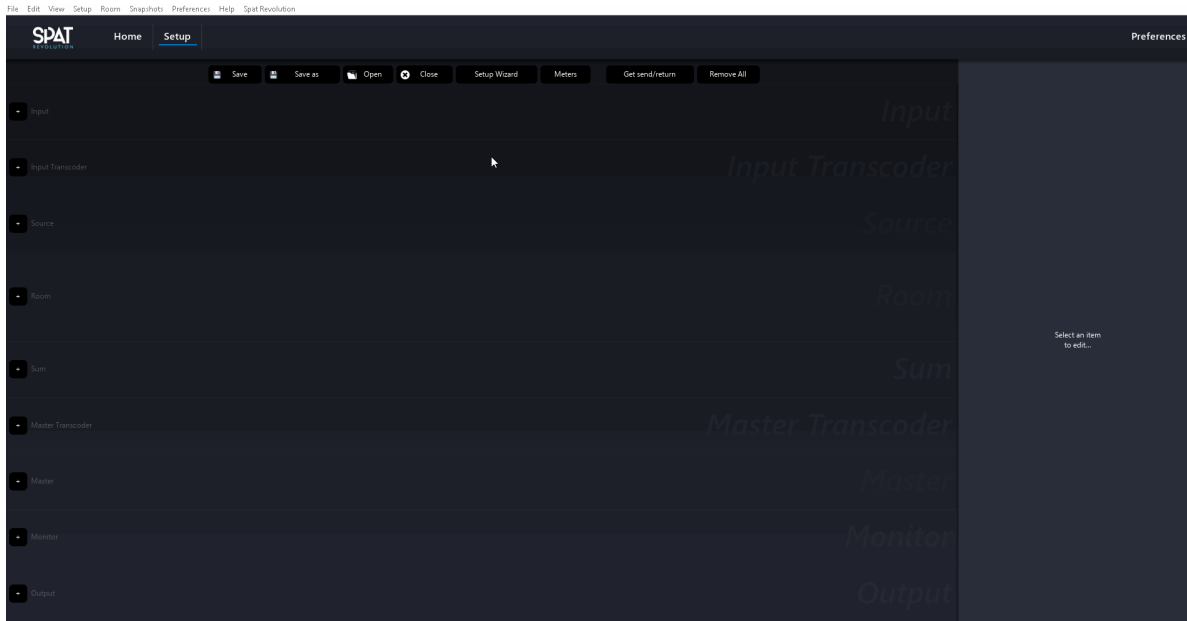


Figure 15.4: Opening the Setup Wizard

The top part of the setup wizard allows to create a new room (with associated options) or to select an existing room to patch new sources into. If a new room is created, we can choose its stream type and many options linked to it. We can also choose to associate a binaural monitoring block to it (virtualizing the room output). Lastly, for each new room created, a master block and an output block is also created.

The main part of the wizard allows the creation of different types of sources. It works like a table where each line can be used for a specific input stream type. To add or remove a line, simply click on the + or - sign on the left side of a line. You can also use the shortcut **Ctrl/Cmd + Go Down** or **Ctrl/Cmd + Go Up**.

Other shortcuts have been implemented in this wizard: - **Go up** and **Go down** to increase / decrease the number of sources - **Go Left** or **Go Right** to change the Stream Type. - **Ctrl/Cmd +**

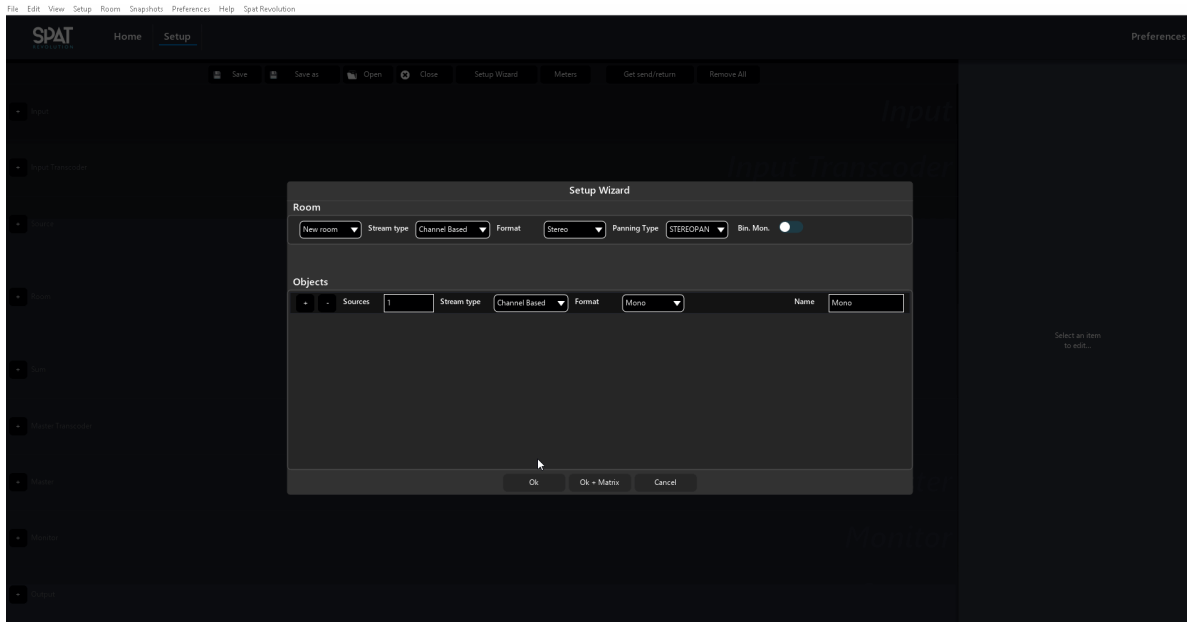


Figure 15.5: Setup Wizard in action 1/2

Go Left or Ctrl/Cmd + Go Right to change the format (if Channel Based), or the Dimension (if HOA). - Ctrl/Cmd + Shift + Go Left or Ctrl/Cmd + Shift + Go Right to change the Order (if HOA).

When we are done creating the different sources, we have two ways to validate the operation. We can either click on **Ok**, and all the sources, rooms and outputs will be created, with a straight routing, or, we can choose to click on **Ok + matrix**. This last option will open the input and output matrix of our whole *SPAT Revolution* session to allow us to quickly customize or validate our patch. Also, if you need to easily create a line in SPAT matrix, simply hold **Ctrl** and click on the starting point of your line.

15.4 Action menu

This menu handles the basic operation of modules, like connecting, disconnecting and renaming. Some of these actions are contextual, and depend on the block, or the number of modules selected.

Warning

Note that there is no undo in *SPAT Revolution*. Think twice before operating and save regularly.

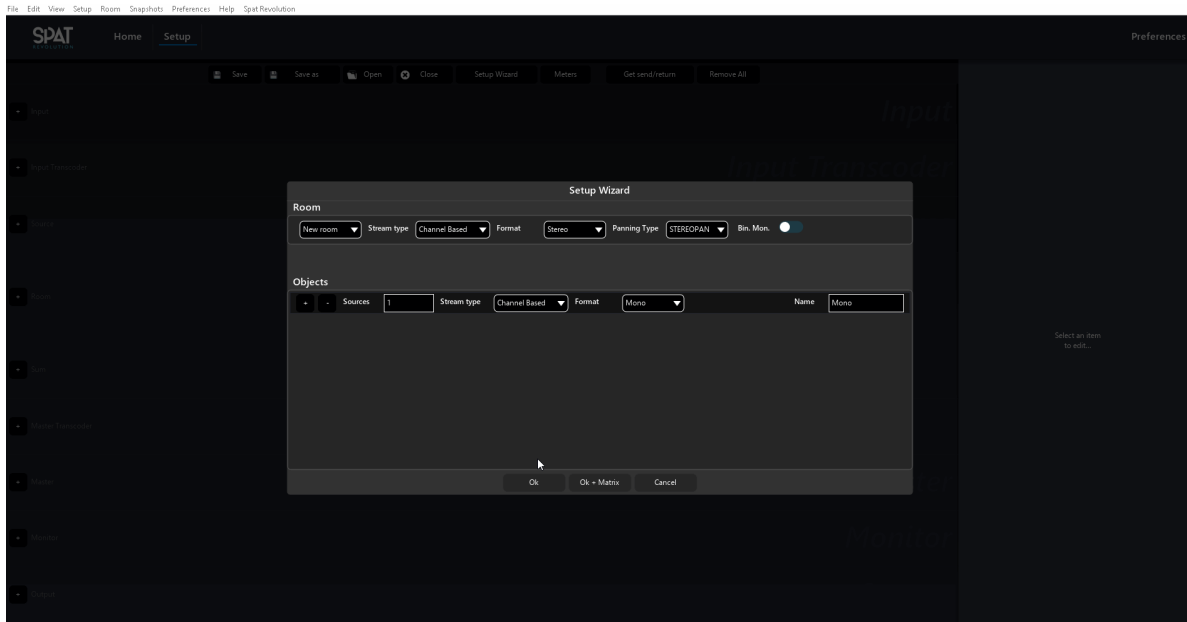


Figure 15.6: Setup Wizard in action 2/2

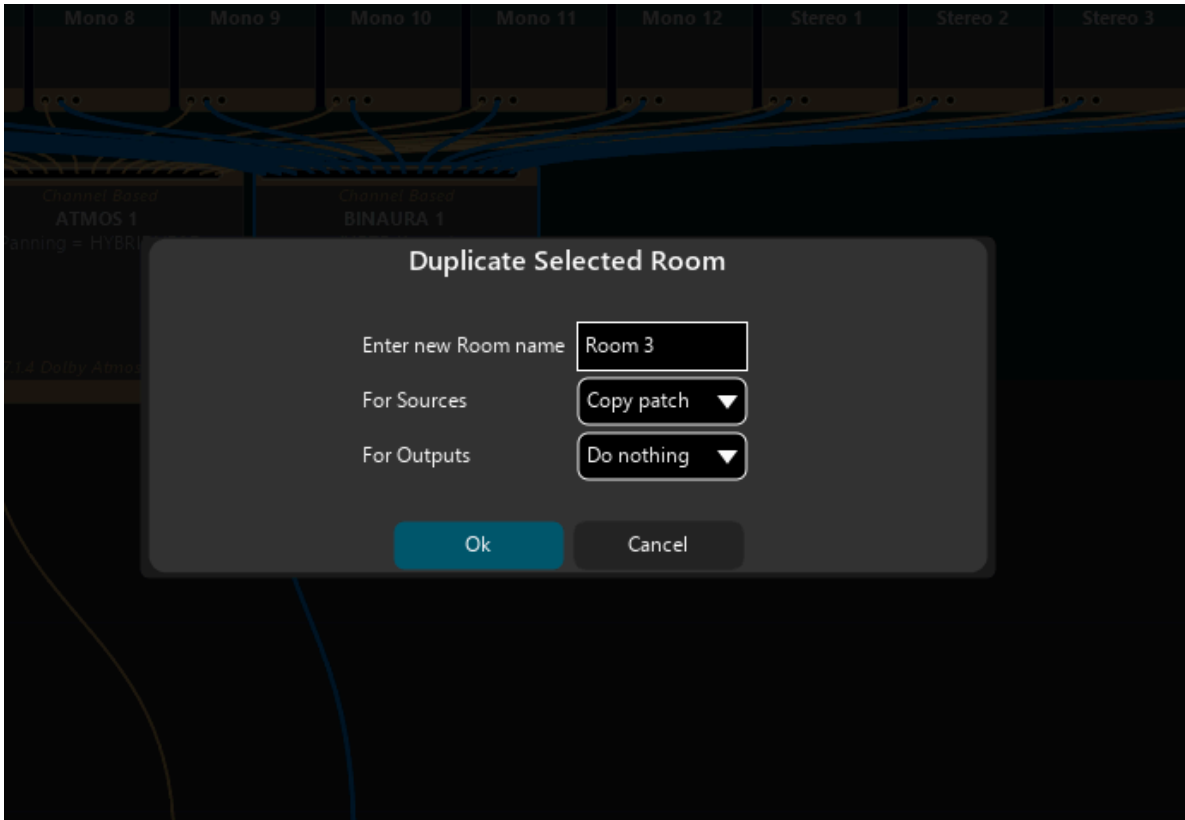
15.4.1 Remove selected

Remove the selected modules.

15.4.2 Duplicated selected

Duplicate the selected modules.

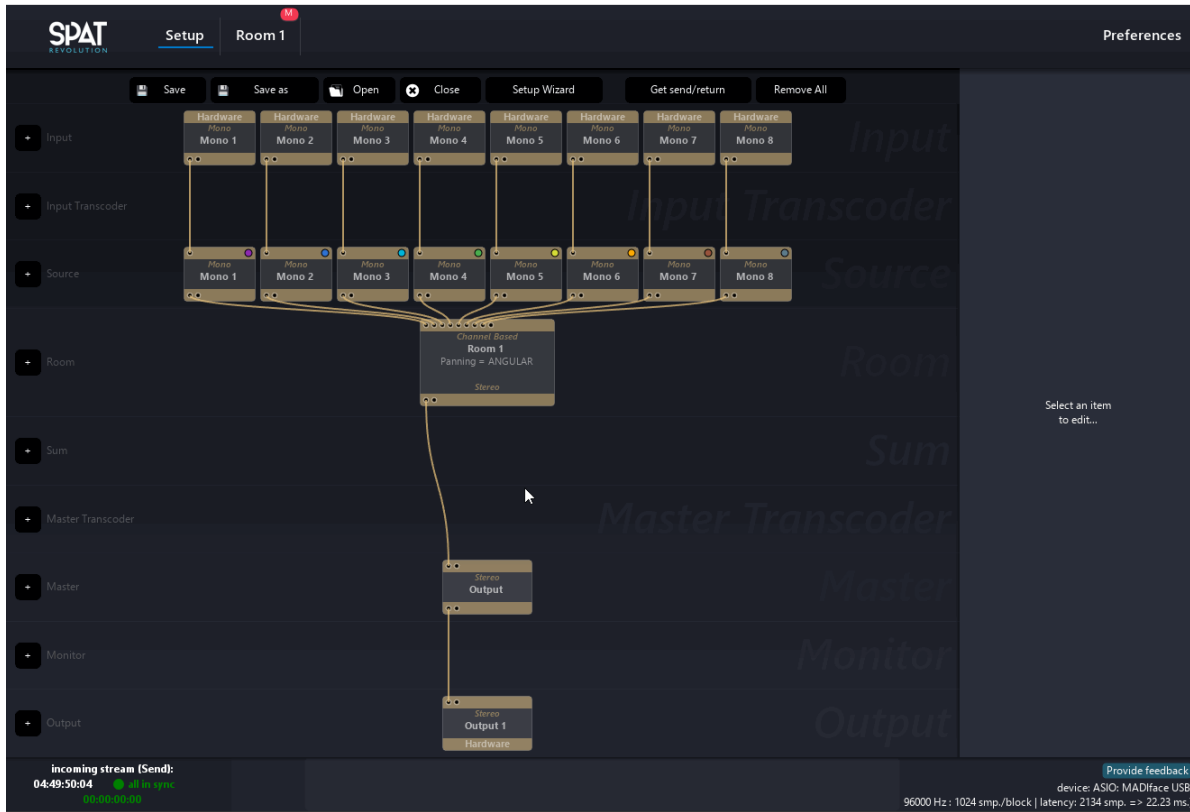
Room duplication



SPAT Revolution allows quickly duplicating a room with a few options to help the user to optimize the routing process. To access this menu, simply click on the **Duplicate Selected** button, when only a room is selected.

The new pop-up window allows to:

- Rename the duplicated room
- Choose if the sources routed to the original room are routed to the new one, or duplicated, or nothing is patched.
- Choose if the outputs of the original rooms are duplicated, mirrored or nothing is done to the duplicated room.



15.4.3 Connect selected

Connect selected modules. Modules can be connected to multiple destinations.

i Note

SPAT Revolution will always try to guess what you try to achieve. For example, you can connect an input block directly to an output block and *SPAT* will create the necessary modules in-between.

15.4.4 Disconnect selected

Disconnect **both** inputs and outputs of the selected modules.

15.4.5 Disconnect selected inputs

Disconnect connected inputs of selected modules.

15.4.6 Disconnect selected output

Disconnect connected outputs of selected modules.

15.4.7 Disconnect between selected

Disconnect the connections between selected modules.

15.4.8 Record arm

Arm the block for recording.

15.5 Routing Matrix

As you can imagine routing and patching high-density channel counts can get complicated. When it comes to that, the SPAT routing matrix is there to help. You will find it at many points throughout the **Environment Setup** graph.

i Note

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

The routing matrix is available on hardware input and output for routing as well as for remapping within some module's input and output: Input transcoder, Master, and Master transcoder.

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

i Note

The shortcut **Ctrl + click** will route one per one all the following channels.

Master transcoder and master matrix support summing, thus, one input can be connected to several outputs, or the opposite.

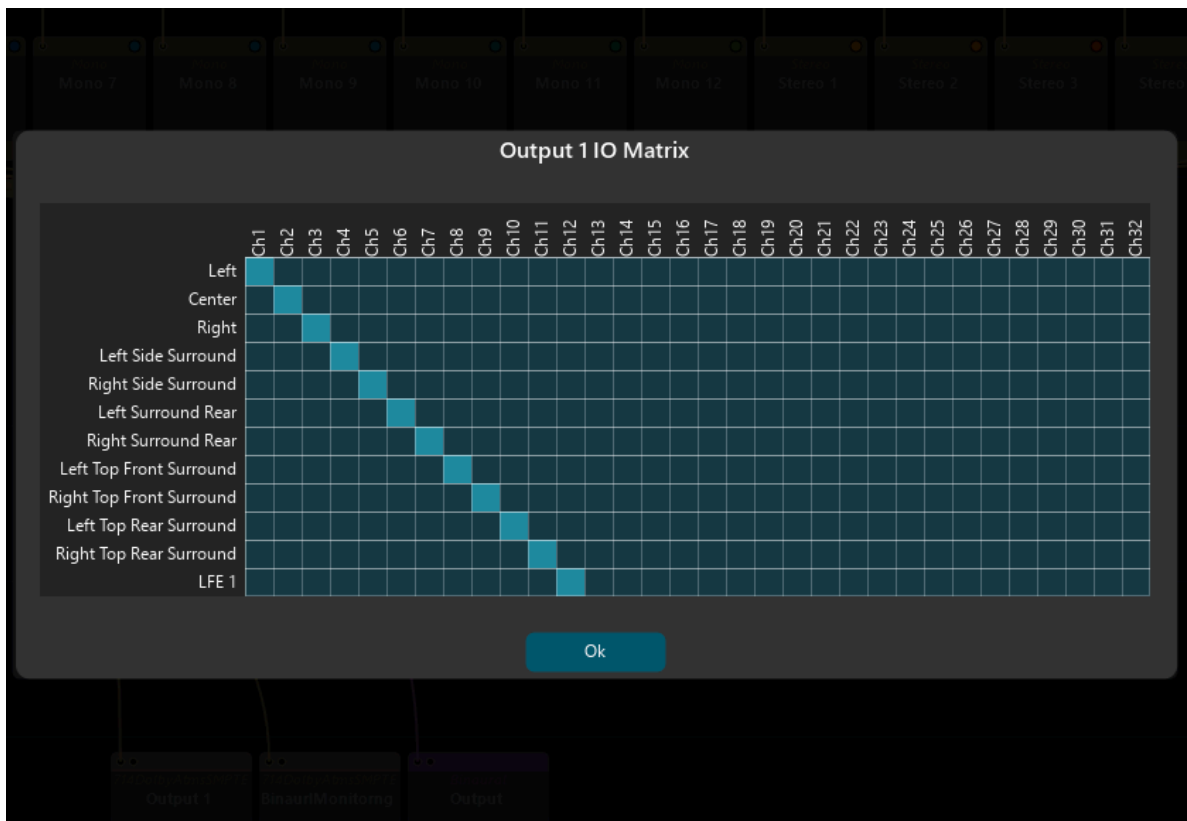


Figure 15.7: Output Matrix

16 Items' page

This page gives an overview and allows editing the major parameters for each item type (Input, Source, Room, etc.)

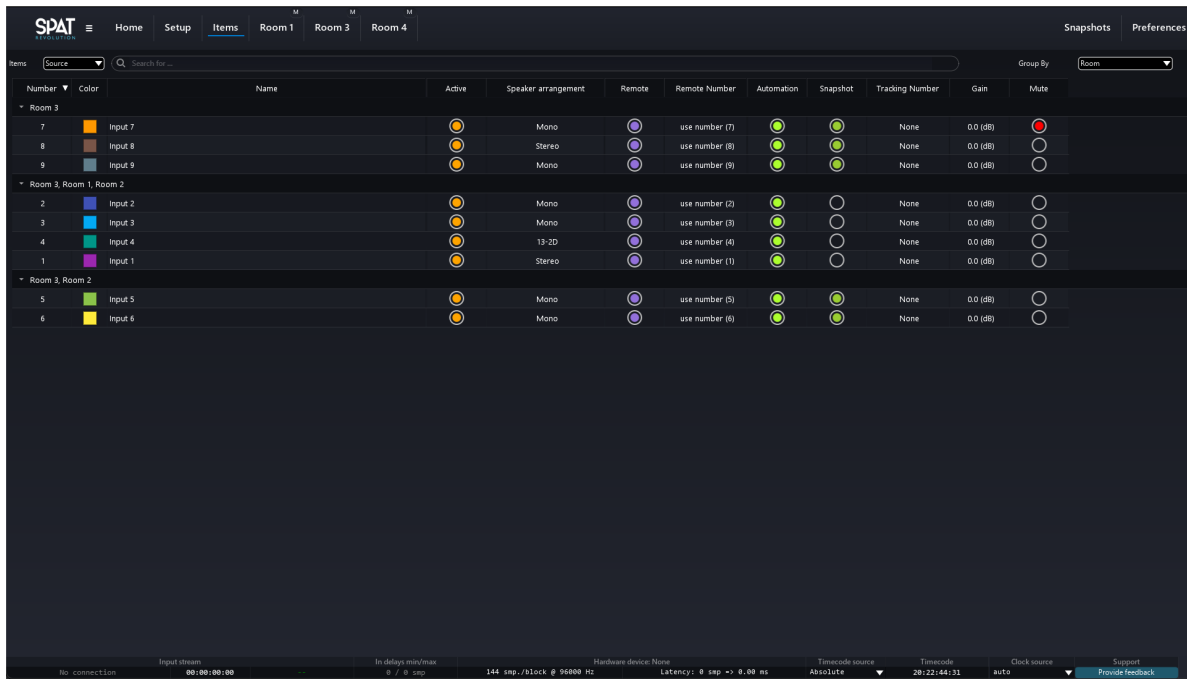


Figure 16.1: The items' page

The following video illustrates some possibilities that this page offers.

16.1 Items type selection

The upper left list control allows choosing which kind of items you want to display on the Items' Page.

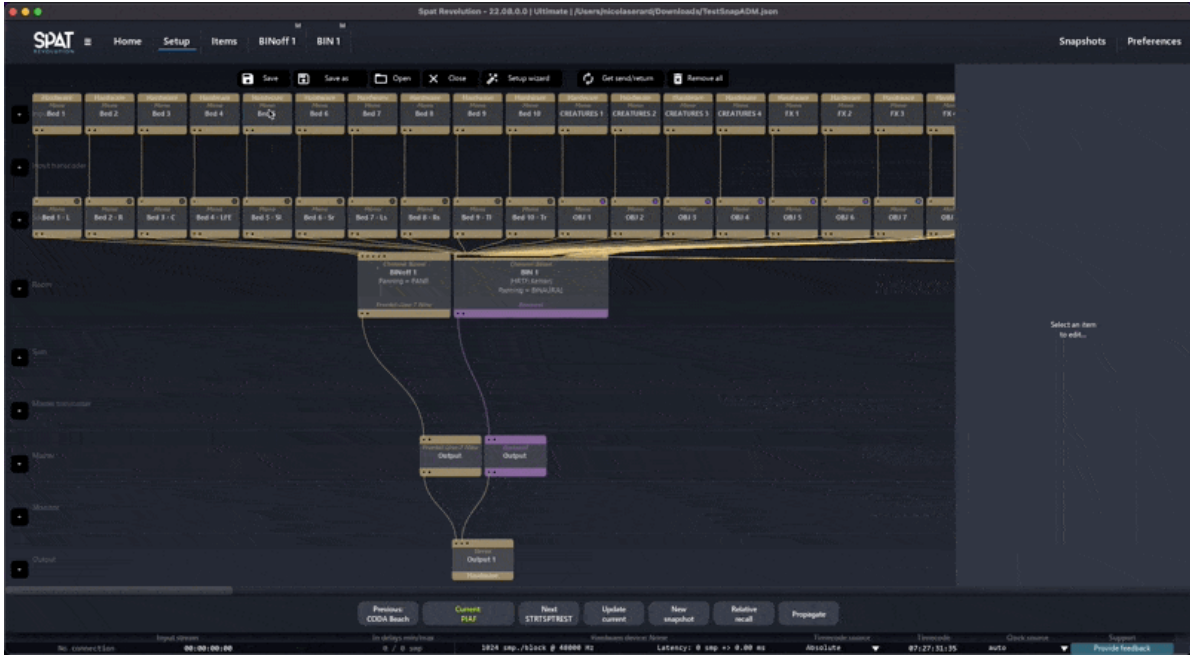


Figure 16.2: The Items' page demo

Num	Name	Active	Type	Stream type	Speaker arrangement	Delay (ms)	Delay (ms)	Delay (meters)	Delay (feet)	Connected	Send t.
1	Bed 1	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
2	Bed 2	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
3	Bed 3	●	Live Input	ChannelBased	Mono	216	4.5000	1.5452	5.0855	●	○
4	Bed 4	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
5	Bed 5	●	Live Input	ChannelBased	Mono	285	5.8333	2.0050	6.5716	●	○
6	Bed 6	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
7	Bed 7	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
8	Bed 8	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
9	Bed 9	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
10	Bed 10	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
11	CREATURES 1	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
12	CREATURES 2	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
13	CREATURES 3	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
14	CREATURES 4	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
15	FX 1	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
16	FX 2	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
17	FX 3	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
18	FX 4	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
19	FX 5	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
20	FX 6	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
21	FX 7	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
22	FX 8	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
23	FX 9	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
24	FX 10	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○
25	FX 11	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○

Figure 16.3: Items' selection

16.2 Parameters

Depending on the selected item's type, the editable parameters are the following.

16.2.1 Common parameters

For all items, the items' page allows quick edition of:

- **Number:** the item number order of the list.
- **Name:** edit this field to change the name of the item.
- **Active:** define if the item is computed or not. *If the item is inactive due to license restriction, the field cannot be edited.*
- **Record:** arm the block for recording.

16.2.2 Input specific parameters

In addition to the common parameters, when the *Input* item's type is selected, the Items' Page shows:

- **Type:** displays if the input is *Live input* or *Signal generator*.
- **Speaker arrangement:** select the speaker arrangement of the input.
- **Delay (smp):** define the input delay in sample.
- **Delay (ms):** define the input delay in ms.
- **Delay (meters):** define the input delay in meters.
- **Delay (feets):** define the input delay in feets.
- **Connected:** displays if the input is connected to other items.
- **Invert phase:** invert the phase of the selected item.

16.2.3 Source specific parameters

In addition to the common parameters, when the *Source* item's type is selected, the Items' Page allows quick editing of:

- **Color:** select the color of the source.
- **Room:** list the rooms' names the source belongs to. *The rows can be grouped by rooms. See Group sources by rooms below*
- **Group:** select the group in which you want to add the sources.
- **Speaker arrangement:** select the speaker arrangement of the source.
- **Remote:** define if the source can be controlled by OSC or not.

Num	Name	Active	Type	Stream type	Speaker arrangement	Delay (smp)	Delay (ms)	Delay (meters)	Delay (feet)	Connected	Send L...	Invert	Record
1	MONO 1	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
2	Violon 1-2	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
3	Mono 3	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
4	Mono 4	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
5	MONO 5	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
6	Violon 1-6	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
7	MONO 7	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
8	Violon 2-1	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
9	Violon 2-2	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
10	MONO 10	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
11	MONO 11	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
12	Violon 2-5	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
13	MONO 13	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
14	Alto 1	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
15	MONO 15	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
16	Alto 3	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
17	Mono 17	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
18	Mono 18	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
19	MONO 19	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
20	Cello 1	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
21	MONO 21	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
22	Cello 3	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
23	MONO 23	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
24	Cello 5	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○
25	MONO 25	●	Live Input	ChannelBased	Mono	0	0.0000	0.0000	0.0000	●	○	○	○

Figure 16.4: Items page for sources

- **Remote number:** define the index of the source on OSC side. *If set to 0, the remote number is equal to the source number. Careful with this behavior: this will be edited with the source order.*
- **Automation:** define if the source can be controlled by automation via Local Audio Path.
- **Snapshot:** define if the source can be controlled by snapshot recall.
- **RTTrPM number:** define the RTTrPM beacon number used for control this source. If set to 0, the tracking will be disabled. *Ultimate license only.*
- **Gain:** define the gain of the source.
- **Mute:** define the mute status of the source.
- **Invert phase:** invert the phase of the selected item.

16.2.4 Room specific parameters

In addition to the common parameters, when the *Room* item's type is selected, the Items' Page shows:

- **Connected sources:** display the number of connected sources.
- **Speaker arrangement:** select the speaker arrangement of the room. For HOA and Binaural room, this field displays respectively the dimension and the HRTF.
- **Panning type:** select the pan law of the *Room*, if stream type is Channel-Based or Binaural. For an HOA room, this field displays the HOA order.
- **Reverb enable:** define if the reverb is enabled in this room.
- **Protection zone width:** define the width of the protection zone [32.1](#).

Number	Color	Name	Room	Group	Active	Speaker arrangement	Remote	Remote number	Automation	Snapshot	RTT/PM Nu...	Gain	Mute	Send to an...	Invert phase
1	Red	Violon 1-1	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	1	●	●	None	1.9 (dB)	○	○	○
2	Red	Violon 1-2	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	2	●	●	None	1.9 (dB)	○	○	○
3	Red	Violon 1-3	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	3	●	●	None	1.9 (dB)	○	○	○
4	Red	Violon 1-4	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	4	●	●	None	1.9 (dB)	○	○	○
5	Red	Violon 1-5	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	5	●	●	None	1.9 (dB)	○	○	○
6	Red	Violon 1-6	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	6	●	●	None	1.9 (dB)	○	○	○
7	Red	Violon 1-7	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	7	●	●	None	1.9 (dB)	○	○	○
8	Red	Violon 2-1	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	8	●	●	None	1.9 (dB)	○	○	○
9	Red	Violon 2-2	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	9	●	●	None	1.9 (dB)	○	○	○
10	Red	Violon 2-3	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	10	●	●	None	1.9 (dB)	○	○	○
11	Red	Violon 2-4	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	11	●	●	None	1.9 (dB)	○	○	○
12	Red	Violon 2-5	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	12	●	●	None	1.9 (dB)	○	○	○
13	Red	Violon 2-6	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	13	●	●	None	1.9 (dB)	○	○	○
14	Blue	Alto 1	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	14	●	●	None	0.0 (dB)	○	○	○
15	Blue	Alto 2	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	15	●	●	None	0.0 (dB)	○	○	○
16	Blue	Alto 3	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	16	●	●	None	0.0 (dB)	○	○	○
17	Blue	Alto 4	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	17	●	●	None	0.0 (dB)	○	○	○
18	Blue	Alto 5	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	18	●	●	None	0.0 (dB)	○	○	○
19	Blue	Alto 6	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	19	●	●	None	0.0 (dB)	○	○	○
20	Blue	Cello 1	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	20	●	●	None	0.6 (dB)	○	○	○
21	Blue	Cello 2	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	21	●	●	None	0.6 (dB)	○	○	○
22	Blue	Cello 3	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	22	●	●	None	0.6 (dB)	○	○	○
23	Blue	Cello 4	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	23	●	●	None	0.6 (dB)	○	○	○
24	Blue	Cello 5	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	24	●	●	None	0.6 (dB)	○	○	○
25	Blue	Cello 6	WFS SYS 1.STEREO DM 1	None	●	Mono	⊗	25	●	●	None	0.6 (dB)	○	○	○

Figure 16.5: Items page for sources

- **Efficiency zone:** select the wanted behavior when a source is out of the efficiency zone 32.3 of the room.
- **Scaling distance:** define the scaling distance 29.3.5 of the room in meters.
- **Tracking scaling:** define the tracking scaling of the room in meters.
- **Invert phase:** invert the phase of the selected item.

Num.	Name	Connected sour...	Active	Stream type	Speaker arrangement	Panning type	Rever...	Protecti...	Efficiency zone	Scall...	Track...	Gain	Mute	Send...	Inver...	Record
1	Room 1	9	●	Binaural	HRTF: Kemar	SPHERICAL HEAD MODEL	4.00	CLAMP...y Zone	1.000	1.000	-2...B)	○	○	○	○	○
2	Room 2	9	●	ChannelBased	Stereo	ANGULAR	4.00	CLAMP...y Zone	1.000	1.000	0.0..B)	○	○	○	○	○
3	Room 3	9	●	ChannelBased	5.1 - L-R-C-LFE-Ls-Rs	AMBISONIC EQUIVALENT	4.00	CLAMP...y Zone	1.000	1.000	-2...B)	○	○	○	○	○

Figure 16.6: Items page for rooms

16.3 Navigation

In order to give easy navigation on this page and its parameters, keyboard shortcuts have been implemented, including powerful multi-selection and edition.

- **Up:** select the cell of the previous row.
- **Shift + Up:** add the previous row to the row selection.
- **Down:** select the cell of the next row.
- **Shift + Down:** add the next row to the row selection.

- **Shift + click**: select all the rows in between the two selected
- **Ctrl + click**: select only the wanted rows.
- **Left key**: select the previous cell.
- **Right key**: select the next cell.
- **Space** on an on/off button: it will toggle the button for the actual selection.
- **Enter** (return) on a string or float edit, or a list: it will begin the edit of the selected cell. Using it a second time will edit the cell of the next row.
- **Home**: go to the first row.
- **End**: go to the last row.
- **Page Up**: select the cell on the previous page.
- **Page Down**: select the cell on the next page.

It is also possible to sort the items according to a specific field, shift-clicking on the header of it.

16.4 Filtering and Group by

To go further in easing the edition of parameters, the Items' page allows to filter of the displayed data. Moreover, for sources, they can be grouped according to the rooms they belong to.

16.4.1 Filtering

Enter text in the top search field to filter the displayed objects. This will search on all the fields. To search for a specific field, type the name of the field and add ":". For example, to search the object with the name beginning by "Violin", type `name:Violin`.

16.4.2 Group sources by rooms

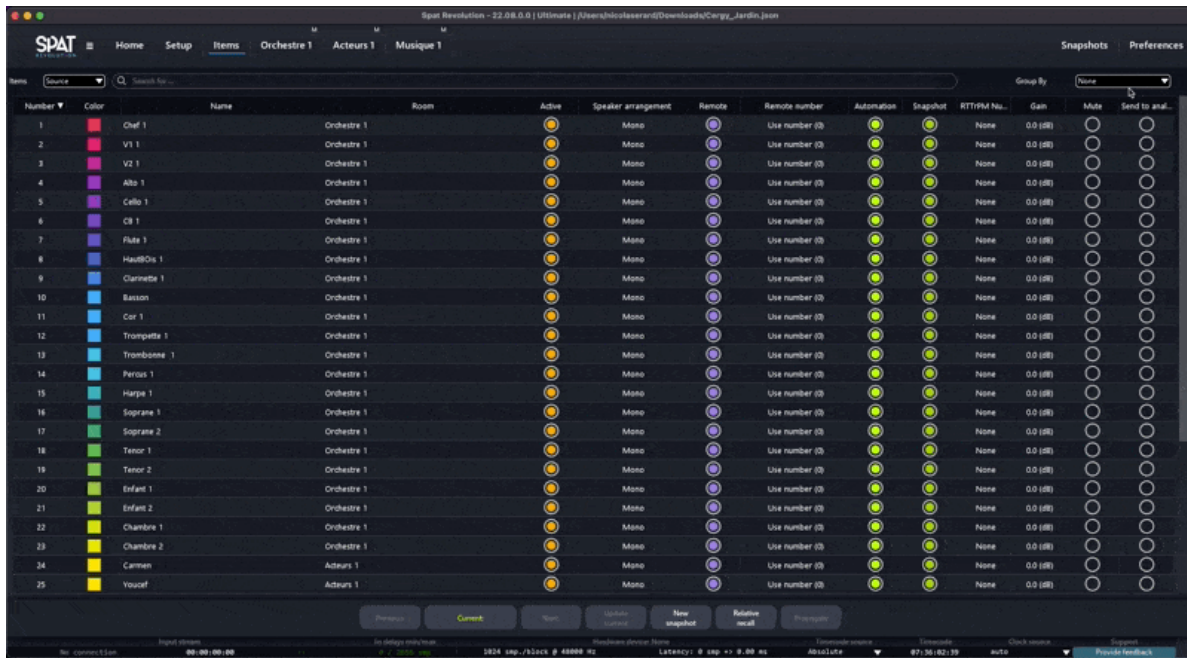


Figure 16.7: Group by rooms

17 Snapshots page

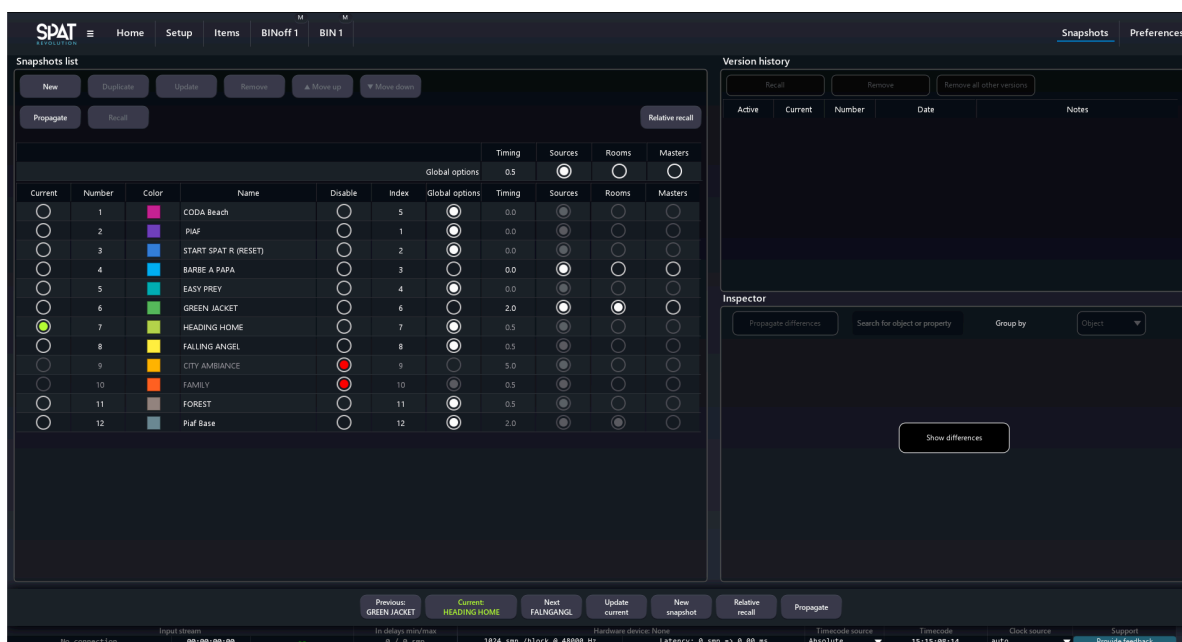


Figure 17.1: Snapshots page

The new Snapshots page of *SPAT Revolution* allows the user to deeply manage its show. Snapshots are an easy way to create different session states. A classic example would be to have different snapshots for different songs, or sections of a song, in a live show.

The system built in *SPAT Revolution* is both simple and powerful. As a generality, when a snapshot is created, it stores the mix parameters, i.e. the source, room and master parameters. All parameters are always stored, but it is possible to filter the recalled parameters.

The page is divided into three different sections: the snapshot list, the versions' history and the inspector.

							Timing	Sources	Rooms	Masters
Global options							0.5	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Current	Number	Color	Name	Disable	Index	Global options	Timing	Sources	Rooms	Masters
<input type="radio"/>	1	■	CODA Beach	<input type="radio"/>	5	<input checked="" type="radio"/>	0.0	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	2	■	PIAF	<input type="radio"/>	1	<input checked="" type="radio"/>	0.0	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	3	■	START SPAT R (RESET)	<input type="radio"/>	2	<input checked="" type="radio"/>	0.0	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	4	■	BARBE A PAPA	<input type="radio"/>	3	<input type="radio"/>	0.0	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
<input type="radio"/>	5	■	EASY PREY	<input type="radio"/>	4	<input checked="" type="radio"/>	0.0	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	6	■	GREEN JACKET	<input type="radio"/>	6	<input type="radio"/>	2.0	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
<input checked="" type="radio"/>	7	■	HEADING HOME	<input type="radio"/>	7	<input checked="" type="radio"/>	0.5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	8	■	FALLING ANGEL	<input type="radio"/>	8	<input checked="" type="radio"/>	0.5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	9	■	CITY AMBIANCE	<input checked="" type="radio"/>	9	<input type="radio"/>	5.0	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	10	■	FAMILY	<input checked="" type="radio"/>	10	<input type="radio"/>	0.5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	11	■	FOREST	<input type="radio"/>	11	<input checked="" type="radio"/>	0.5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	12	■	Piaf Base	<input type="radio"/>	12	<input checked="" type="radio"/>	2.0	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 17.2: Snapshot list

17.1 The snapshot list

17.1.1 Generality

The snapshot list is the main section of snapshot management and serves many purposes. First, it displays the snapshot organization of your session. The snapshot name can also be edited by clicking on the text field. To recall a snapshot from the snapshot list, simply double-click on the number of one of them.

Snapshot selection

A snapshot can have different states. First, it can be selected or unselected inside the list. It will serve to make a specific action on the selected snapshot(s) and reveals the versions' history and inspector section (if only one is selected). The state of selection is a graphical element that let the user know on which snapshot an action is performed.



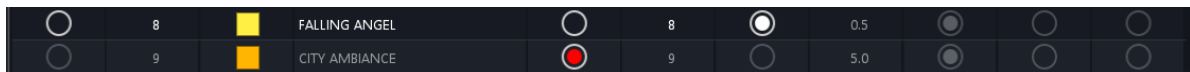
Unselected snapshot



Selected snapshot

Snapshot activation and deactivation

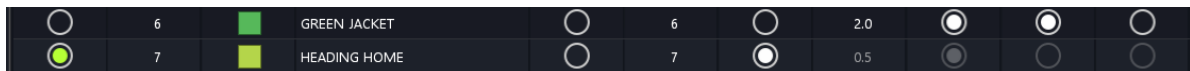
A snapshot can also be disabled using the dedicated toggle. A disabled snapshot will be darkened in the list. It cannot be recalled. The **next** and **previous** snapshot buttons on the snapshot toolbar will automatically skip a disabled snapshot.



Activated snapshot (1) VS. deactivated snapshot (2)

Current snapshot

Lastly, the last recalled snapshot is displayed with the “current” field set on. This allows the user to monitor where he is in his show.



In-play snapshot (2) VS. not-in-play snapshot (1)

17.1.2 The actions

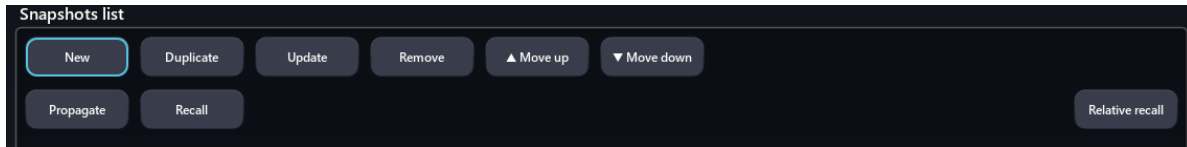


Figure 17.3: Action bar

This snapshot list's top bar regroups all the snapshot actions:

- **New** allows creating a new snapshot, storing the current sources, rooms and masters state.
- **Duplicate** allows duplicating the current snapshot selected in the snapshot list.
- **Update** allows updating the selected snapshot in the snapshot list with the current state of the session. It will create a new snapshot version.
- **Remove** allows deleting the snapshot(s) selected in the snapshot list.
- **Move up** moves the selected snapshot(s) up one row in the snapshot list.
- **Move down** moves the selected snapshot(s) down one row in the snapshot list.
- **Recall** allows loading a previously saved snapshot selected in the snapshot list.

Warning

Keep in mind that a source with “Snapshot” parameter set off will be stored, but not recalled. Please check the “item page” section for more information.

17.1.3 Snapshot Index

To recall through OSC a snapshot, the name or the index is required. As snapshot order can be changed, the snapshot index does not depend on the snapshot position in the list. The snapshot index is indicated on each snapshot, on the left side of the recall options.



Figure 17.4: Snapshot index

It is possible to generate new indexes for all the snapshots according to list order. To do so, click on the menu **Snapshot/Reindex all snapshots**.

Last, the index can be manually edited, by right-clicking on a snapshot and selecting **Edit snapshot index**.

The action **Reindex Snapshots**, found in the upper **Snapshot** menu reindex all the snapshots, starting from 1 and by their order in the snapshot list.

17.1.4 Recall options

Recalling a snapshot will use four different options: - The **timing** is the transition time between the current session state and the session state recalled from a snapshot. This allows smoothing out the transition between two scenes and can also be used to create some movements. - The **source option** defines if the sources state should be recalled. This refers to the sources' position and other properties. - The **room option** defines if the rooms' state should be recalled. This refers to the reverb parameters of rooms and also the listener's head. Be careful with this as some parameters of the reverberation can't be recalled without sound dropping. - The **master option** defines if the masters' state should be recalled. This refers to the master level output.

i Note

By default, only sources' properties are recalled from snapshots. If only some parameters are recalled, the whole session is stored on the creation of a snapshot. For rooms, the setup parameters like distance scaling, protection zone parameters, etc. are deactivated from snapshot recall.

17.1.5 Using global options or overriding them

The snapshot list exposes the four global recall options in its top right corner: the recall timing, the sources recall option, the rooms recall option and the masters recall option. These parameters affect each snapshot with global options activated.

Beside each snapshot, there is a checkbox, under a column named “**Global options**”. If the checkbox is checked, the default values of the recall preferences refer to the global values. If the checkbox is unchecked, it will override the global preferences and use the specific recall values for the snapshot.

17.1.6 Relative recall option

Relative recall is an option of the recall function. It allows recalling a snapshot while preserving anything that was offset from the previous state. This can help to prevent technical problems during a show.

Here is an example: let's say you have two snapshots, *A* and *B*. Sources inside your project have different gains between the two snapshots: 0 dB for snapshot *A*, -5 dB for snapshot *B*. Now, let's presume that once the show is started, you feel that one of the sources is too quiet (the singer preserves his voice). So you grab the gain parameter and trim by 4 dB. The “Relative recall” function will preserve this offset to any of the future recalled snapshots.

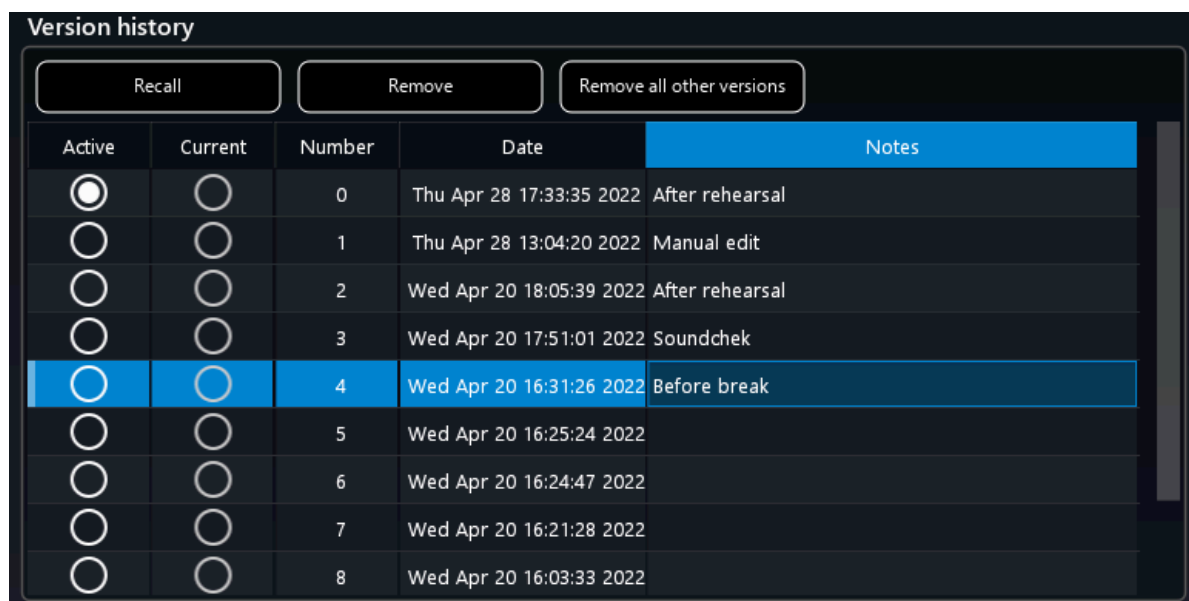
Recalling snapshot *B* will set the gain of the singer to -1 dB instead of -5dB. This allows a perfect blend of live mixing and preparation work.

17.1.7 Changing snapshots' color

To help to a better organization of the session, snapshots can be recolored. To do so, right-click on the selected snapshot and choose *Set colors*. A palette, from which you can pick a color, will appear.

It is also possible to *Generate colors* automatically for a multiselection of snapshots. Right-click on one of them and click on the corresponding list entry.

17.2 Version history



Active	Current	Number	Date	Notes
<input checked="" type="radio"/>	<input type="radio"/>	0	Thu Apr 28 17:33:35 2022	After rehearsal
<input type="radio"/>	<input type="radio"/>	1	Thu Apr 28 13:04:20 2022	Manual edit
<input type="radio"/>	<input type="radio"/>	2	Wed Apr 20 18:05:39 2022	After rehearsal
<input type="radio"/>	<input type="radio"/>	3	Wed Apr 20 17:51:01 2022	Soundchek
<input type="radio"/>	<input checked="" type="radio"/>	4	Wed Apr 20 16:31:26 2022	Before break
<input type="radio"/>	<input type="radio"/>	5	Wed Apr 20 16:25:24 2022	
<input type="radio"/>	<input type="radio"/>	6	Wed Apr 20 16:24:47 2022	
<input type="radio"/>	<input type="radio"/>	7	Wed Apr 20 16:21:28 2022	
<input type="radio"/>	<input type="radio"/>	8	Wed Apr 20 16:03:33 2022	

Figure 17.5: Snapshot version history

On this panel takes place a very powerful feature. For each snapshot, SPAT Revolution automatically stores the last versions. By default, **ten versions** are stored. It is possible to increase or decrease this number on the Snapshot panel of the **Preferences** page. Of course, you can recall any of these ten previous states, or remove one.

An entry in the version history has for indication its of creation date and time. To describe the version, you can insert a custom note.

The **Active** field allows choosing which version will be used when the snapshot is recalled. Only one version can be activated.

A set of actions is presented at the top: - A previous snapshot version can be recalled by clicking on the **Recall** button. The recalled version will be indicated by the **Current** field. - Versions can be deleted, by clicking on **Delete**. - All versions excluding the selected one(s) can be deleted, clicking on **Delete all other versions**.

The action **Consolidate Snapshots**, found in the upper **Snapshot** menu remove all the unused snapshot version to lighten the weight of the session file.

17.3 Inspector

The inspector lets you visualize the difference between the current state of your session and the selected snapshot, and the selected version of a snapshot.

It allows easy monitoring of what was changed and how it was changed.

Property	Snapshot value	Current value
Object: BINoff 1		
DistanceScale	1.000	2.500 (+1.500)
ZoneProtectionW	6.000	15.000 (+9.000)
ZoneProtectionD	6.000	15.000 (+9.000)
ZoneProtectionH	6.000	15.000 (+9.000)
ZoneEffD	95.657	95.000 (-0.657)
Object: Bed 1 - L		
PositionDistance	5.000	7.071 (+2.071)
DropFactor	0.00	6.00 (+6.00)
Object: Bed 2 - R		
DropFactor	0.00	6.00 (+6.00)

Figure 17.6: Inspector

Select the snapshot you want to compare, and click on the **Show differences**, or use the shortcut **Ctrl + D** on Windows, **Command + D** on macOS.

You can group the list by an object name or a property name, or research for objects or properties.

17.4 Propagate through snapshots

In order to edit a large number of snapshots, a propagation system has been implemented into the new **Snapshot Page**.

There are two possibilities to propagate values: - Propagate the differences between the *current state* and the *current snapshot*.

This behavior happens by clicking on the **Propagate** button located on the **Snapshot bar**, or on the top of the snapshot list of the **Snapshot Page**. - Propagate the differences between the *current state* and the *selected snapshot and version*. It is deeply linked to the **Snapshot inspector**. The **Propagate differences** action is located on it.

It is possible to propagate the selected differences or all the differences (if no differences are selected). To select differences, use **Ctrl + Click** on Windows, or **Cmd + Click** on macOS to add several items, or **Shift + Click** to select all between two items.

17.4.1 Propagate dialog

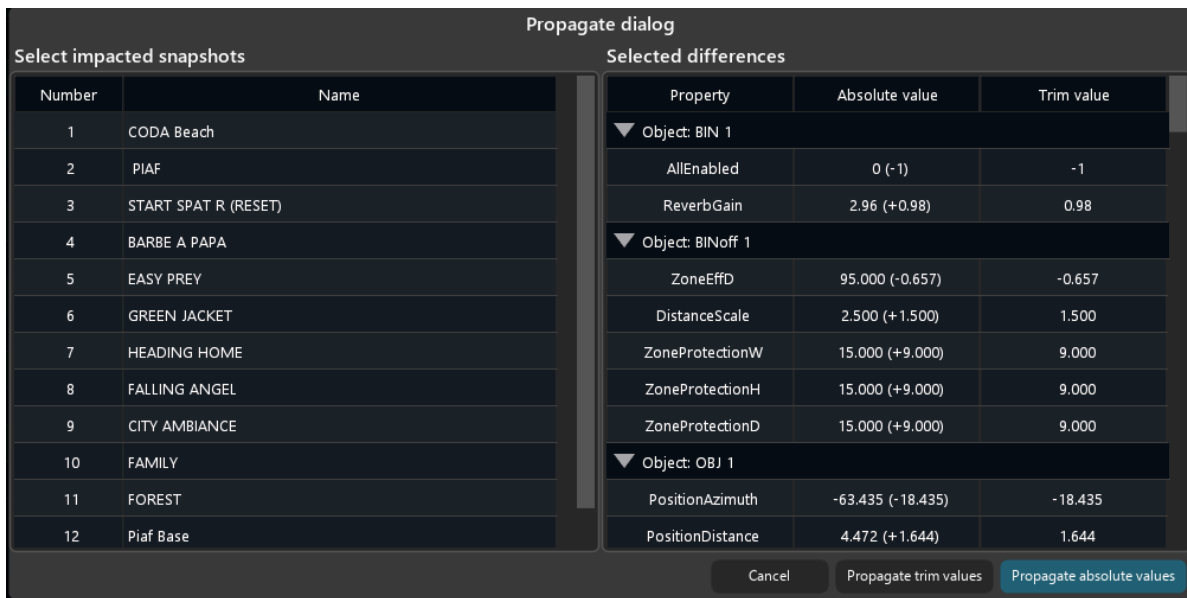


Figure 17.7: Inspector

On the propagate dialog, the selected data will be retrieved, displaying the absolute value which can be applied when validating on `Propagate absolute values` and the trim value which will be added to each snapshot value when validating with `Propagate trim values`.

i Note

For example, if my Accordion has an azimuth value of 50.0, it will become 13.475 propagating with absolute value, and 50.22 propagating with trim value.

On the left panel, the snapshot list is displayed in order to choose the snapshot for which the propagation will be applied.

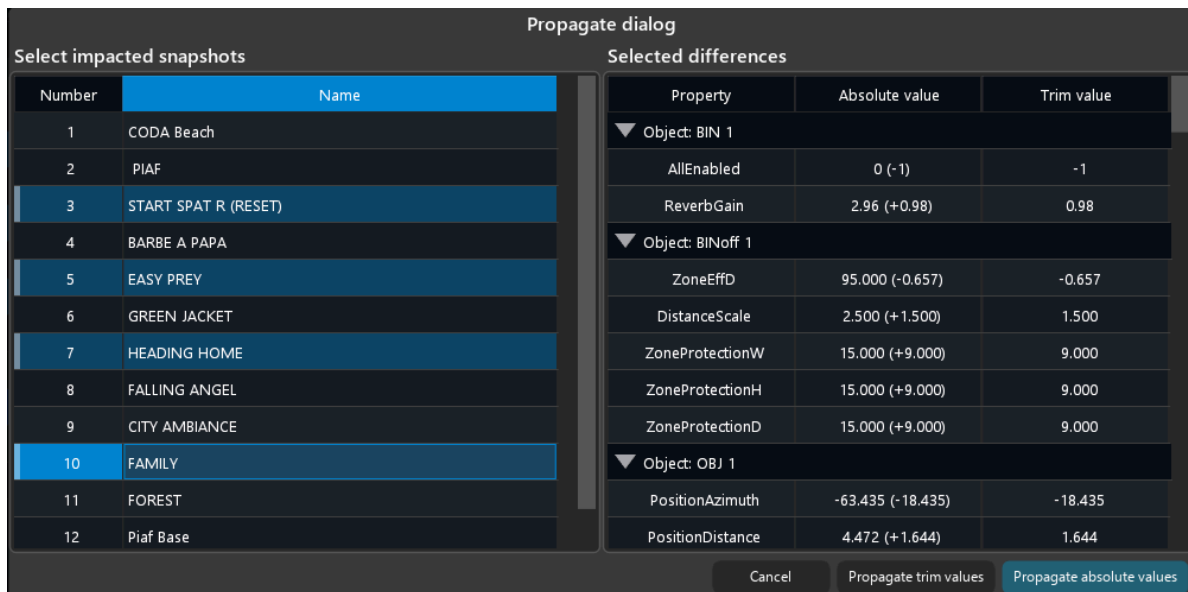


Figure 17.8: Inspector

On the right panel, the differences are displayed. It is possible to select them in order to filter which ones will be propagated. Do not select rows to propagate all the values.

18 Speaker Arrangement Editor

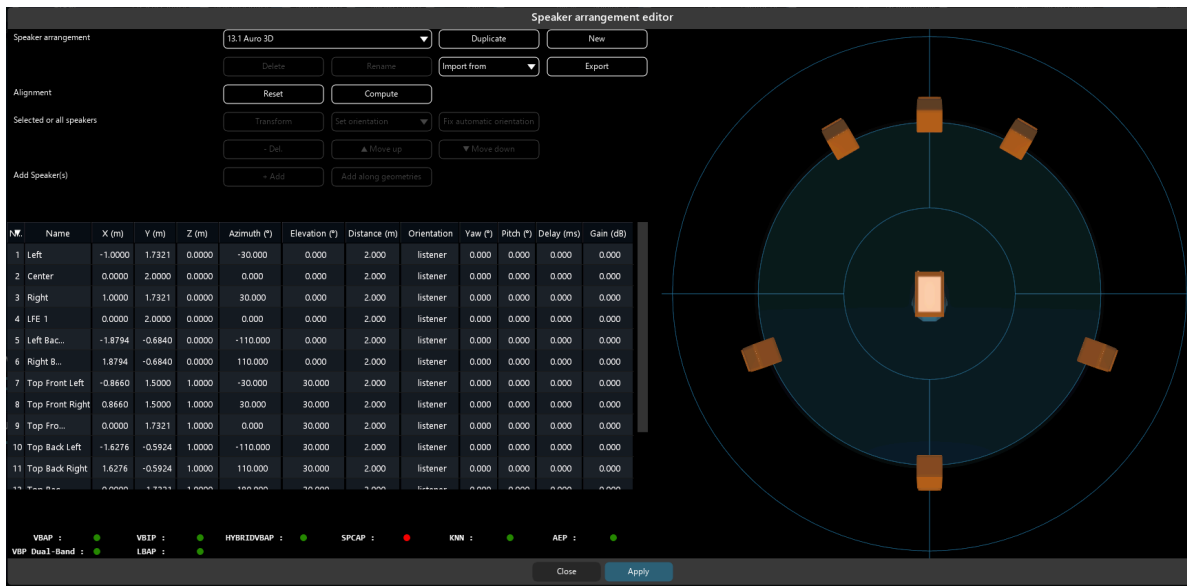
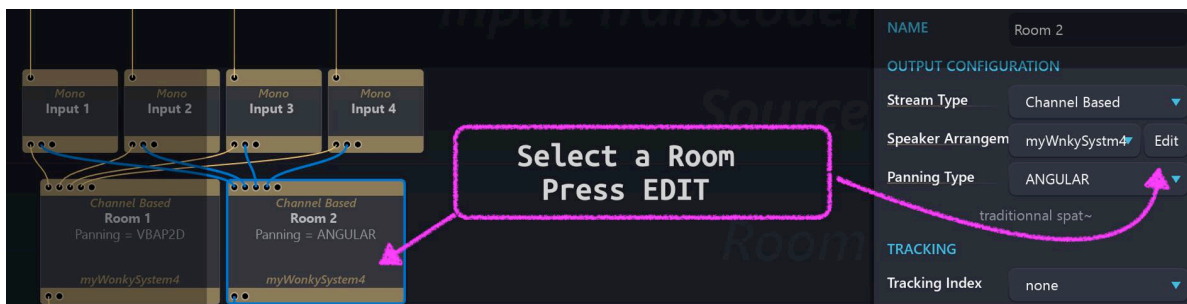


Figure 18.1: The speaker arrangement editor window showing a predefined 13.1 Auro 3D speaker arrangement.

The **speaker arrangement editor** offers the ability to prepare the model of the sound diffusion system you are actually using.



When you find yourself working on a custom multi-channel speaker arrangement, the **Speaker Config** editor is where a model of the sound diffusion system can be defined and stored in the list of speaker arrangements.

Open the **speaker arrangement editor** by clicking on the **Edit** button of a Channel-Based room, by the *Speaker Arrangement* pull-down menu. Pre-configured arrangements cannot be deleted or modified but can be **duplicated** (a copy will be generated for editing). A **new** config can also be created.

18.1 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 *multichannel speaker arrangement* which will be used to map the multichannel 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 specialized 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.

18.2 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 an 18 (**Essential**) or unlimited (**Ultimate**) speakers array and anything in between. In some of the more virtualized 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. ->

18.3 Creating & editing a speaker arrangement

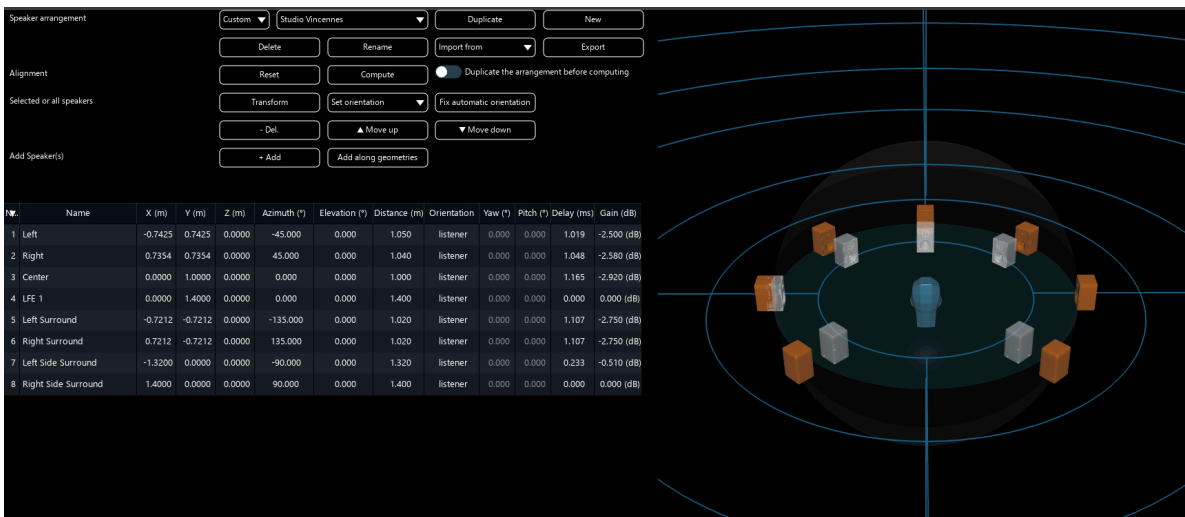


Figure 18.2: SPAT Revolution_ Speaker Config

While there are many standardized speaker arrays, such as Dolby Surround, Atmos, Auro 3D, DTS, Quad or even stereo, we are in many cases confronted with a very specific speaker

positioning.

For such a use case, it is possible to create custom speaker arrangements. Using a well-defined custom speaker arrangement will improve the sound stage and the overall impression of immersion of the audience. We always highly recommend taking the time to measure the position of the speakers of your system and to input them in *SPAT Revolution*.

18.3.1 Speaker Arrangement section

Arrangement drop-down menu: this menu allows access to a list of all your existing speaker arrangements.

Duplicate: create a copy of the current speaker arrangement.

 Warning

You cannot overwrite a factory arrangement. Duplicate it before trying to edit it.

New: create a new empty speaker arrangement.

Delete: delete the current speaker arrangement.

 Warning

There is no confirmation before deleting an arrangement.

Rename: rename the current speaker arrangement.

Import from: allow importing a speaker configuration from a third-party software. See the next section [18.5](#) for more detail.

Export: open a window to export a selection of speaker arrangements.

18.3.2 Alignment section

Reset: revert the *compute* process.

Compute: opens a dialog to help to set delay and gain on the speaker arrangement. More information of the dedicated section below [18.4.1](#)

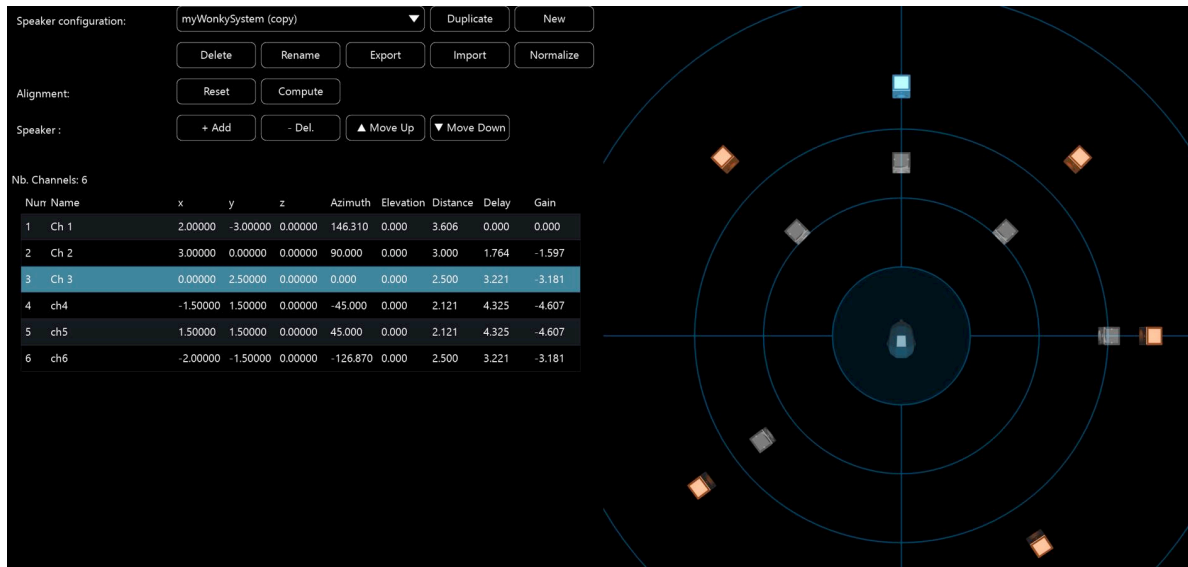


Figure 18.3: *SPAT Revolution* Speaker Config

18.3.3 Selected or all speakers section

Transform: modify the selected or all speakers according to a specified transformation. For more information about it, check out the transformation section [19](#).

Set orientation: set the orientation of selected or all speakers.

Fix automatic orientation: for speakers whose orientation is set to automatic, replace it to real orientation (front, back, listener, custom with the given angle, ...) instead of automatic.

Del.: remove the selected speaker(s) in the arrangement.

Move up: move the selected speaker(s) one row above.

Move down: move the selected speaker(s) one row below.

18.3.4 Add speaker(s) section

Add: create a new speaker in the arrangement.

Add along geometries: add new speakers according to a specified geometries.

18.3.5 The speakers list section

Nb.: identify the channel of the speaker.

Warning

The *SPAT Revolution Essential* license limits the total number of channels to 32 input channels and to 18 output channels.

Name: identify the name of the speaker.

Warning

To create an LFE channel, simply name one speaker of the arrangement “LFE”. It is possible to create 4 LFEs, naming them “LFE 1”, “LFE 2”, “LFE 3” and “LFE 4”. The corresponding sends button will appear on the source parameters panel. SPAT Revolution uses speaker naming to sum speaker arrangement.

X (m): defines the position of the speaker on the X-axis, in meters.

Y (m): defines the position of the speaker on the Y-axis, in meters.

Z (m): defines the position of the speaker on the Z-axis, in meters.

Azimuth (°): define the angle of incidence of the source to the center of the 3D space, in the horizontal plane.

Elevation (°): define the angle between the source, the center of the 3D space and the horizontal plane.

Distance (m): define the distance between the source and the center of the 3D space, in meters.

Orientation: define the orientation of the speaker. The default settings is “Automatic”

- Automatic: default behavior, the yaw and the pitch angles are determined by the shape of the speaker arrangement. It should be the standard orientation of most cases, but can be inefficient on certain specific designs.
- Listener: the speaker points to the listener’s head
- Front: the speaker points to the bottom of the 3D view
- Back: the speaker points to the top of the 3D view
- Side Left: the speaker points to the left side of the 3D view
- Side Right: the speaker points to the right side of the 3D view
- Custom: the user can define the orientation with the *yaw* and *pitch* parameters.

Note

The orientation (0°,0°) is pointing to the listener.

Yaw: rotate the speaker around the Z-axis. 0° degree is pointing to the listener. **Pitch (°):** rotate the speaker in elevation.

Warning

Orientation, yaw and pitch are only useful in WFS use case.

Note

Tips: to point down (speaker with positive elevation) or up (speaker with negative elevation) a speaker, set the pitch to 90 minus the speaker elevation value;

Delay (ms): add a fixed amount of latency to a speaker.

Gain (dB): change the sound level of the speaker.

Warning

In many cases, it is wiser to let the **delay** and **gain** parameters untouched, and let the “compute” function do the job.

18.3.6 Panning Type Tips section

Here, we can find information about which pan law is available in regard to our speaker array. Some explanations are displayed to help understand why some pan laws aren't available.

XY :	●	AB :	●	STEREOPAN :	●	SURROUND :	●	PANR :	●
ANGULAR :	●	SPCAP :	●	CSP :	●	VBAP :	●	VBIP :	●
DUALBANDVBP :	●	DBAP :	●	KNN :	●	HOA :	●	LBAP :	●
WFS :	●								

There are five colors associated with the possible panning types:

- **Green:** this panning type is valid and functional with the selected speaker array.
- **Orange:** this panning type is somewhat functional, but there certainly is a better solution available. Hovering the panning type with the mouse will display a message to help improve the arrangement.
- **Red:** this panning type is not functional with the selected speaker array. Hovering the panning type with the mouse will display a message to explain why it does not work.
- **Grey:** Two speakers are located at the same spot (i.e. the speakers are coincident). The speaker arrangement is incorrect.



i Note

Error displayed with HOA panning type.

For more information about each pan law, check out the section [Panning algorithms 8](#).

18.4 Understanding when a speaker arrangement, or a sound scene, is 2D or 3D

We can quickly be tempted to create very precise speaker arrays, including height information where it may not be necessary and can do more harm than good.

The rule of thumb is to think in plan and not in absolute height. If speakers are meant to reproduce the same sound plan, they should have the same height. Also, the main sound plan has to be set at a height of zero meters. If you try to validate a speaker arrangement that only has one plan and is not set to a height of 0, it will correct it automatically. It also applies to older sessions that were created before *SPAT Revolution* had this feature.

18.4.1 Alignment

Delay and Gain settings

Method Cartesian with symmetry ▼

Align left/right and front/back speakers' lines symmetrically.

Apply on speakers All
 front

Additional parameters

Align listener oriented speakers on a sphere

Cancel Compute

The speakers alignments section lets *SPAT Revolution* compute the delay and gain for the selected speaker arrangement. To help to deal with them with complex systems, several methods are provided, allowing adapting your alignment according to the wanted behavior:

- **Spherical:** will align the speaker according to a sphere. The farthest speaker will determine the sphere radius. *This is the old behavior of the compute feature.*
- **Cartesian with minimum delay:** will align the speakers according to straight lines.
- **Cartesian with symmetry:** will align left/right and front/back speakers according to symmetrical straight lines. *A typical use case will be to help to deal with an asymmetrical arrangement.*

Alignment can be made on all arrangement, but also selecting a certain speaker orientation type. It is possible to select the speakers on each the alignment will be done according to their orientation.

18.4.2 Automatic Scaling

When going on the road with a show, the idea of *SPAT Revolution* is to always keep the same session, which stores our input routing, our mixing metadata and all of our snapshots. But in such use cases, each time the venue changes, the speaker arrangement changes too and its size (or diameter) will likely be different each time.

As we use absolute, we solve this common use case by using an automatic scaling process.

If you change the speaker arrangement of a room where you already routed some sources, the global sound scene could be scaled to the new speaker arrangement.

For example, if you first created your session with a speaker arrangement with a diameter of four meters, and then deploy it in a venue where the actual diameter of the sound system is eight meters, then all the source positions, incoming automation, and snapshot data will be scaled by a factor of two.

18.4.3 Speaker arrangements examples

Here are some examples of speaker configurations you can find in some venues, like RadioFrance Studio 115, Montreal Satosphere, or Berlin Planetarium. To discover it into *SPAT Revolution*, copy the python file on your desktop: [Speaker arrangements for various domes and planetariums](#).

18.4.4 Speaker arrangement Python file

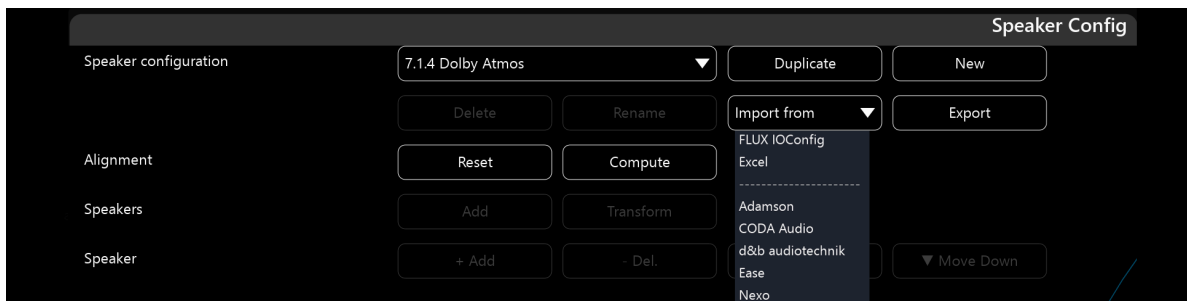
Writing the speaker arrangement on a Python file provides the erasing of the data. It is recommended to do so on fixed installation, or if you don't want to lose the speaker arrangement data. If you want to build your own custom speaker arrangement script, explanations are provided in the following Python file: [Custom Speaker Arrangement Script](#).

18.5 Import Speaker Array

For systems that are regularly changing such as in live production, setting up the speaker configuration in the simulation software and then repeating it in the immersive software is not the most fun part!

To ease this part of the setup, you can now import from the software of Nexo (NS-1), Adamson (Blueprint), d&b (ArrayCalc), CODA Audio (System Optimiser), and the standard EASE software.

This import is integrated into the **Speaker Arrangement** editor where you can find the **Import from** button.



To begin is the FLUX IOConfig. This is the FLUX:: speaker arrangements file format. This can be a great tool to export and save your FLUX:: arrangements or to import an arrangement into another session / computer system.

In *SPAT Revolution*, it is important to understand that speaker arrangements **DO** follow the session file. So if you are opening a .JSON with an arrangement new to your system, *SPAT Revolution* will copy that arrangement and it will then be part of the user-created arrangements. In consequence, this new arrangement will be accessible for any new *SPAT Revolution* session.

If you care to know where these arrangements are stored, Document/FLUX SE/Spat Revolution/Config is where your master IOConfig file, containing all the user-defined arrangements, is located. Note that the repertory Document/FLUX SE/Spat Revolution is now the folder that contains all of your SPAT preferences. If you ever need to erase your

preferences, for troubleshooting purposes, remember to export your speaker arrangements first, and then back them up outside this folder.

18.5.1 How to import a speaker configuration from

Ease

Since 23.08, XLD files coming from Ease are supported and recommended. To export such a file, in EASE software, open the project, and click on **File, Export Ascii Files, Loudspeakers**. The .xld created file can now be imported into SPAT Revolution

Legacy (.txt file): In EASE software, open the project, and click **Edit Project Data** on the upper toolbar. In the new window, click on the **Edit** toolbar button, then **Loudspeakers, File** and finally **Send table to file**.

A popup window will open, asking to use only selected rows. If you want to export all the speakers, click on **NO**. Select the target catalog for the text file you are going to generate, and click on **save**. The newly created file can now be imported into SPAT Revolution.

i Note

Chances are the reference point used in EASE won't be the central reference point typically used in SPAT. Simply use the Transform speaker arrangement option. Typical, you will want to offset X / Y position or Rotate the arrangement 180 degrees.

Nexo NS-1

In NS-1, to export all the speakers: Go to the **Speaker Positions Windows, Speakers/Speakers Position** or **Ctrl + P**. Select all the Speakers, and click on **Export, File...** This will export a .txt file, readable by *SPAT Revolution*.

Adamson BluePrint

In BluePrint, to export all the cabinets: Go to the **Cabinet** tab. In the line **Cabinet Group**, click on the 4th button, **Export All Cabinets**. This will export a .arys file readable by *SPAT Revolution*.

d&b ArrayCal

In ArrayCal, export all the sources with “Files/Export/EASE/Coordinates of all sources”. This will export a .xld file readable by *SPAT Revolution*.



Figure 18.4: image(1)

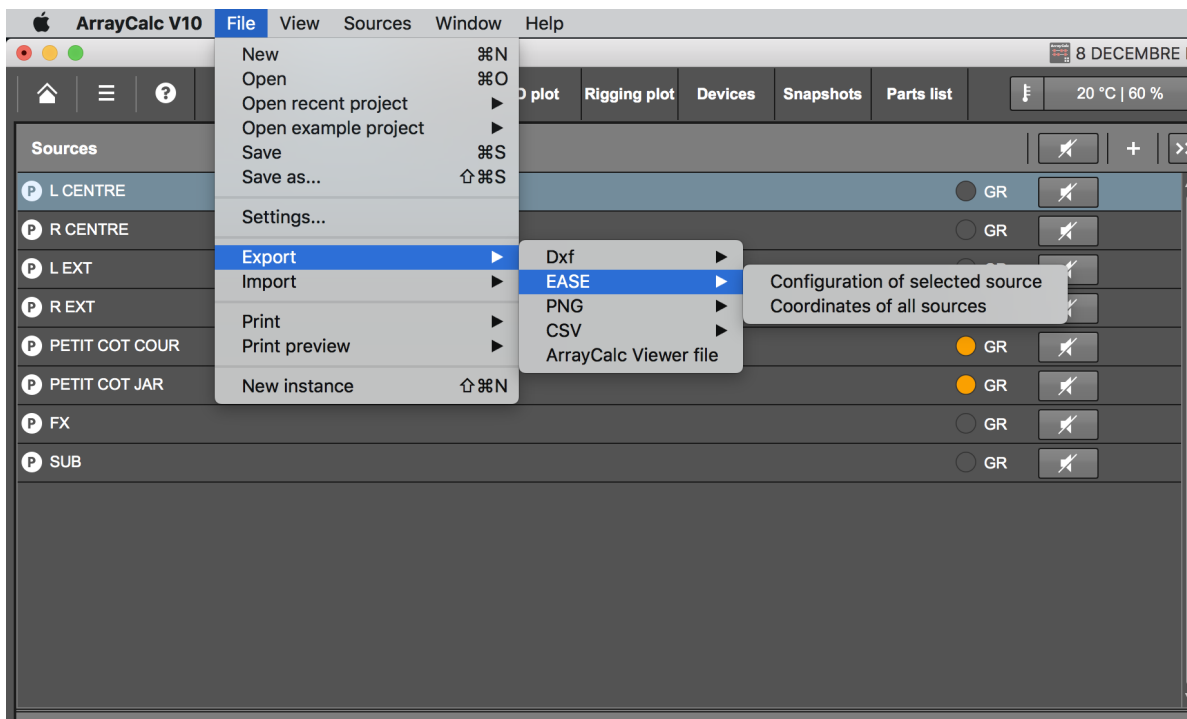


Figure 18.5: image(2)

Excel

A	B	C	D
SPAT FLUX IMPORT	X [m]	Y [m]	Z [m]
Main Left	-4.00	2.00	5.00
Main Right	4.00	2.00	5.00
L&R Surround 1	-6.00	-2.00	2.00
L&R Surround 2	-6.00	-4.00	2.50
L&R Surround 3	-6.00	-6.00	3.00
L&R Surround 4	-6.00	-8.00	3.50
L&R Surround 5	-6.00	-10.00	4.00
L&R Surround 6	-6.00	-12.00	4.50
L&R Surround 7	6.00	-2.00	2.00
L&R Surround 8	6.00	-4.00	2.50
L&R Surround 9	6.00	-6.00	3.00
L&R Surround 10	6.00	-8.00	3.50
L&R Surround 11	6.00	-10.00	4.00
L&R Surround 12	6.00	-12.00	4.50

Figure 18.6: image(2)

Speaker import from Excel template

19 Transformations

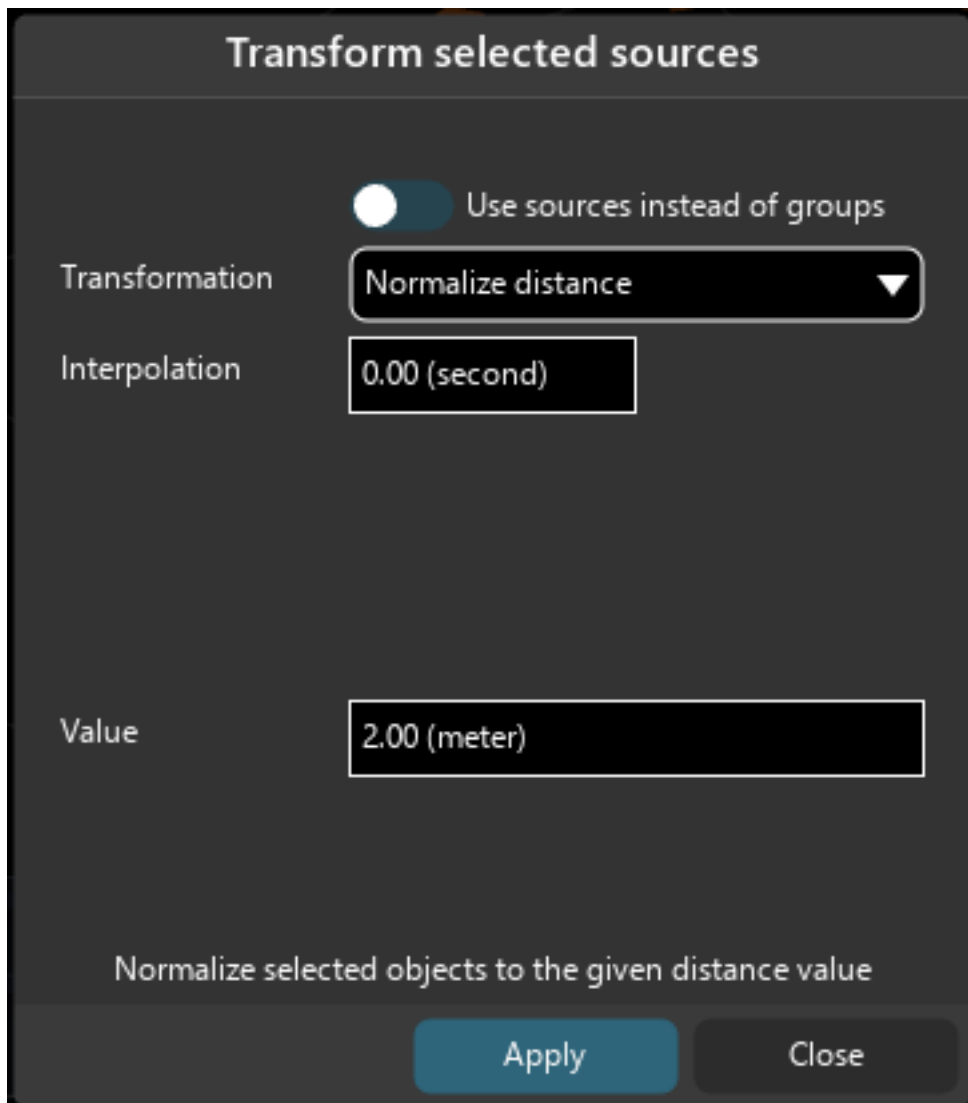
It is possible to edit objects all at once using geometric transformation. In *SPAT Revolution*, these transformations are mathematical laws used to distribute points (in our case, speakers) in the space, and can be used in either speakers or sources. In other words, it can help to create circles, spheres, or lines of speakers in seconds instead of minutes.

19.1 Sources Transformations

As with the custom speaker arrangement editor, we can apply some transformations to one or multiple sources. This feature is especially handy if you wish to quickly set sources on a circle or to put a selection of sources at a given distance for example.

To open the transform menu, right-click on a source in the source panel and choose “Transform.” You can also use the shortcut `CMD/CTRL + SHIFT + T`.

Sources’ transformations also include an integration time which allows to create a smooth transition between the current and the new source position. It also allows to choose to take into account groups, or sources contained on them.



19.2 Speaker Transformation

To modify a speaker arrangement with a predefined action, you can use the “transform” menu. To access it, click on the **Transform** button. A pop-up window will appear.

Normalize Distance: The further speaker will be placed at the given distance. All the other speakers will be placed at a relative distance from it.

Set Elevation: The first ring of speakers will be placed at the given elevation.

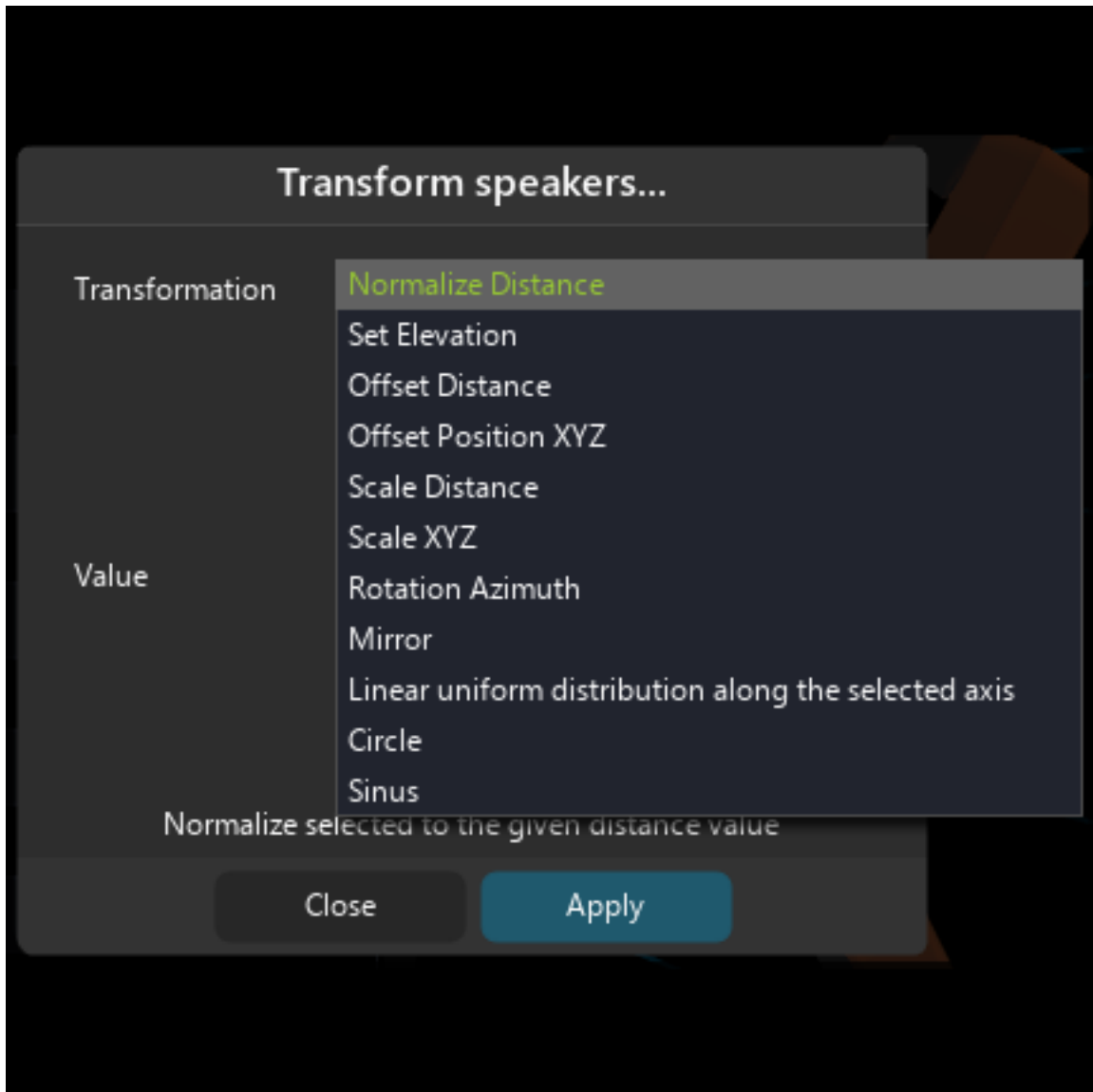


Figure 19.1: *SPAT Revolution* Transform Speaker

Offset Distance: This transformation applies an offset to the distance parameter of all speakers. It preserves the relative position of speakers.

Offset Position XYZ: Same as above but with XYZ parameter.

Scale Distance: This transformation allows scaling all the speakers inside a certain range of distances. It preserves the relative position of speakers.

Scale XYZ: Same as above but with XYZ parameters.

Rotation Azimuth: With this transform, we can apply a rotation to our speaker array.

Mirror: As its name implies, this transform creates a mirror of the speaker arrangement regarding a certain axis.

Linear uniform distribution along the selected axis: This transformation puts all the speakers on the same axis, with even space between them.

Circle: This transformation places all the speakers in the same circle, with an even space between them.

Sinus: This transformation creates a sinus shape with the speakers.

20 Signals visualization and ballistic

Even if ears are the most valuable tool of a sound engineer, having a good and reliable set of tools for metering and visual monitoring is crucial, especially when handling as many channels as we are in immersive sound.

In *SPAT Revolution*, we have three main features regarding this subject: + Vu-meters for each setup bloc. + Nebula for spatial audio-visual feedback. + Option to send audio to the **FLUX:: Analyzer** at many points of the signal path.

20.1 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 Ambisonics 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 setup 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

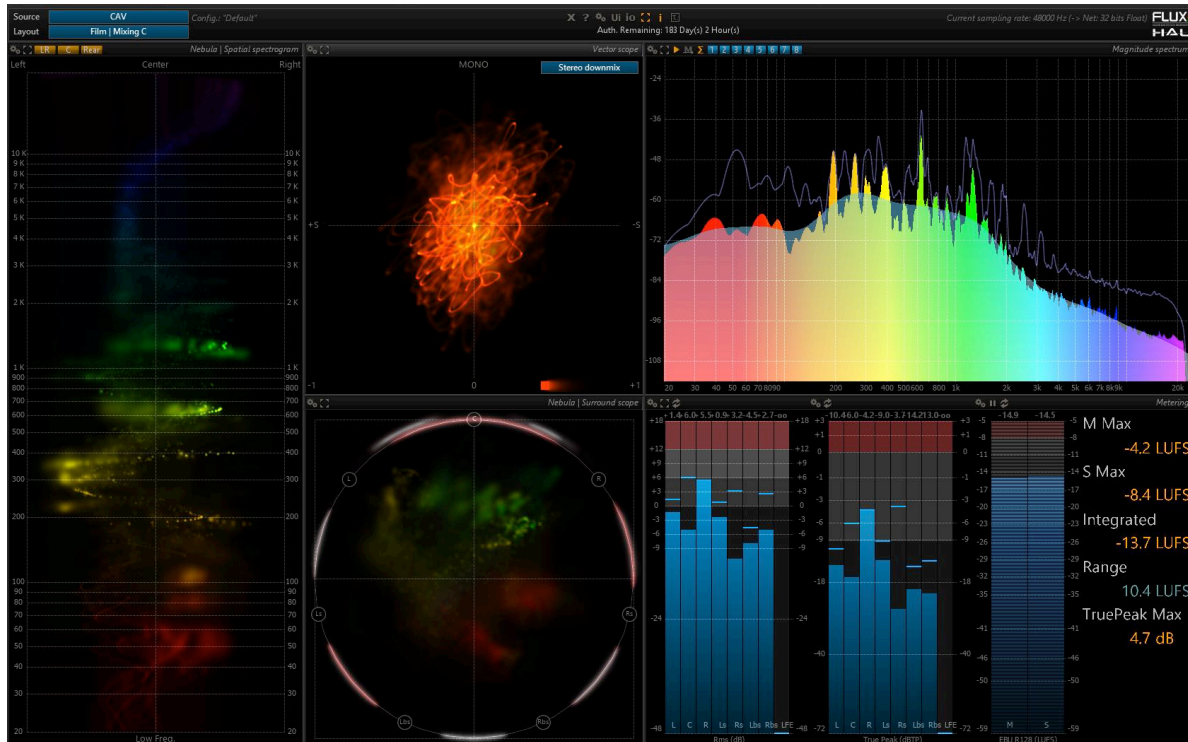
Stream Type Channel Based ▼

Speaker Arrangement Half Sphere | 24. Edit

OUTPUT LEVEL

20.2 Nebula Spatial Spectrogram

Nebula is a technology adapted from our flagship **FLUX:: Analyzer System**, a suite of highly regarded professional mastering and mixing visualization tools.



i Note

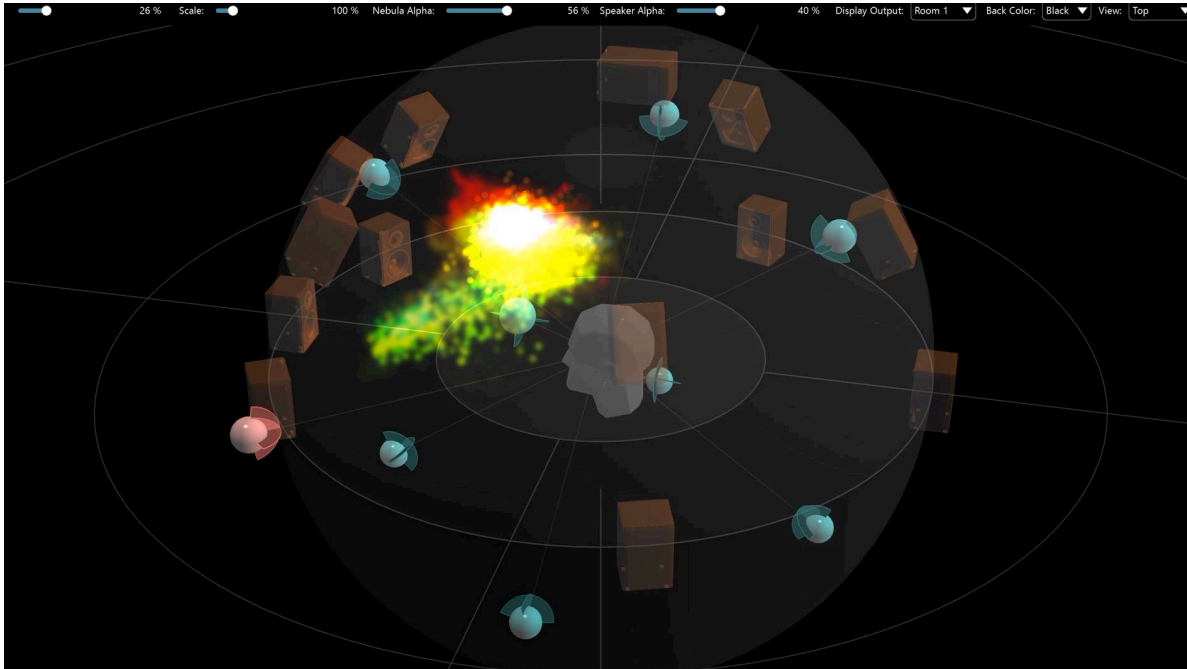
Screenshot from a Flux FLUX:: Analyzer session.

Nebula in *SPAT Revolution* provides a unique representation of the sound field in terms of spectral content and localization rendered directly inside the 3D speaker simulation and virtual room display. It combines the functionality of a spectrum analyzer and a vector scope in a novel real-time display. It is a useful tool to get a real-time overview of your SPATial mix in terms of spectral-spatial diffusion, and can give quite accurate representations of ‘where’ and ‘how’ sound will manifest over a real-world sound system. A lot of work has gone into optimizing the real-time rendering of the display, not solely for aesthetic 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 multichannel 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 color-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.



i Note

Nebula is available for Channel-Based rooms only.

20.3 FLUX:: Analyzer Workflow

SPAT Revolution provides a close integration with the analyzer designed by FLUX::, [the FLUX:: Analyzer](#), allowing to send audio directly to the FLUX:: Analyzer.

Nothing more simple to do: on the *Setup Page*, select a block. On the list **Send to Analyzer**, select **True**. On the FLUX:: Analyzer, you will see it as a new Source, called “SPAT Revolution”. Select it, and select the correct speaker arrangement into the Analyzer: that it.

i Note

Send to Analyzer feature is not compatible between all versions of *SPAT Revolution* and *FLUX:: Analyzer*. Please note that:

- *SPAT Revolution* 22.02.50151 version and below versions are compatible with *FLUX:: Analyzer* 22.01.50131 version and below.
- *SPAT Revolution* 22.09.50200 version and up are compatible with *FLUX:: Analyzer* 23.09.50XXX (coming version) and up.

21 Application preferences

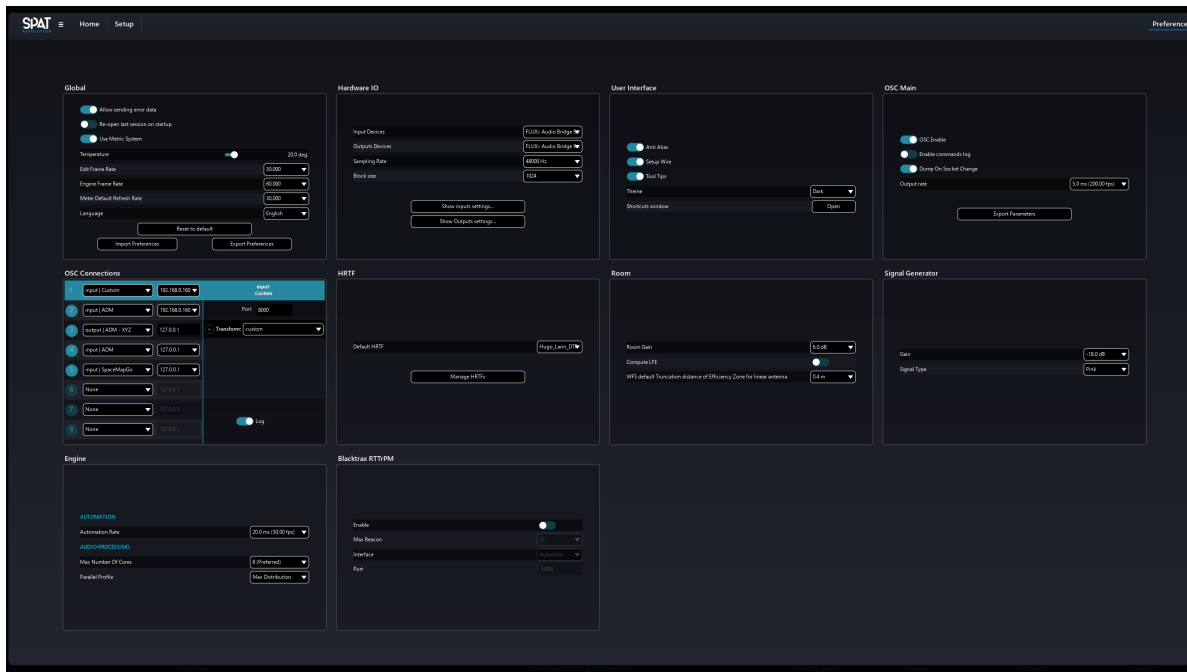


Figure 21.1: Preference Page

This page allows you to change the global preferences of the application.

21.1 Global

21.1.1 Allow sending error data

Allow sending error data to FLUX:: support. All the data are sent anonymously.

Off by default

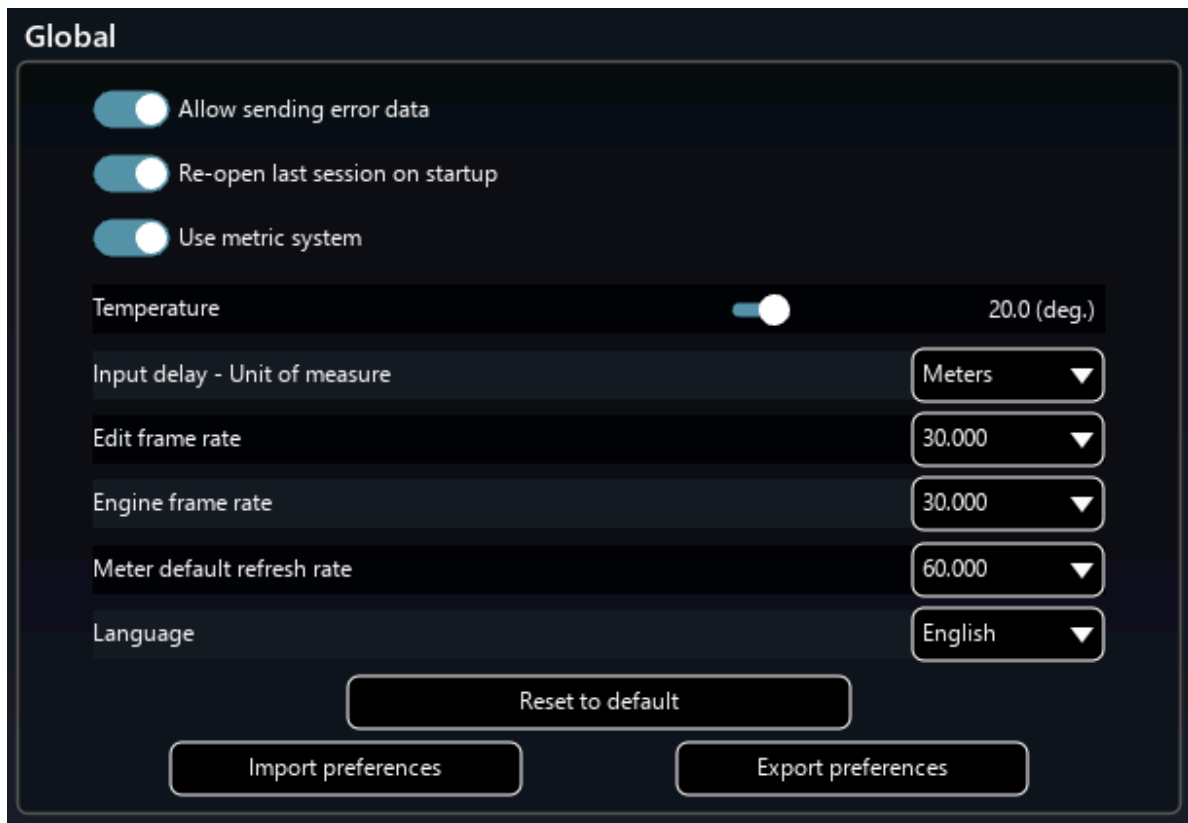


Figure 21.2: Global Section

21.1.2 Reopen last session on startup

The last opened session will be automatically reopened at the start of SPAT Revolution.

On by default

21.1.3 Use Metric System

Toggle between the imperial system and the metric system.

On by default

21.1.4 Temperature

Change the temperature used in the delay line calculation.

Default is 20 °C

21.1.5 Input delay - Unit of measure

Change the unit format for the delay.

Default is sample

21.1.6 Edit Frame Rate

Define how many times per second the UI is refreshed.

Default is 30 Hz

21.1.7 Engine Frame Rate

Define how many times per second the engine is refreshed.

Default is 60 Hz

21.1.8 Meter Default Refresh Rate

Define how many times per second the meters are refreshed

Default is 30 Hz

21.1.9 Language

Allow to change the language of SPAT Revolution. English, Italian, German, Korean, French, and Spanish are available. If you wish to help us to translate SPAT into another language, don't hesitate to contact us.

Default is English

21.1.10 Reset to default

Reset back all the global parameters to default.

21.1.11 Import/Export Preferences

Allow exporting or import preferences.

21.2 Hardware IO

The hardware IO panel allows choosing the audio interface that can be accessed by SPAT Revolution.

21.2.1 Input/Output Devices

This menu allows selecting the audio interface. SPAT Revolution audio engine supports all devices compatible with ASIO (Windows) or CoreAudio (macOS).

Warning

Note that you should select “None” when using the Local Audio Path connections.

SPAT Revolution audio engine also supports different input and output interfaces, on both macOS and Windows, for maximum flexibility.

In the case of different input and output devices, the sample rate and the block size need to be the same.

21.2.2 Sampling Rate

Select the sampling rate used by the audio engine.

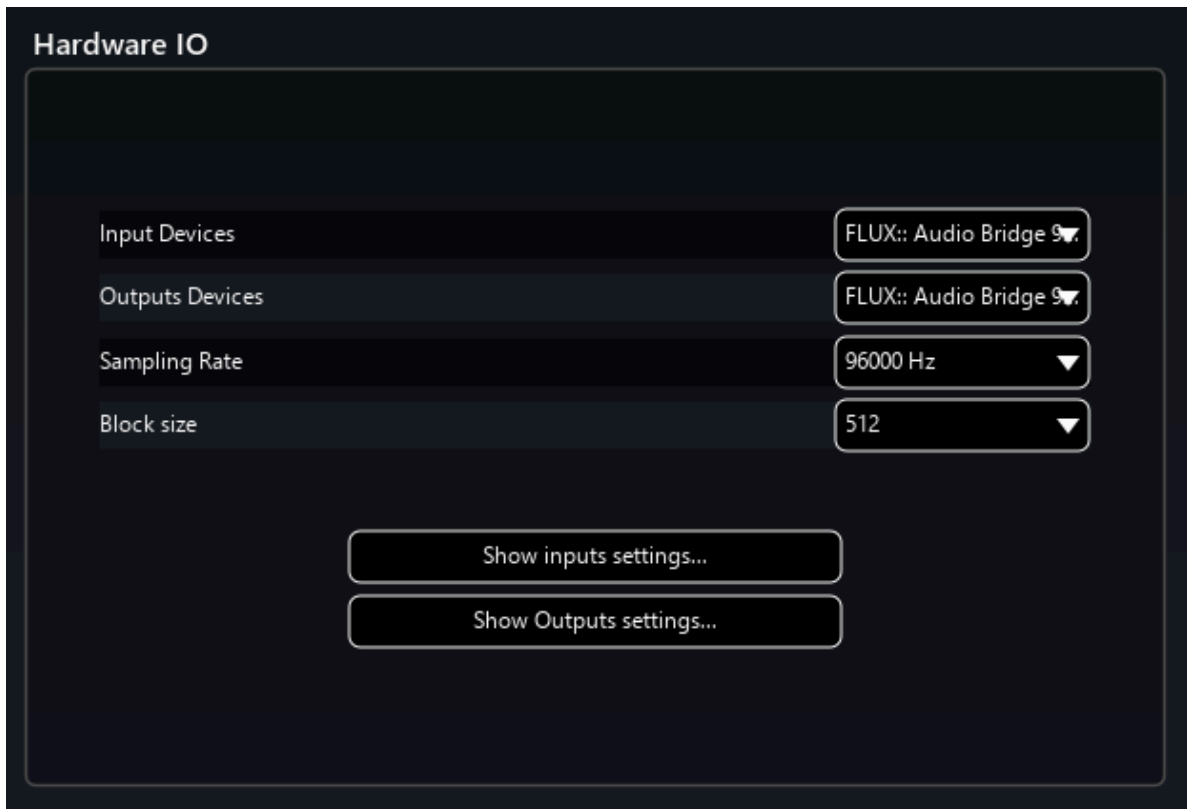


Figure 21.3: Hardware IO Section

21.2.3 Block Size

The block size defines the global latency added by SPAT Revolution. The lower it is set, the lower the latency is, but the greater the CPU consumption is.

21.2.4 Show input/output settings

This button opens the driver audio panel of the selected interface, if it exists.

21.3 User Interface

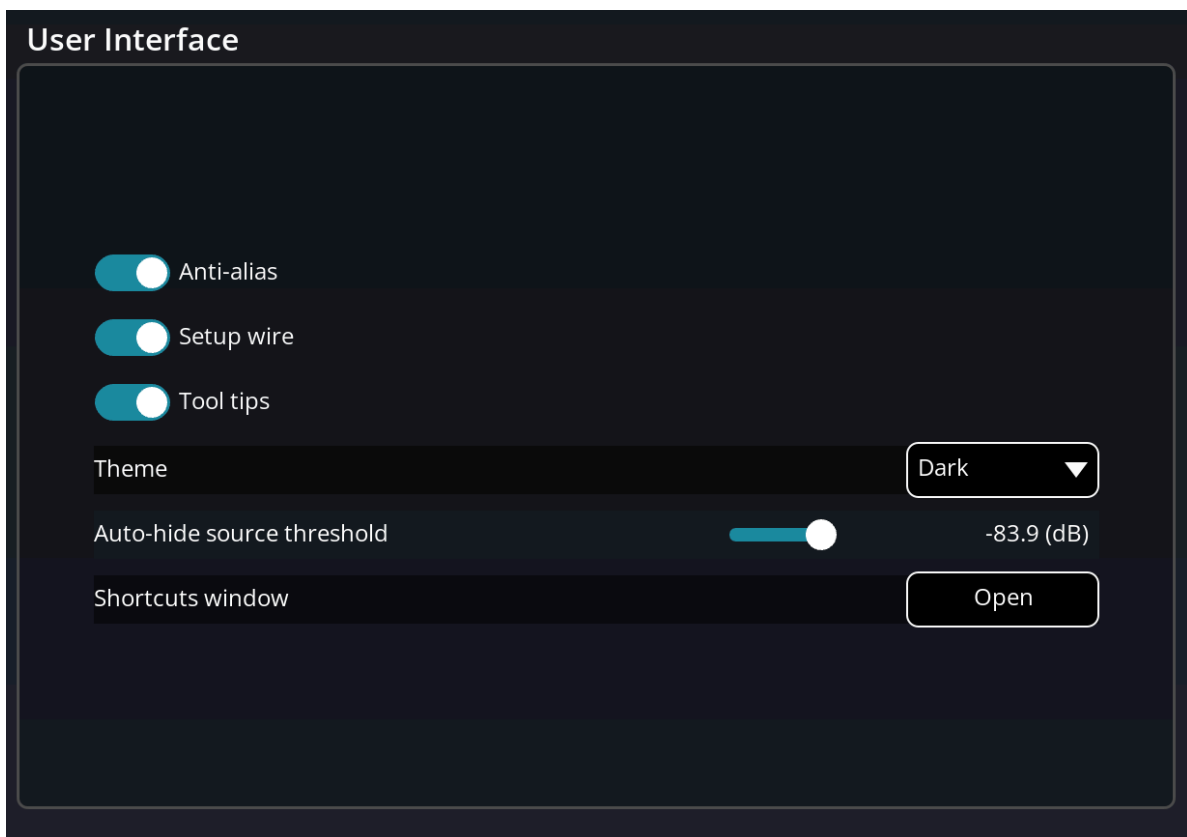


Figure 21.4: User Interface Section

This panel contains some general user preferences.

21.3.1 Anti-Alias

When enabled, this option smooths out lines in the graphic engine. If you are using an Intel-based computer without a dedicated GPU, we highly recommend deactivating this option to improve performance.

On by default

21.3.2 Setup Wire

When on, the wires between blocks in the setup page are splined. When off, they are straight.

On by default

21.3.3 Tooltips

Tooltips contain help and complementary information. When on, they are displayed when the mouse is over a parameter or a control.

21.3.4 Theme

Allows switching between the dark theme and the light theme of SPAT Revolution.

Set on “dark” by default

21.3.5 Shortcuts window

Display the list of all the shortcuts of SPAT Revolution.

21.4 OSC Main

This panel contains basic OSC options.

21.4.1 OSC Enable

Enable or disable the OSC communication between SPAT Revolution and other devices.

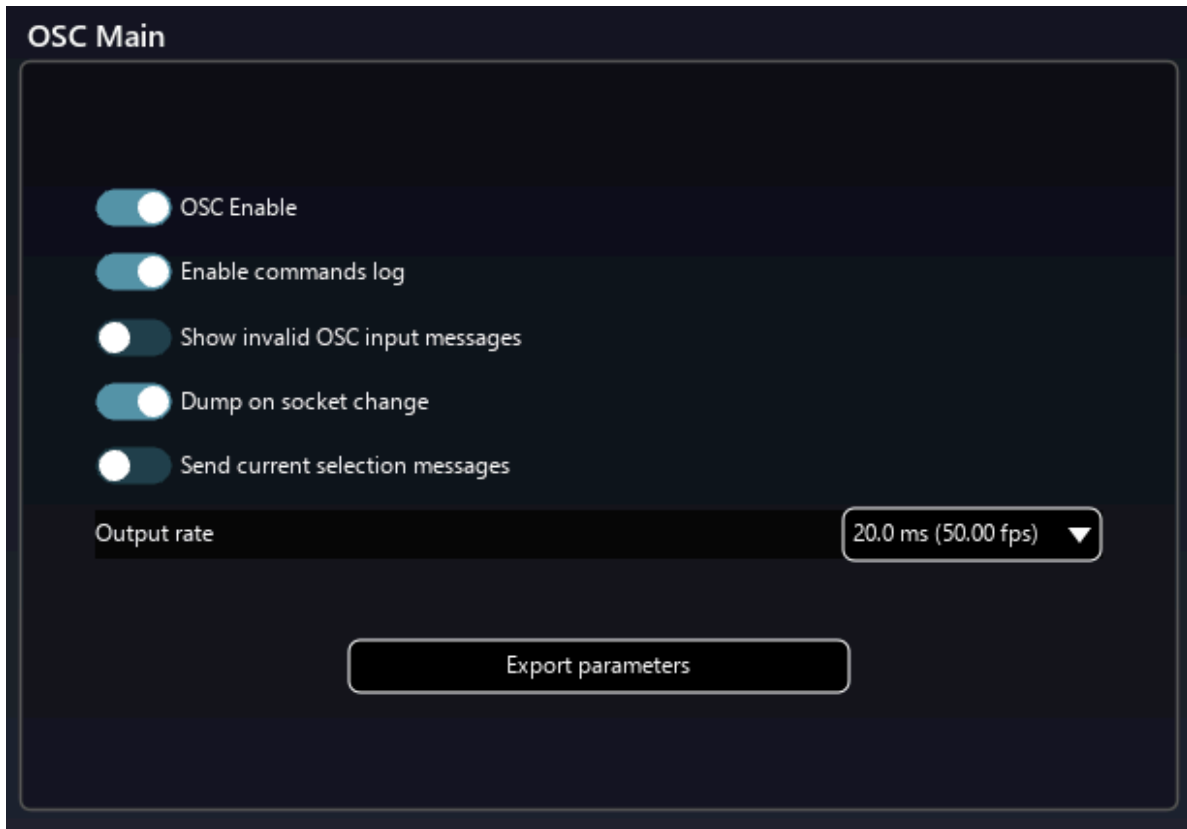


Figure 21.5: OSC Main Section

21.4.2 Enable commands log

Allow displaying the OSC messages in the application's console.

21.4.3 Show invalid OSC input messages

Allow displaying the unknown and invalid OSC messages in the application's terminal.

21.4.4 Dump on socket change

Allow to dump all properties when a socket changes, i.e. when the IP address or the port number has been edited, or the socket is enabled. This ensures a constant synchronization between any remote control and SPAT Revolution.

21.4.5 Send current selection messages

Allow sending a particular message for the source selection. For example, for the gain, the /source/-1/gain will be sent each time the selection gain change.

21.4.6 Output rate

Define OSC output rates in milliseconds.

21.4.7 Export Parameters

This button exports the OSC parameters to a text file.

21.5 OSC Connections

This panel allows the creation of an OSC connection between SPAT Revolution and other devices, like tablets, phones, computers and many others. More information is available on the OSC section [34](#).

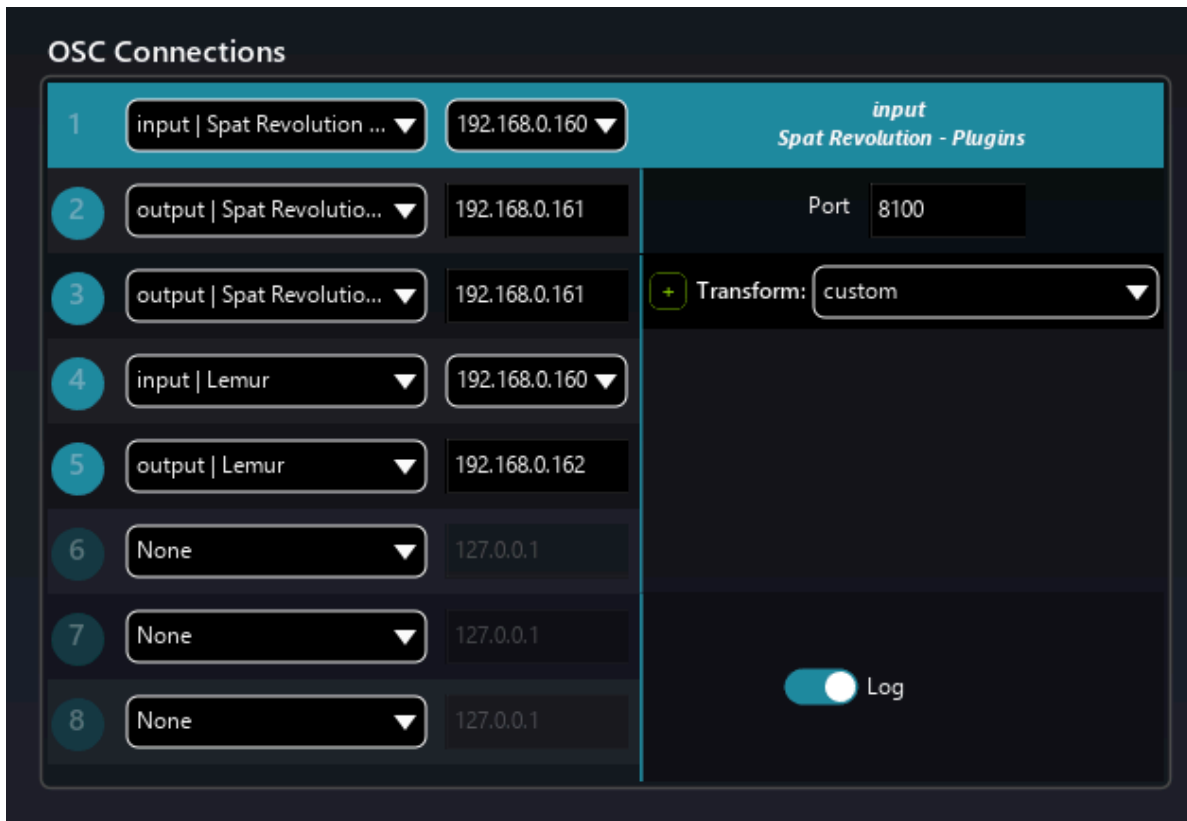


Figure 21.6: OSC Connections Section

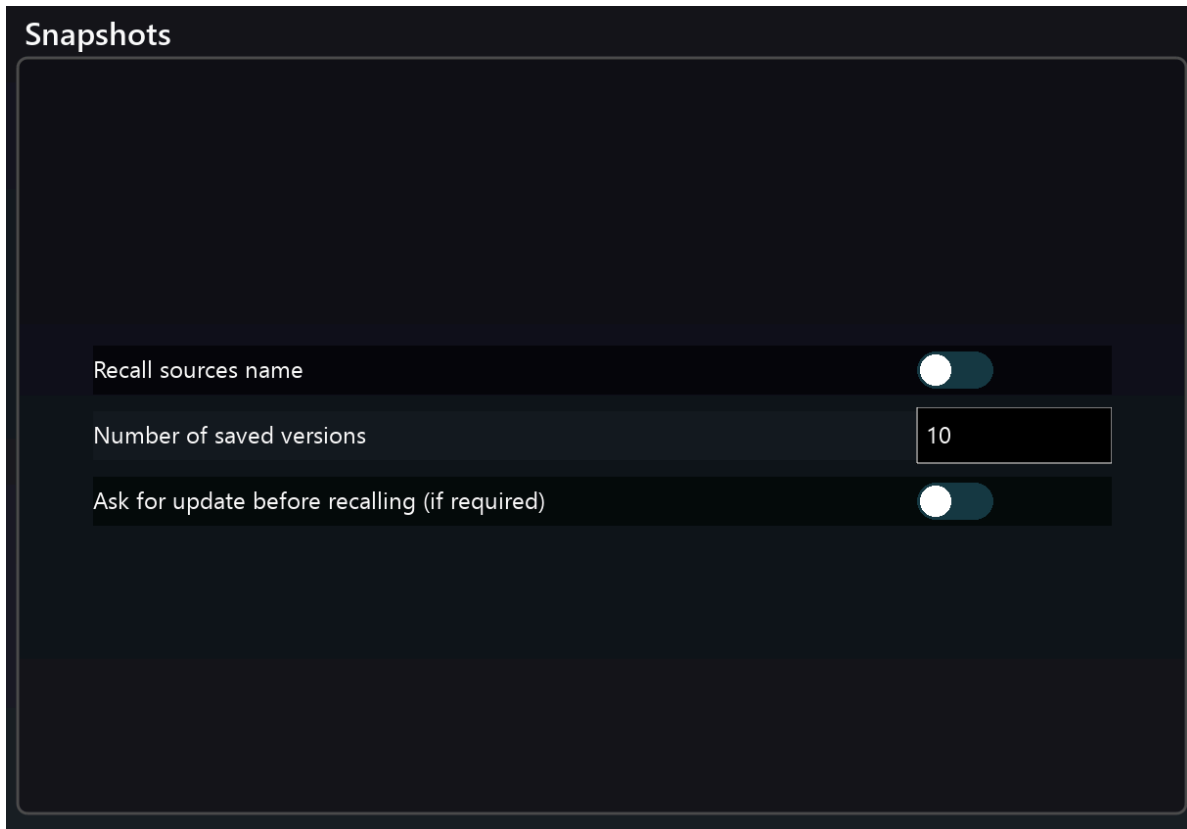


Figure 21.7: Snapshot preferences

21.6 Snapshots

21.6.1 Recall sources name

Determines if the source names are recalled with snapshots.

21.6.2 Number of saved versions

Determines the number of versions saved when updating a snapshot. Careful: increasing this number increases deeply the size of the .json SPAT file.

21.6.3 Ask for update before recalling

When checked, a dialog will ask you if you want to update the snapshot before each recall.

21.7 HRTF

The HRTF panel allows setting up some generic options for HRTF handling.

21.7.1 Default HRTF

Choose the default HRTF used in the binaural room or binaural monitoring blocks

21.7.2 Manage HRTFs - Ultimate license only

The button opens a window that allows downloading or removing HRTF. You can also import your own HRTF from this menu.

21.8 Room

This panel allows changing some room properties.

21.8.1 Room gain

Change the gain of all the rooms of SPAT Revolution.

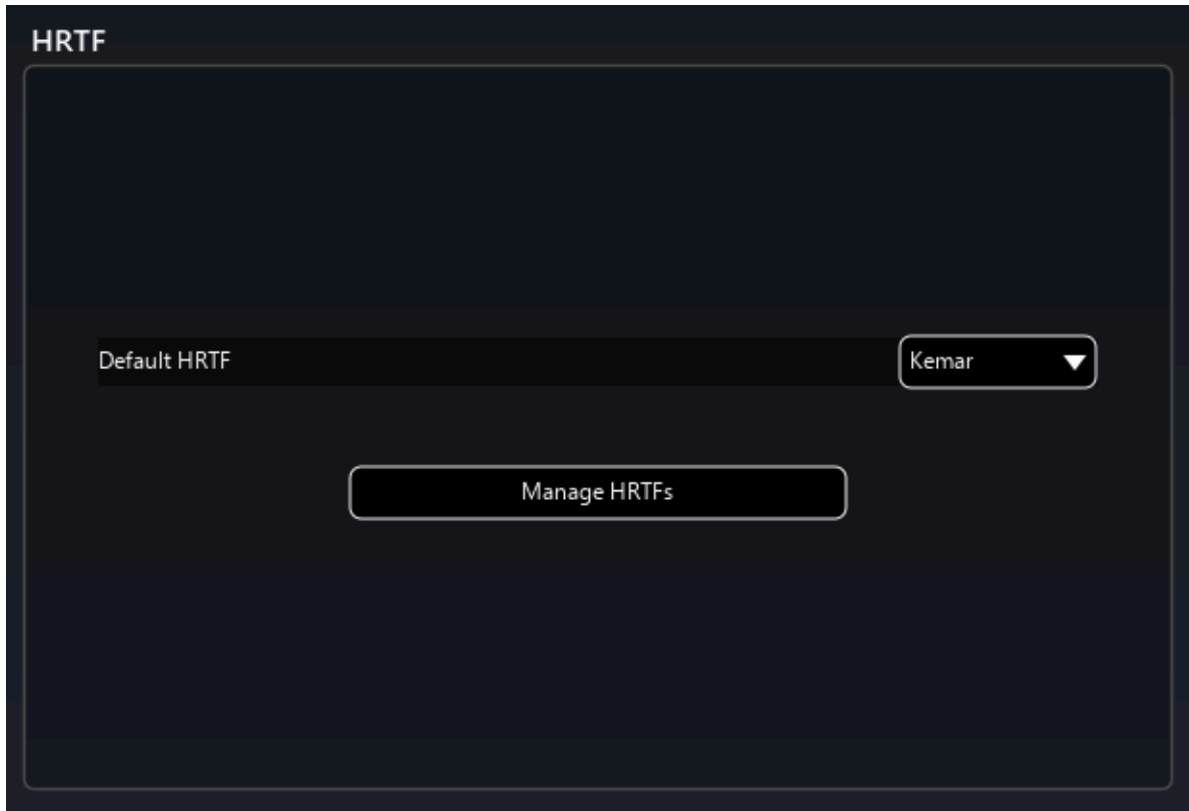


Figure 21.8: HRTF Section

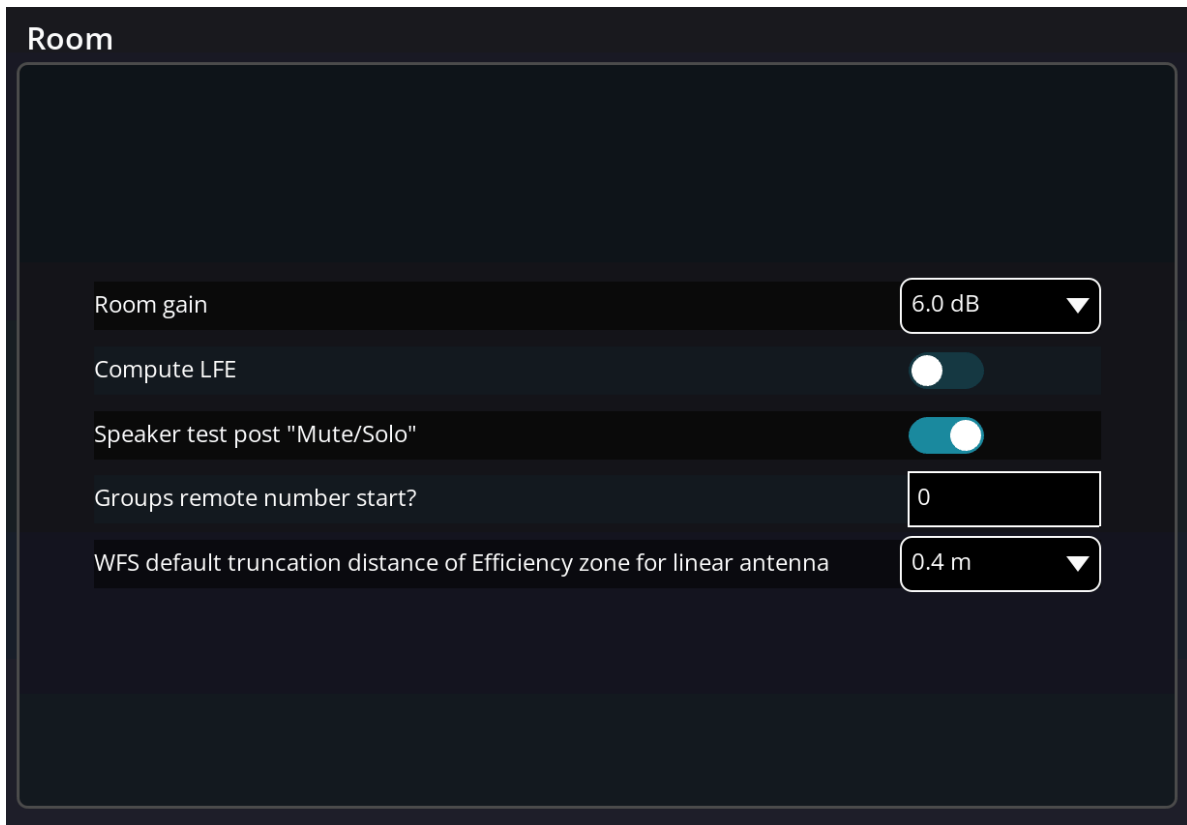


Figure 21.9: Room Section

21.8.2 Compute LFE

Off by default, this option allows including the LFE position in the computing of the speaker alignment process.

21.8.3 Speaker test post “mute/solo”

Change the behavior of the signal generator for speaker, pre or post-speaker’s mute and solo.

21.8.4 WFS default truncation distance of Efficiency zone for linear antenna

Add a distance offset to the efficiency zone when using WFS with a linear antenna to avoid issues with the focus zone.

21.9 Signal Generator

Here, you will find the options related to the test signals generated by the application itself.

21.9.1 Gain

Configure the signal generator gain.

21.9.2 Signal Type

Configure the type of signal generator.

21.9.3 Sinus frequency

Configure the frequency of sinus.

21.9.4 Sweep start frequency

Configure the start frequency of sweep.

21.9.5 Sweep end frequency

Configure the end frequency of sweep.

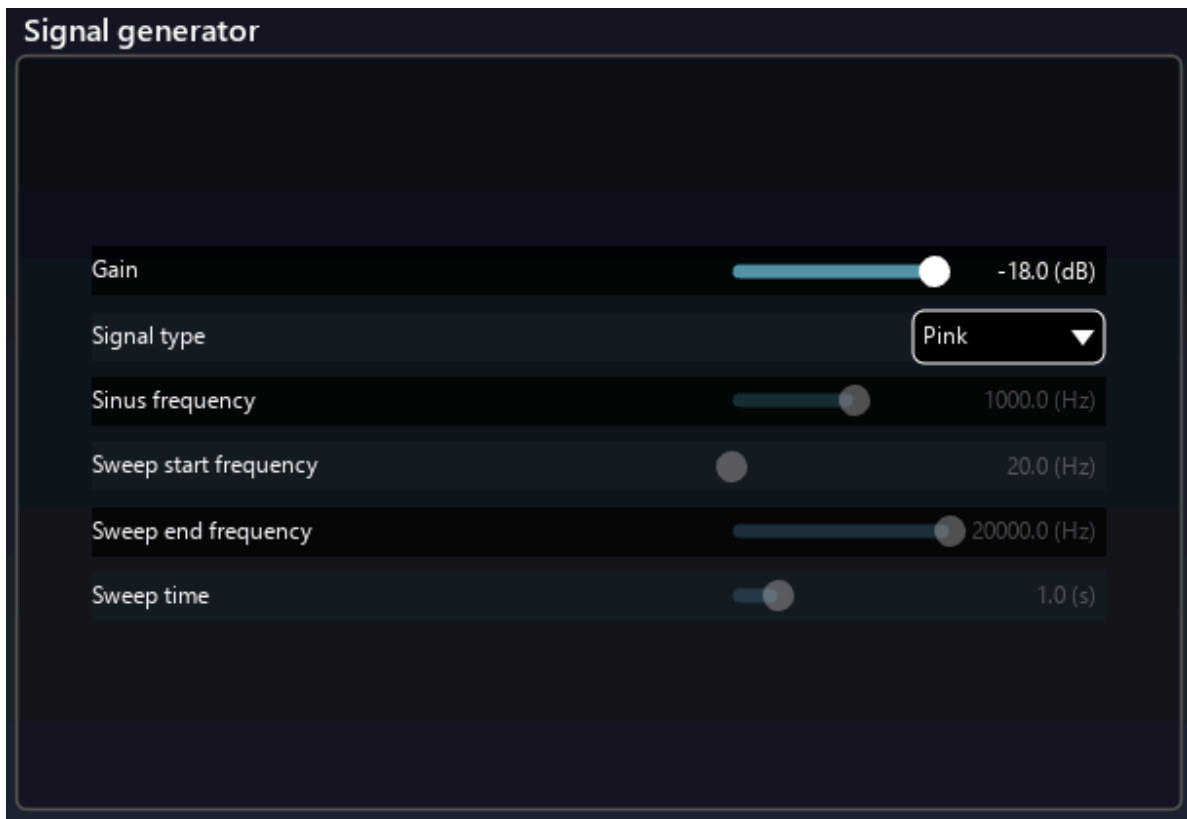


Figure 21.10: Signal Generator Section

21.9.6 Sweep time

Configure the time sweep.

21.10 Engine

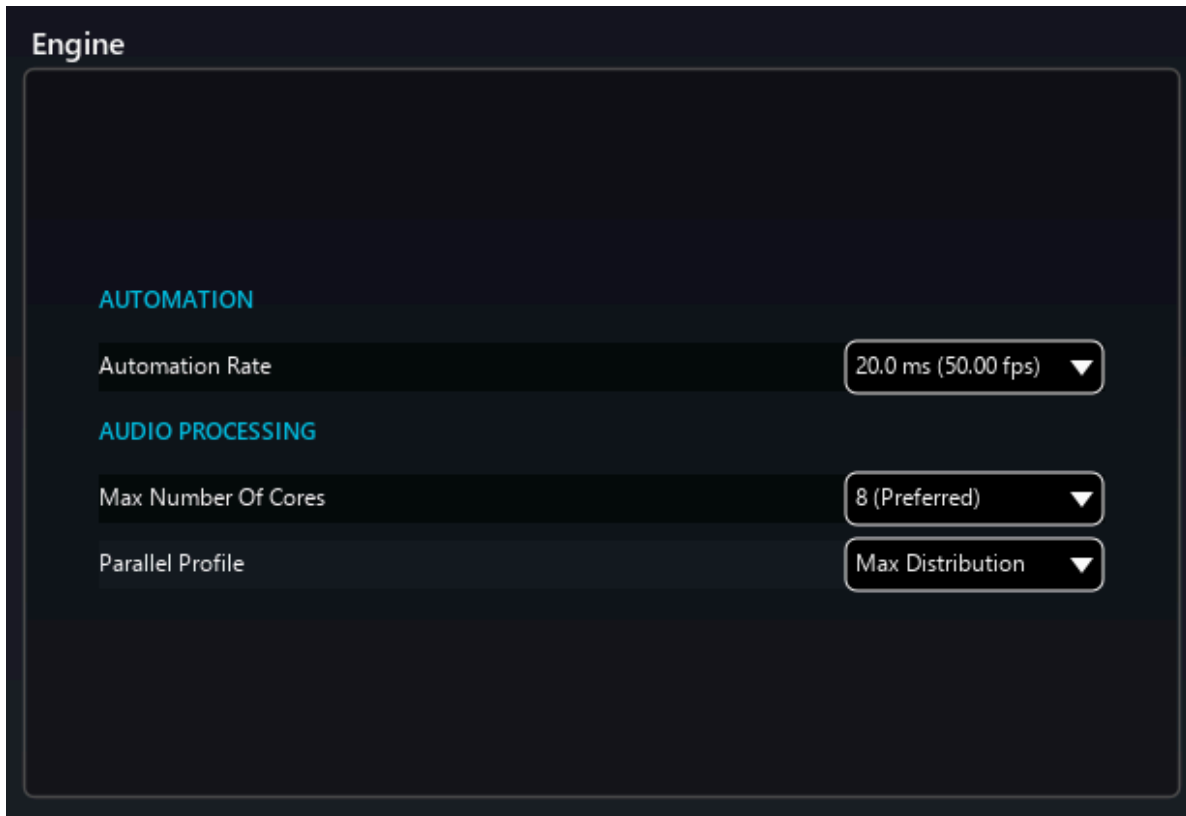


Figure 21.11: Engine Section

This panel allows to modify the behavior of the SPAT Revolution engine.

21.10.1 Automation

Automation Rate

Define the automation rate in milliseconds.

21.10.2 Audio Processing

Max Number Of Cores

Define the number of CPU cores used by the engine.

Parallel Profile

Define how the CPU resources are spread among the cores available. In *Max Distribution* mode, the CPU resources are evenly spread among the available cores. In *Favor First Core*, the engine will fill the CPU cores one by one. The *Balanced Distribution* mode is a compromise between both.

21.10.3 BlackTrax RTTrPM - Ultimate license only

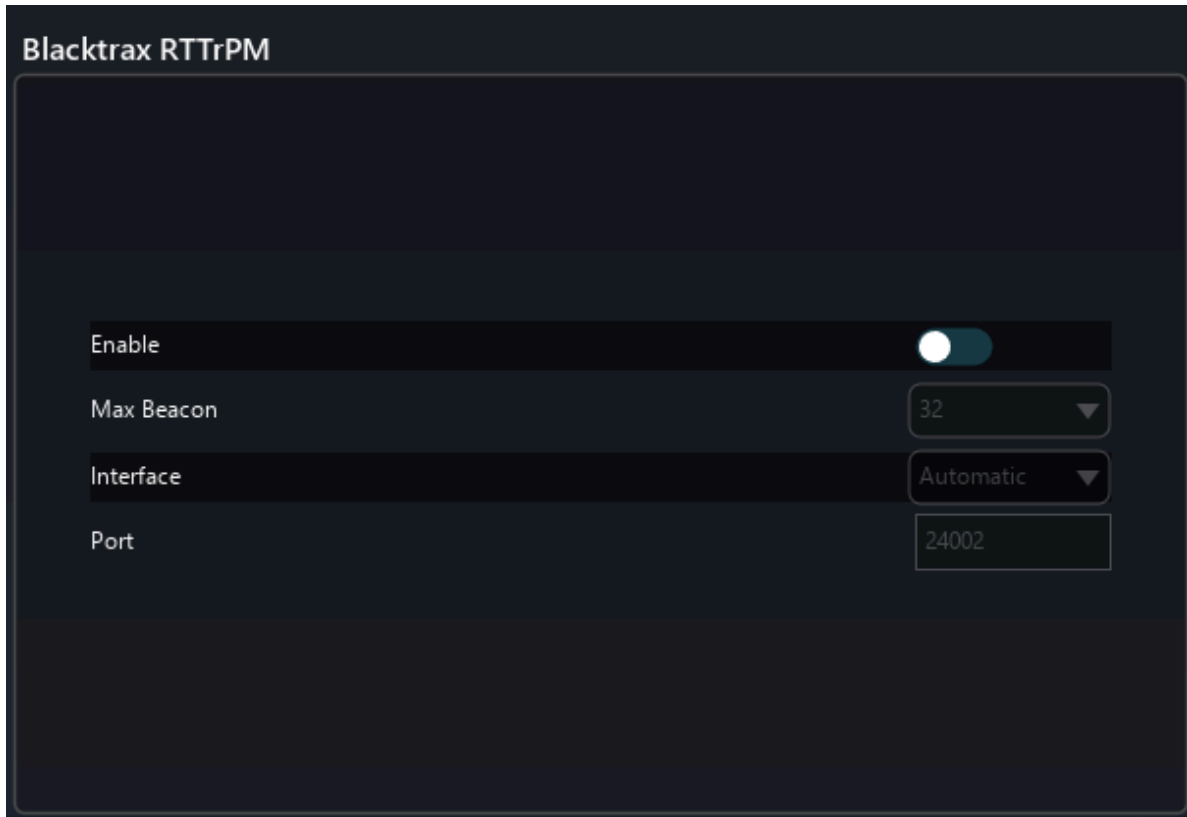


Figure 21.12: BlackTrax Section

See the integration chapter about BlackTrax [52](#)

Part VI

SPAT Setup Modules

We have already covered the **Setup page** in the SPAT Environment section, focusing on its structure and the various ways of interacting with the user interface.

Now we will look more closely at the setup modules themselves. This is an important part of the session design, so be sure to read the following chapters to understand how the signal chain works in SPAT Revolution.



One module, one task

Each module has a specific role. Input and Output modules take care of the routing with the outside world. Sources and Rooms are the heart of SPAT's object-based mixing engine. Basic summing and routing are done with Master and Sum modules. A transcoder can transform audio streams from one type to another. The complete module list is:

- Inputs [22.1](#)
- Input Transcoders [22.2](#)
- Sources [23](#)
- Rooms [23](#)
- Sums [24.1](#)
- Master Transcoders [24.2](#)
- Masters [24.3](#)

- Binaural Monitors [24.4](#)
- Outputs [24.5](#)

Stream types and associated options

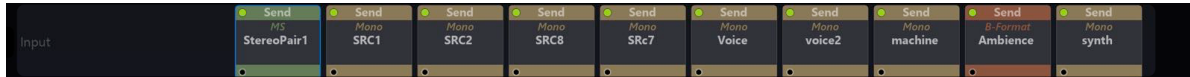
Modules blocs are characterized by their stream type. In spatial audio, audio streams can be of different natures, as seen in the [Spatialization Technology -@spatialization-technology] section.

Stream types and their options are :

- **Channel-based**: most commonly used stream type where one channel corresponds to one speaker or one microphone.
 - Speaker arrangement: allow to select the channel configuration of the bloc.
- **HOA** : generic ambisonic bloc.
 - **Order**: select the ambisonic order from 1 to 7.
 - **Dimension** : select between a 2D or 3D sound scene.
- **A-Format**: pre-encoding ambisonic stream (raw ambisonic microphone output).
- **B-Format** : **deprecated, prefer a 3D HOA first order bloc.**
- **UHJ**: special 3D first order ambisonic stream meant for archiving and stereo compatibility.
- **MS** : Mid-side.
- **Binaural**: headphone-oriented render using HRTF.
- **Transaural**: binaural decoding on speaker.

22 Input Section

In this section, we will discuss how to input audio streams into SPAT Revolution and how to transform them before mixing.



SPAT Revolution itself does not play audio files. The input blocks are here to access to audio streams and process them in the *SPAT Revolution* spatialization engine. Input blocks are represented at the top row of the signal graph.

There are two types of input blocks:

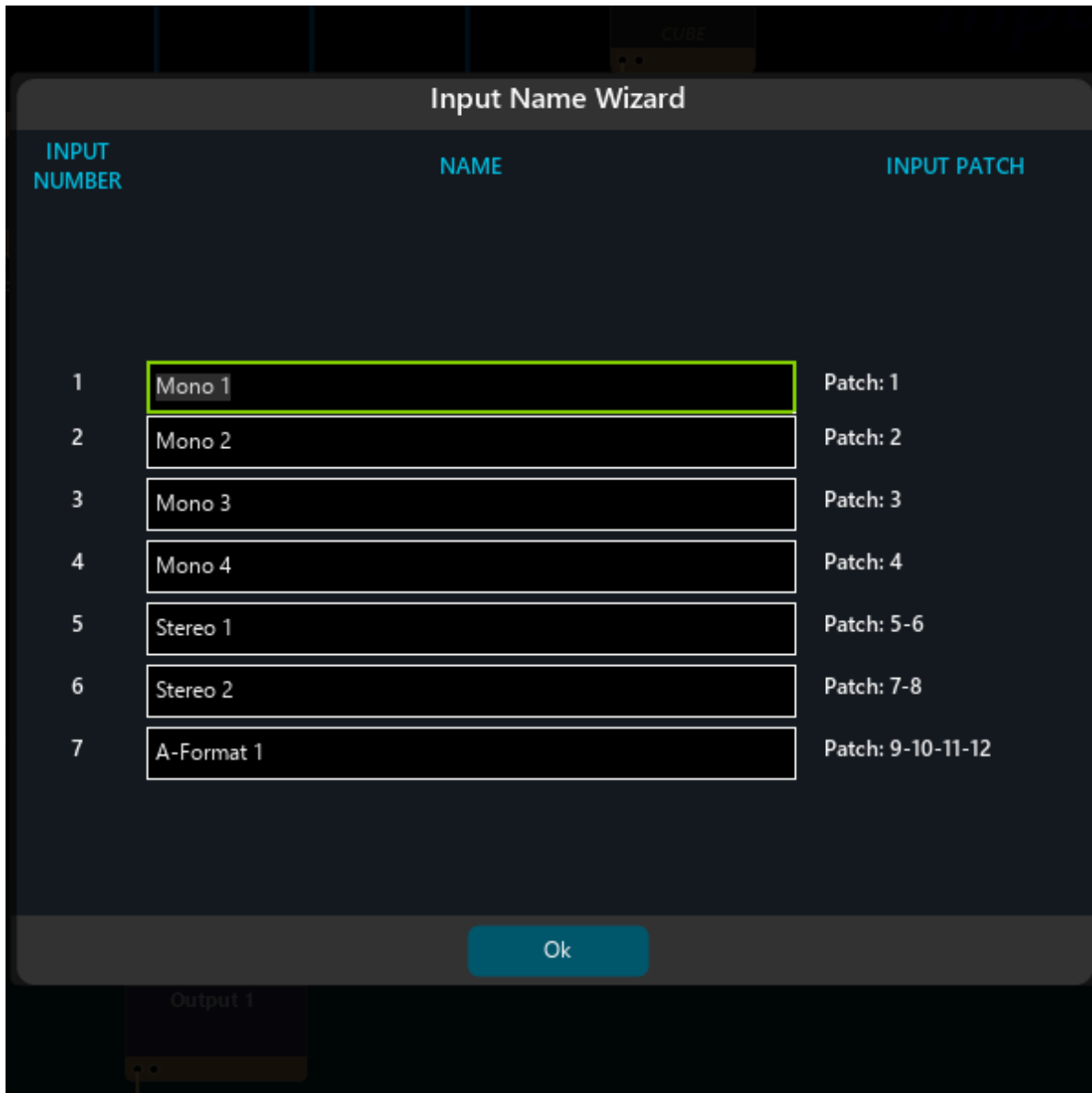
- Hardware blocks, which look at the signal coming from an actual audio interface,
- Software blocks, which are tied to a SPAT Send plug-in instances in a DAW.

To create a hardware block, click one of the + buttons at the left of the row. To create a software block, instantiate a SPAT Send plug-in in a DAW and activate its local audio path. You will find more information on the local audio path in the [8.5 Plug-ins: Local audio path](#) section.

22.1 Input modules

22.1.1 Name

Allows to edit the name of an input. It is possible to edit several names at once by selecting several blocks and clicking on the Input Name Wizard button in the inspector.



22.1.2 Inputs configuration

- Input type: allow to select if the source of the signal is coming from the selected input or the signal generator included in *SPAT Revolution*. The nature of the generated signal can be changed in the Preferences page [21](#).
- Input stream: allow the selection of the stream type of the block.

Note

Input stream types are channel-based, HOA, A-Format, B-Format, UHJ, MS, Binaural, and Transaural.

For more options related to stream type, check the Setup Page [15](#) section.

Tip

When selecting the *MS* stream type, you can tick the **encode** button to convert a stereo stream to MS. This allows you to use an MS-to-Stereo decoder to manipulate the mid and side components of the sound field.

22.1.3 Routing

Give access to the routing matrix

Warning

This option is only available on hardware input.

22.1.4 Setup

Delay

Each input (hardware or software) comes with a delay line which can be useful for:

- live situations, to compensate for delay between microphones.
- studio situation, to compensate for the plug-ins delay when using Local Audio Path. It works in tandem with the **Input delay** option present in the SPAT send plug-in to create the automatic delay compensation mechanism.

The delay can be set in samples, milliseconds, or distance units (meters if the metric system is selected or feet otherwise). This can be chosen on the **Global** panel of the **Preferences** page.

Phase inversion

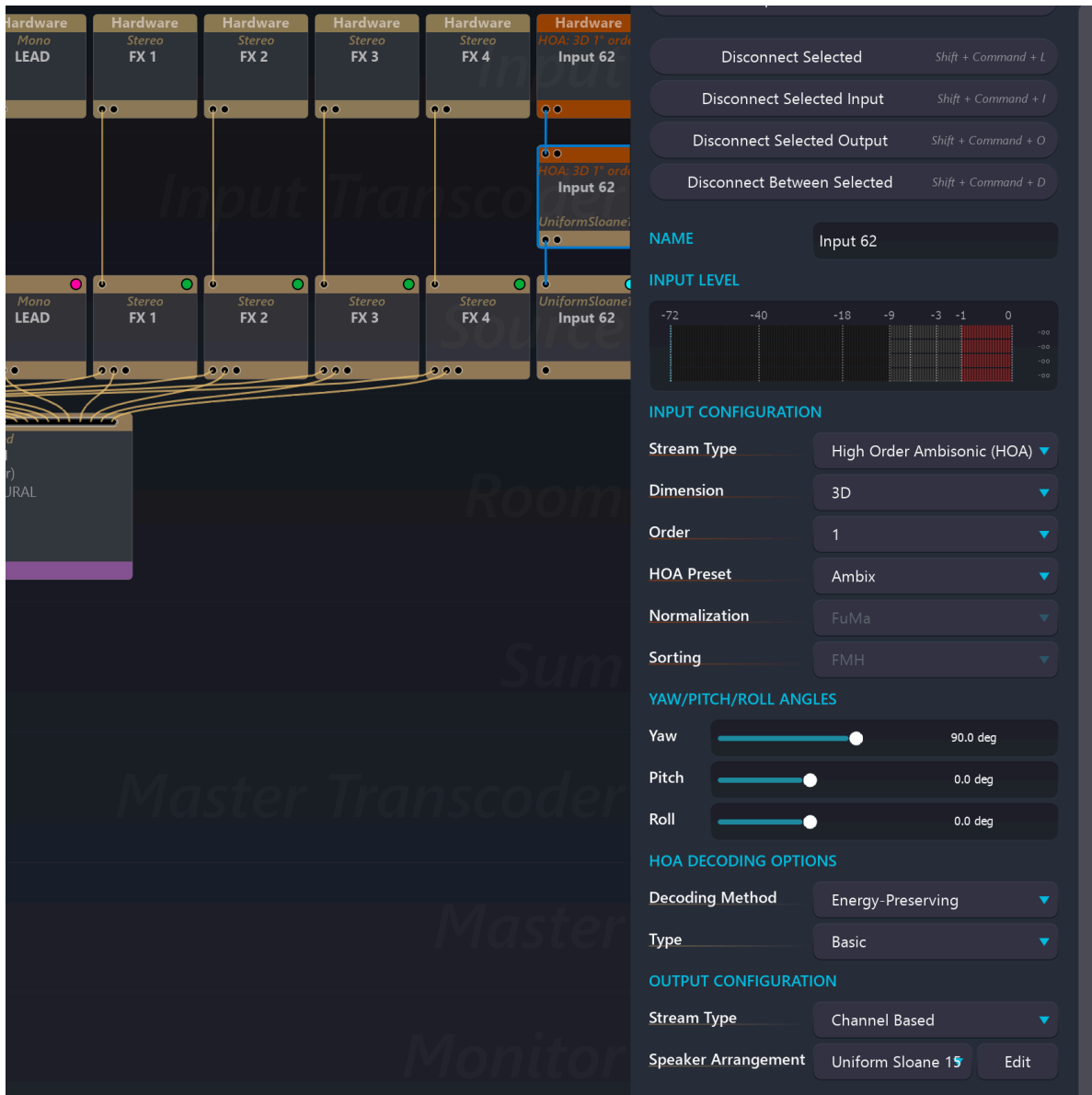
Inverts the polarity of the signal.

Levels

Basic true peak metering for each block's channels.

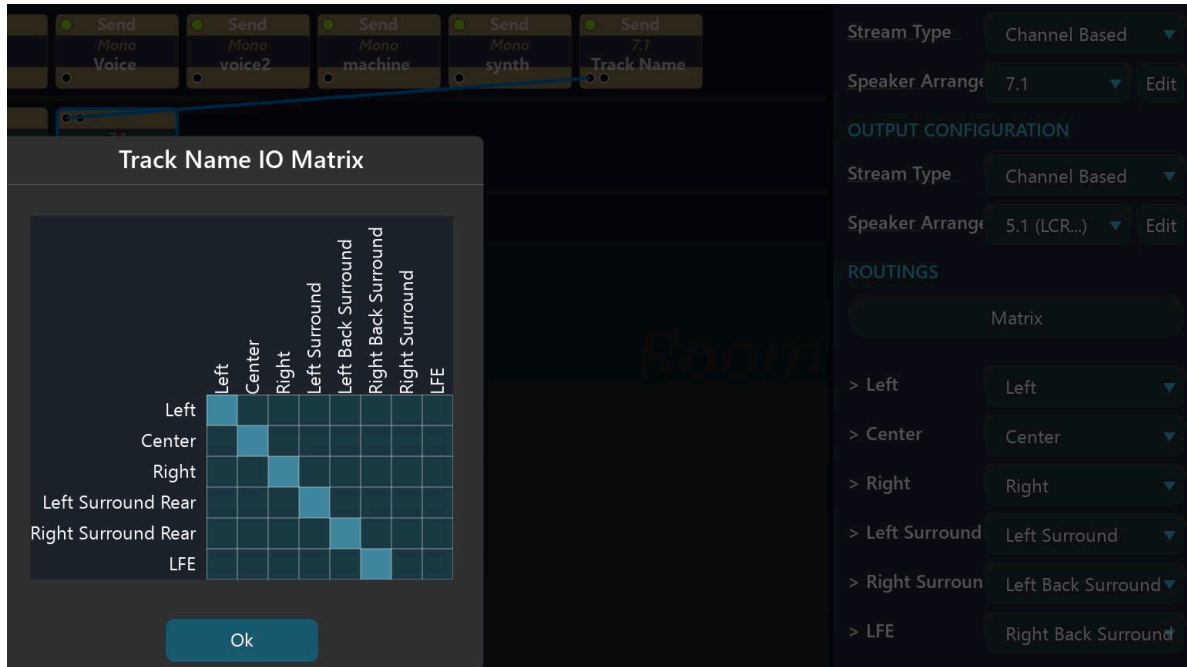
22.2 Input Transcoders

SPAT Revolution can handle many formats of multichannel audio throughout the signal flow, as we have been pointing out. As we approach the actual virtualization 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.



The 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 Ambisonics stream getting transcoded into an eight-channel stream grouped and treated as a specific speaker configuration.

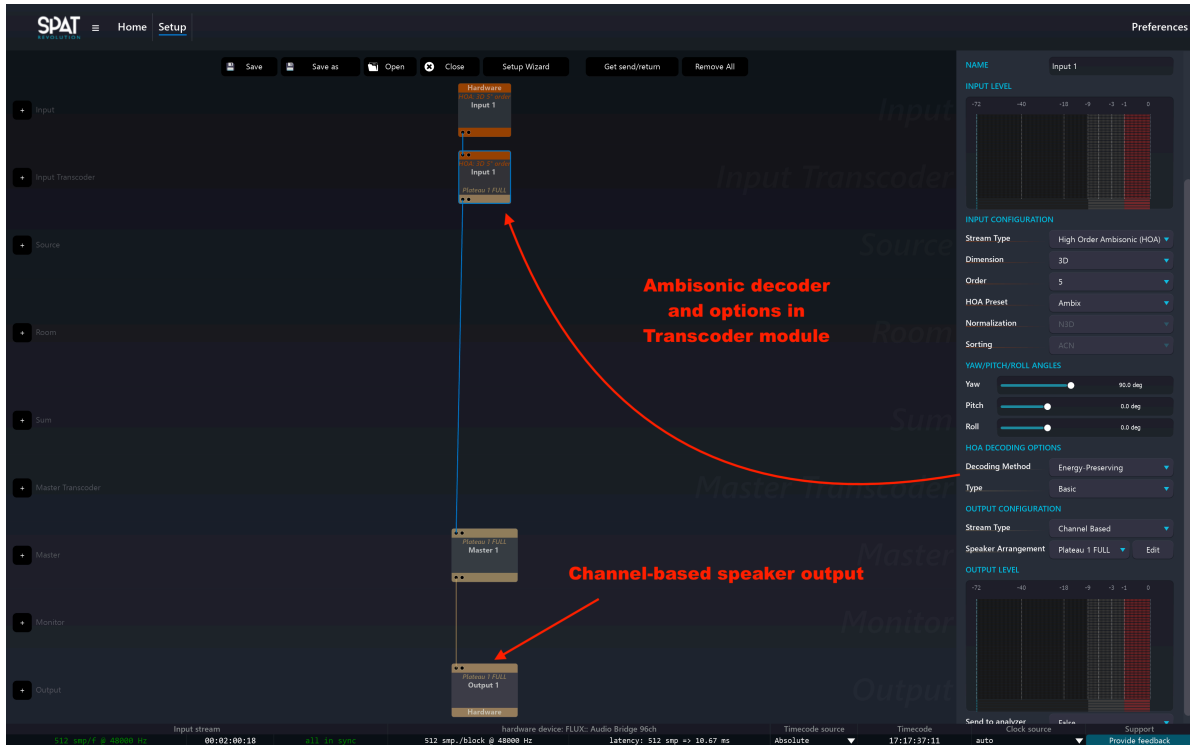
22.2.1 Transcoding Matrix



In the case where an incoming channel-based stream needs transcoding into an outgoing channel-based stream that has fewer channels, the IO Matrix is used to remap the output format by dropping some input channels. This is not strictly *Transcoding* or *decoding* but is a useful tool to have in certain format-changing scenarios. This matrix does not give the possibility to up-mix or down-mix. 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.

22.2.2 When to Transcode Inputs?

The main reason you will need to transcode inputs is when you are mixing and spatializing inputs in a *SPAT Revolution 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, Ambisonics, 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-spatializing in a room. There are some contexts where you will not use a Virtual Room in the signal flow,



Here is an example of decoding an ambisonic signal using an input transcoder module. This could also be done in the master transcoder section of the graph.

As mentioned in Section 10.1, a decoding stage is necessary to render Ambisonics encoded audio to speakers. It can be done like this without the need for a room.

22.2.3 Aggregation of input

As some DAW does not support multichannel tracks, *SPAT Revolution* provides on the input transcoder a way to aggregate stereo or mono input in order to make multichannel sources.

To do it, connect several inputs on the same input transcoder or source and select the wanted format. When selecting it, a combo box **Aggregate Input** will appear, and will allow the aggregation of all the inputs.

Warning

The order of the channel is determined by the connection order to the input transcoder or source.

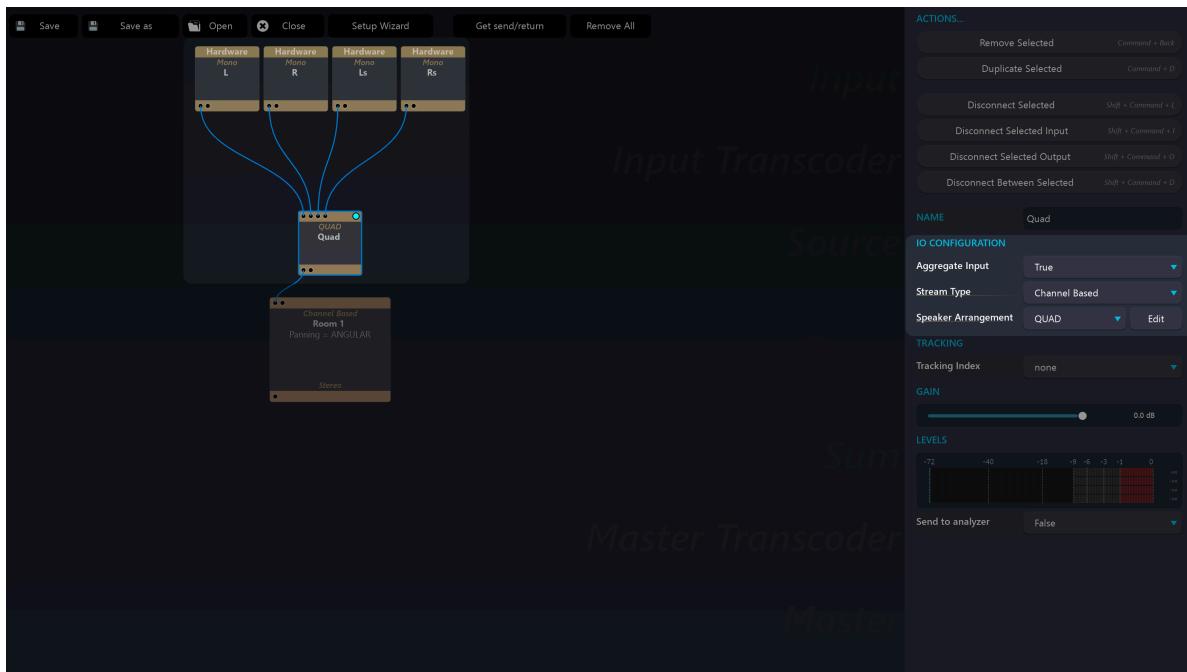


Figure 22.1: Input aggregation

23 Sources & Rooms Module

23.1 Source module

Sources' modules are very important and unique ones. This is where the object-oriented mixing of SPAT Revolution takes in.

Sources' modules associated metadata to incoming audio streams. Because of their object-oriented nature, **they can only input and output channel-based streams**. In practice, this means you cannot patch, for example, an ambisonic stream to a room, even if the room in question is ambisonic itself. It first needs to be transcoded by using an input transcoder.

The so-called metadata is the source parameters, such as position and reverberation information. All these parameters are explained in the section Source properties [27](#).

One source can be connected to several rooms. This is one of the advantages of object-based mixing: it allows the creation of different mixes in different formats with the same exact information.

23.1.1 Name

Allows renaming a source. By default, it takes the name of the input patched into it.

23.1.2 IO Configuration

Allows selecting the speaker arrangement used by the source. It can be selected from the list or user-defined.

Multi-channel sources are one of the unique SPAT Revolution features.

i Note

If you struggle to understand why we talk about speakers at this stage, simply consider them as “virtual” speakers that will be emitting in a “virtual” room.

 Warning

A source can and should only be channel-based!

23.1.3 Tracking

SPAT Revolution can receive data from RTTrPM open protocol tracking systems. This protocol is supported by BlackTrax™. BlackTrax™ is a vision-based system that connects to different third-party applications, such as robotic lights, media servers, and SPAT Revolution. OSC is the other very good method to use for tracking, and various tracking systems support it natively.

 Warning

*RTTrPM protocol is not available with the **Essential** license of SPAT Revolution.*

When you have correctly set up the BlackTrax protocol (see BlackTrax Integration section [52](#)) then you can directly assign Tracking Number to virtual sources, and also to listener position (see Listener position section [4](#)) for advanced virtual reality interactive audio projects.

23.1.4 Gain

Change the input gain of the source. Also includes a phase toggle switch to invert the polarity of the signal.

23.1.5 Levels

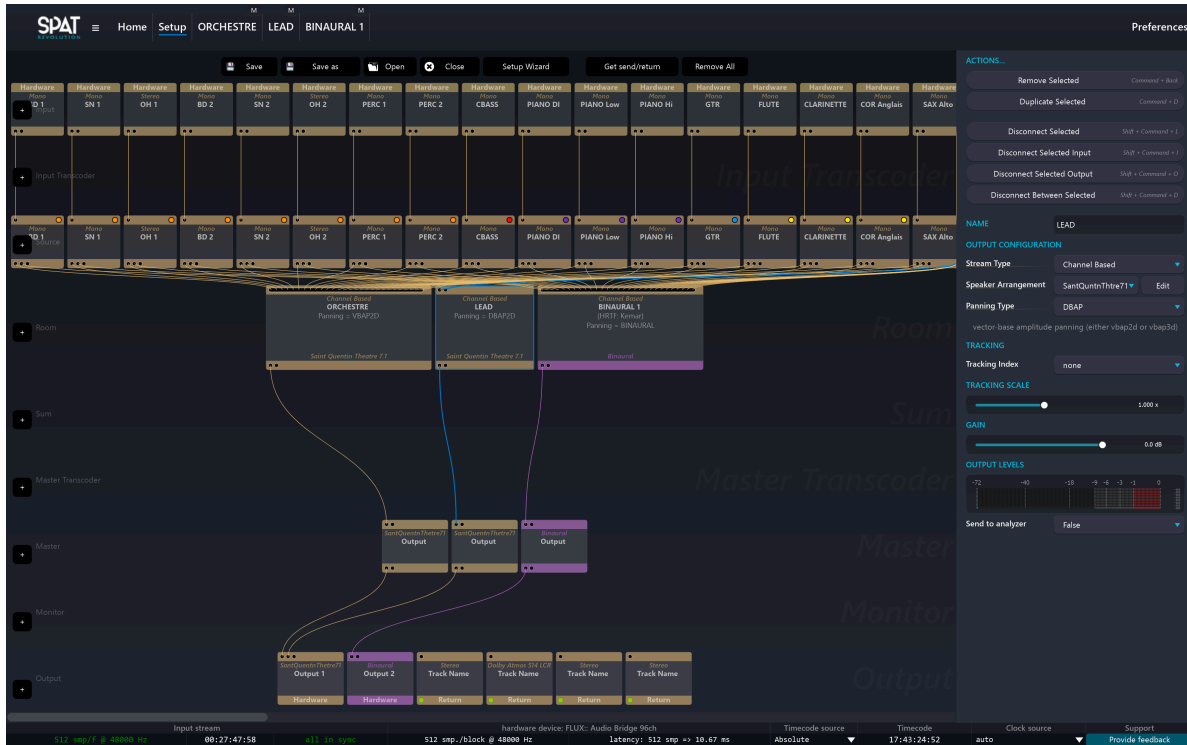
Basic true peak metering for each block's channels.

23.2 Room module

If sources are where localization information is stored, rooms are where the “interpreting” happens. A room is defined by two main criteria: its output stream type and its reverberation. Simply, a room module looks at all the information stored in sources and then acts as a renderer for a given format.

Warning

It is a classic case of object-based mixing, where source modules associate metadata to channels, and room modules interpret them. Only source modules can be connected to room modules.



The first thing to notice is that we can add any number of rooms. SPAT revolution offers flexibility, 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 screenshot above, virtual sources might exist in different spaces that get summed together.

The **Essential** license limits the number of active (processed) rooms to one (1). The others are deactivated (not processed).

23.2.1 Name

Allow renaming a room.

 Warning

Rooms cannot share the same name!

23.2.2 IO Configuration

Allow selecting the rendering type and the associated options.

Stream Type:

- Channel Based
- HOA
- Binaural
- MS
- B-Format (deprecated, please use HOA 1st order room connected to a master transcoder to transform it on B-Format standard)

23.2.3 Tracking

See Listener position section [4](#).

23.2.4 Gain

Change the output level of the room. Also includes a phase toggle switch to invert the polarity of the signal.

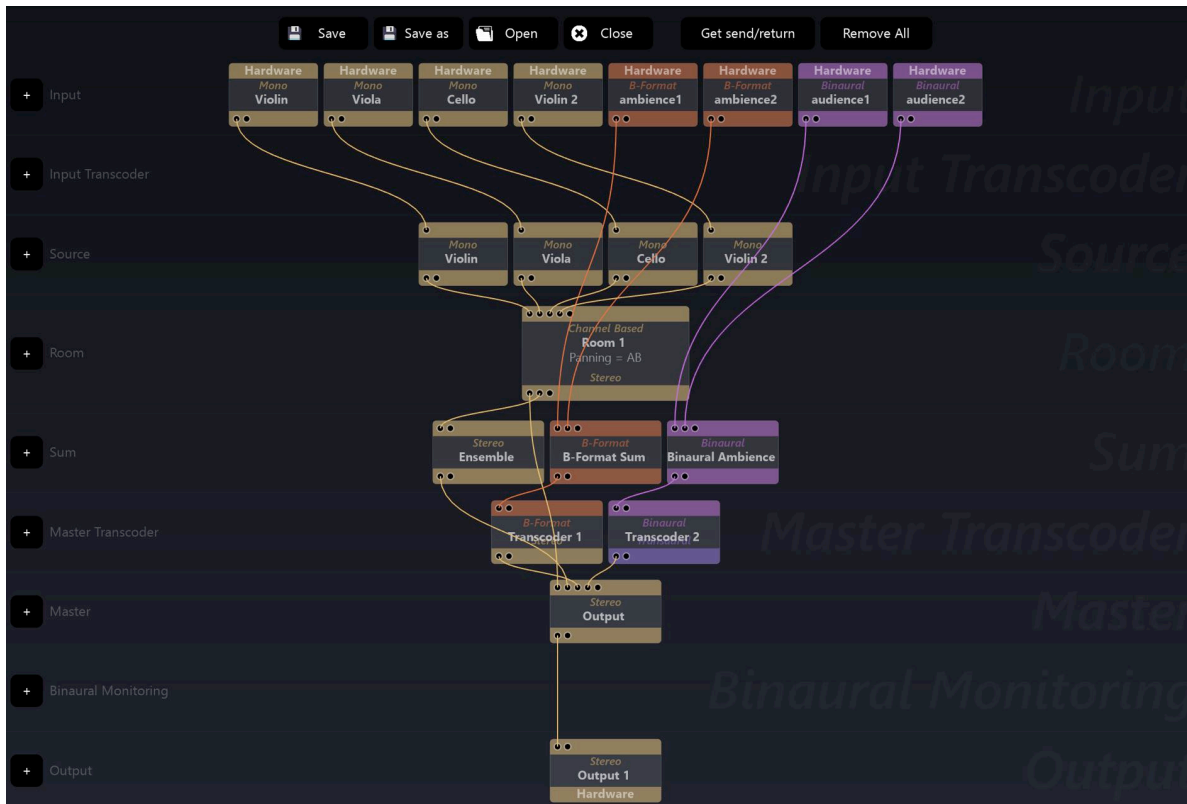
23.2.5 Levels

Basic true peak metering for each block's channels.

24 Master Section

24.1 Sum modules

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 a correct naming convention is important. Summing 5.1 and 7.1 rooms 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.

Note

Summing can also be done directly in **Master** and **Output** Modules, but summing into a **Sum** module will save resources if the same sum is performed in different blocks.

24.2 Master Transcoder 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 [22.2](#).

24.3 Master modules

The **Master** bloc gives you a last gain staging control, as well as down-mixing capabilities. When opening its matrix, it is possible to patch several inputs to one output, or, one input to several outputs. It also includes a phase toggle switch to invert the polarity of the signal.

24.4 Binaural monitoring modules

The **Binaural Monitoring** [5.5](#) row provides a way to decode the whole scene to headphones only in binaural 3D.

Warning

Note that the binaural monitoring block is not designed for binaural production, but for listening to a channel-based mix over headphones. This means that the binaural monitoring modules virtualize loudspeakers and not sources, so the sonic quality of this processing is often less preferable than a proper binaural room.

24.5 Output modules

The main thing to note at the output stage is that you can have any number of output routes. Like the Inputs [22.1](#), they may be either direct hardware routes, which return audio streams (via matrix routing) to the physical outputs of an audio interface connected to your *SPAT*

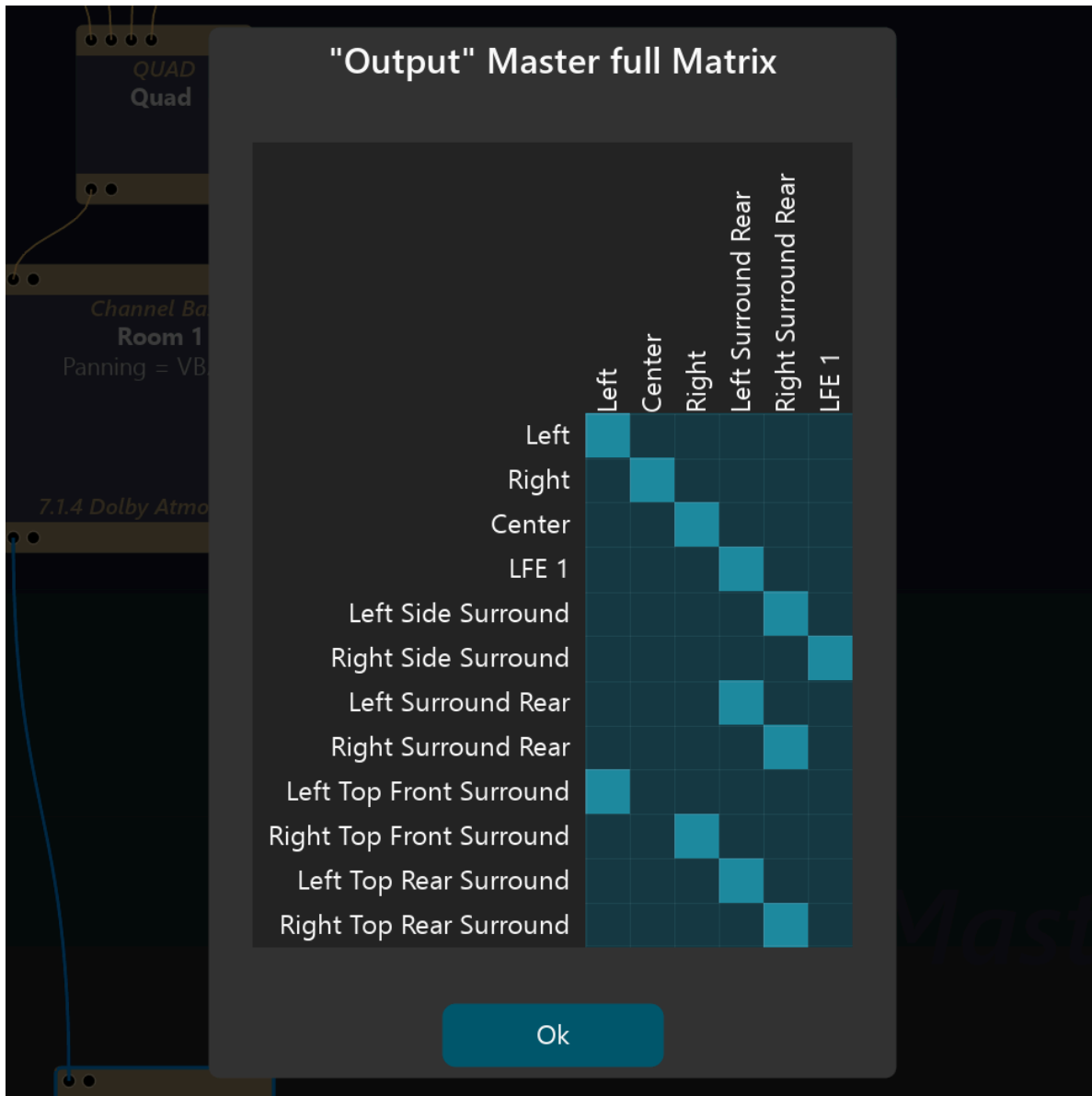


Figure 24.1: Master matrix

Revolution workstation. The Hardware Output workflow is the most direct way to render in real-time to an actual loudspeaker system (or headphones).

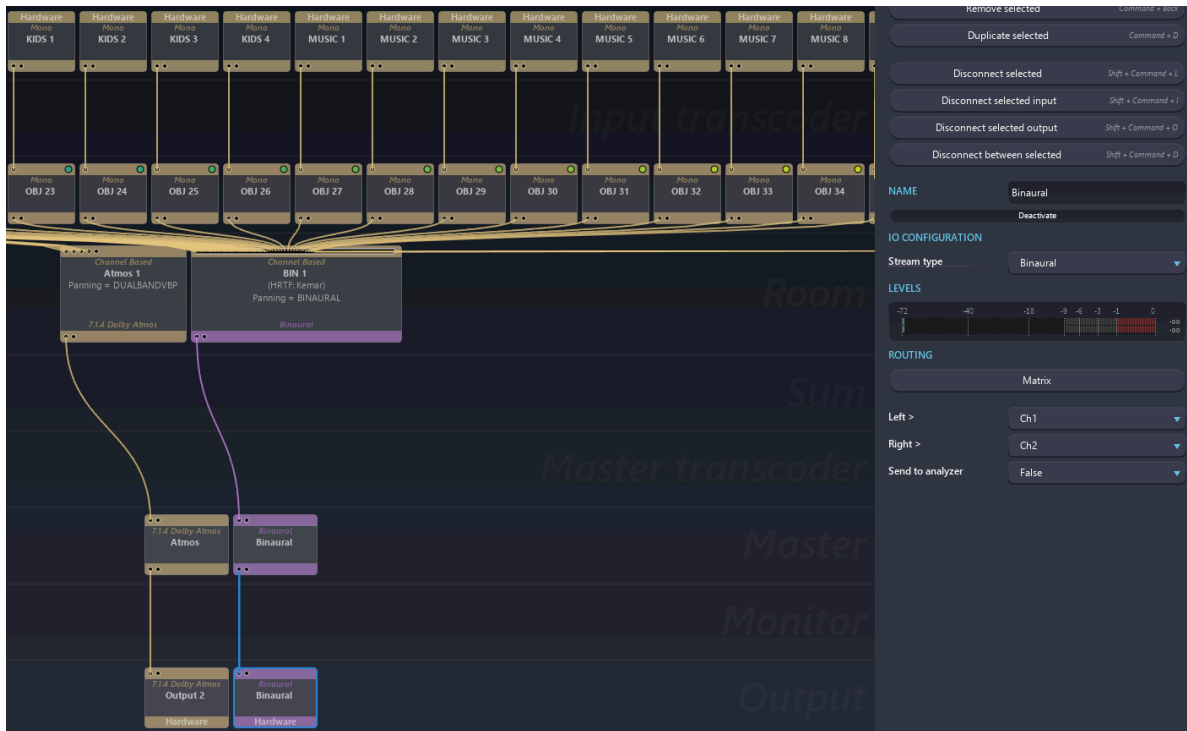
Warning

Software outputs are created by the instantiation of a SPAT Return plug-in in a compatible DAW when its local audio path is set to “on”.

Outputs may also be linked to *SPAT Revolution* RETURN plug-ins, which return 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.

Note

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



25 Modules Channel Count

Into *SPAT Revolution*, modules are treated exactly the same whether they are mono or multichannel, ambisonics, binaural or channel-based.

25.1 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 straightforward 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 localize and balance spatially with others.

i Note

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

25.2 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

25.3 Four Channel Input

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

ACTIONS...

Remove Selected Command + Back

Duplicate Selected Command + D

NAME Input 1

INPUT CONFIGURATION

Input Type Live input ▼

Stream Type Channel Based ▼

Speaker Arrangement QUAD ▼ Edit

ROUTING

Matrix

> Left Surround Rear Ch1 ▼

> Left Ch2 ▼

> Right Ch3 ▼

> Right Surround Rear Ch4 ▼

LEVELS

The level meter displays four channels. The scale ranges from -72 to 0. The right two channels (Right and Right Surround Rear) are highlighted in red, indicating they are active or selected. The left two channels (Left and Left Surround Rear) are in black. The current levels for all channels are at the -00 mark.

Send to analyzer False ▼

A four-channel stream could contain the format of a four-speaker channel-based format (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 [Ambisonics -@scene-based] 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.’

 Warning

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

25.4 Multi-Channel Based Input

Any input module configured to represent a stream of multichannel audio can be configured as a Speaker Arrangement format which would require that number 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 standardized speaker layouts that 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 it’s 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.

 Warning

Essential license of *SPAT Revolution* limits the total number of channels to 32 input channels and 18 output channels.

 Note

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

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

L	C	R
sbL		sbR
surround back Left		surround back Right

i Note

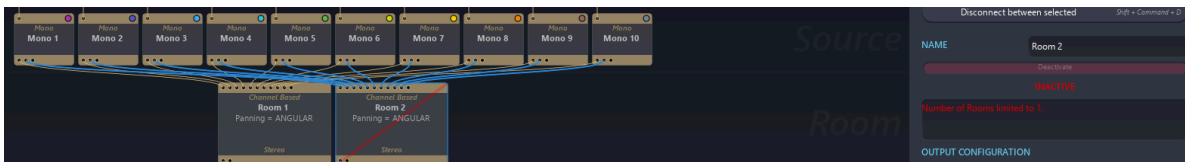
- Some common Speaker Channel naming abbreviations.

26 Session compatibility and modules (de)activation

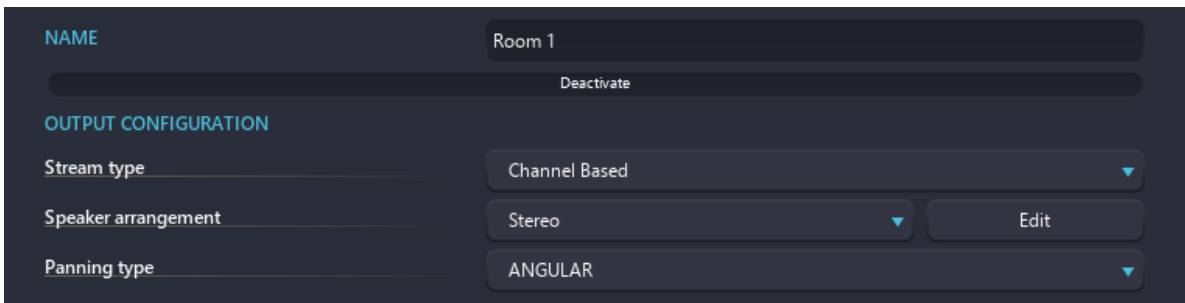
SPAT Revolution Essential version aims at being a limited version of the Ultimate one. Then, it has the same workflow but limited capacities.

26.1 Create and edit a session in the Essential version

Any type of session can be created and edited with the *SPAT Revolution* Essential version. This includes the opening and editing of Ultimate sessions with the Essential version. However, due to license restrictions, the blocks that do not fit the restrictions are automatically deactivated, and thus not processed. If the block deactivation is due to a license restriction, the deactivation reason appears in the block's configuration panel on the **Setup** page:



It is possible to manually activate/deactivate blocks on the block inspector of the **Setup** page:



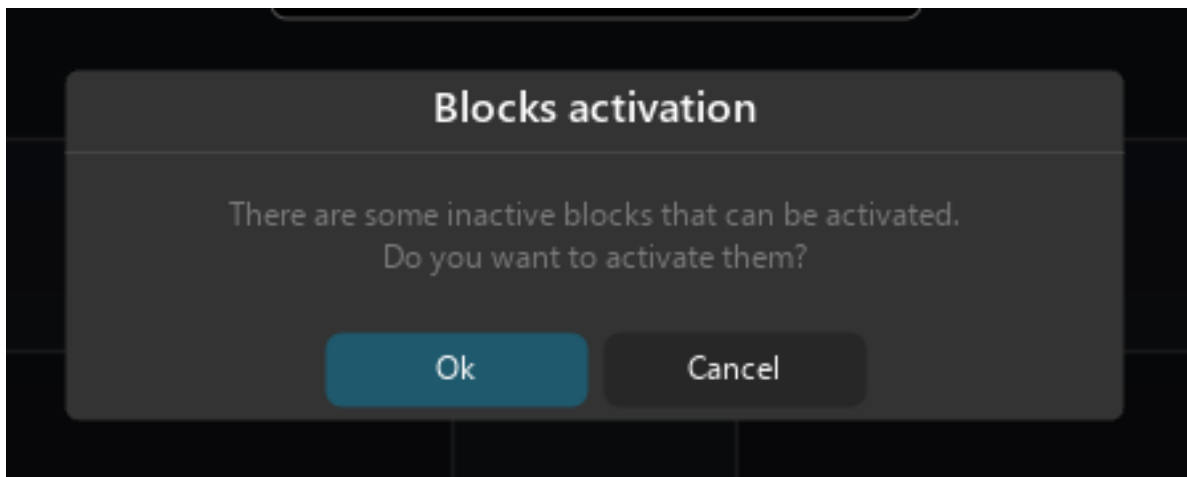
Evidently, if a block has been deactivated for license limitations reasons, the block cannot be manually activated.

26.2 Opening an Ultimate session with the Essential version

If a session contains non-authorized objects or configurations, the concerned blocks are deactivated and the session behavior is the same as described in the “Create and edit a session in SPAT Essential” section above.

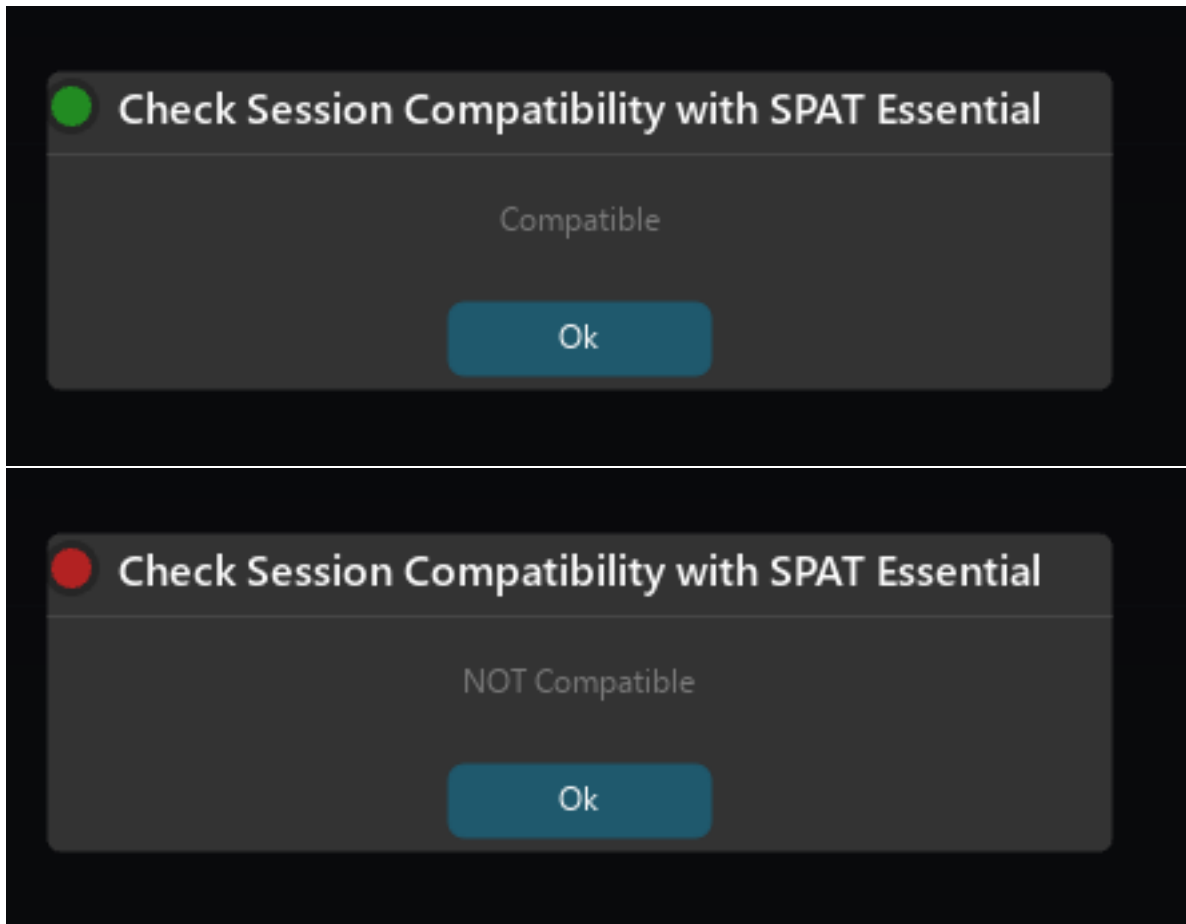
26.3 Opening an Essential session with the Ultimate version

If a session contains inactive blocks that could be activated, a dialog offers to activate them:



26.4 Check Essential compatibility in SPAT Revolution Ultimate

In the Menu File, click on the Check Essential Compatibility item to check if the current session is compatible with the Essential version restrictions. A dialog then informs of the compatibility.



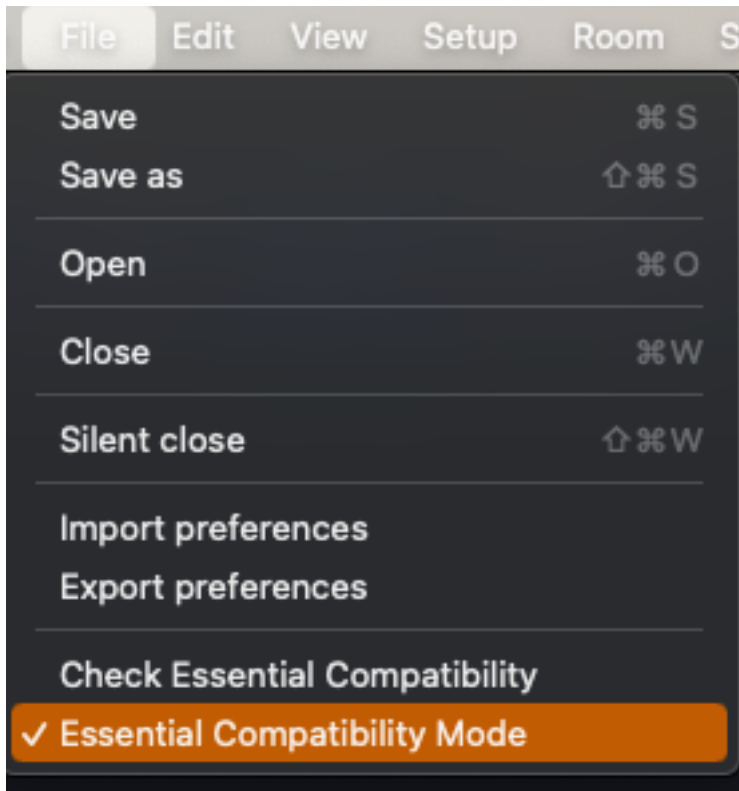
If the session is not compatible, it can be opened in the Essential version but the non-authorized objects will be automatically deactivated (see paragraph “Create and edit a session in *SPAT Revolution Essential*” above).

i Note

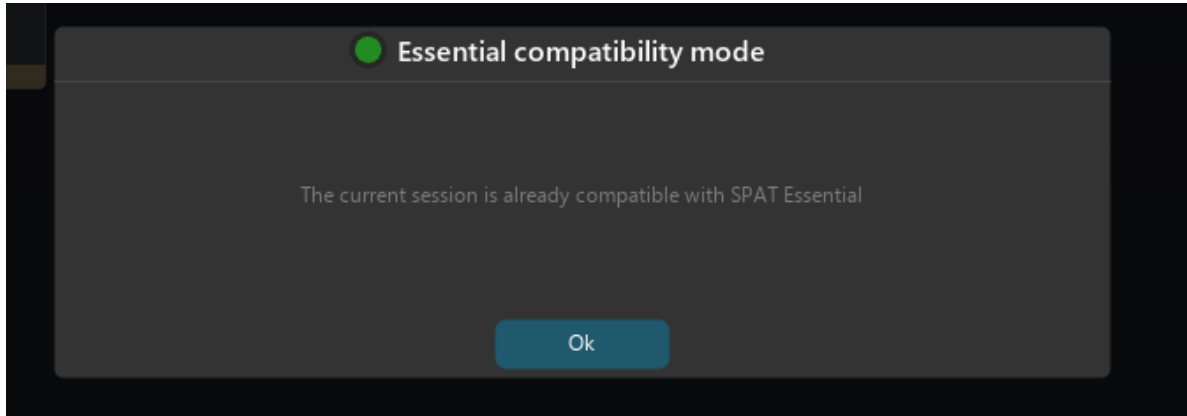
The blocks contained in an Ultimate session can be manually deactivated to fit the Essential version restrictions. The check for compatibility answers that the session is compatible.

26.5 Essential Compatibility Mode for Ultimate version

When in the Ultimate version, a Compatibility Mode allows it to behave as an Essential version (see paragraph “Create and edit a session in *SPAT Essential*” above).



If the current session is compatible with the Essential version a dialog informs the user:



Part VII

SPAT Mixing Engine

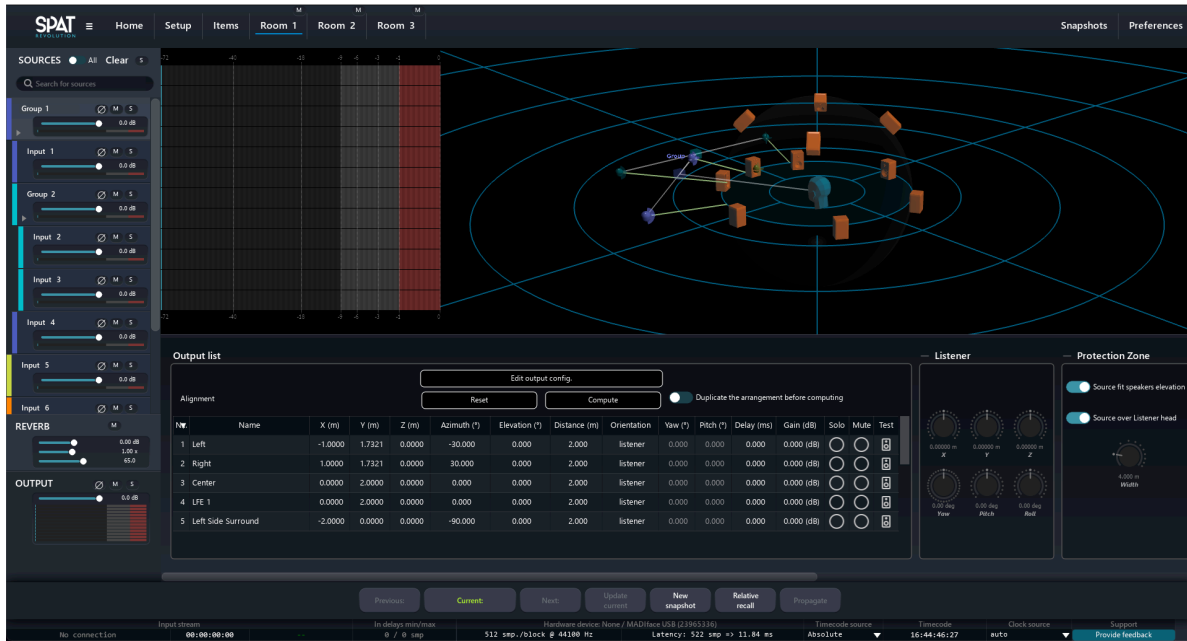


Figure 26.1: Main mixing view of SPAT Revolution

SPAT Revolution mixing engine takes advantage of object-based mixing. Such engines are generally presented in two parts:

- First, there are the *objects* (called **Sources** in SPAT Revolution), which contain all the mixing data, such as source position, gain, acoustic behavior, etc. Objects don't process audio, they're more like a set of instructions, like a recipe.
- Then you have the *decoders*, which look at the instructions in the sources, look at which loudspeaker array is connected, and try to make the best of it. In SPAT Revolution, the job of decoding the source metadata is done by **Rooms**. Rooms are very powerful decoders that can be configured to use many different spatialization techniques, as well as create realistic acoustic simulations.

In the following chapters, we will look at these sources and rooms in detail and discuss all their parameters.

27 Source

A source in SPAT Revolution is a very special object that associates metadata with some audio channels. These metadata are actually all the source parameters that we will discuss in the next chapter. Once we have this association of channels and mixing information, we can send it to a room that acts as a decoder, or interpreter if you prefer. This explains why **only sources can be connected to rooms**.

The various source parameters define the source's simulated positional information, psychoacoustic 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 the corresponding in-lane in the inspector. Alternatively, grab its 'emitter' object in the 3D room visualization (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 categorized groups with which you can alter the properties of the Virtual Source in the Room.

The source's parameters are ordered inside panels. Each panel can be minimized or expanded.

Additionally, a *right-click* on the source lane will bring up some further options, especially useful is the **Color** option, which allows you to set an identification color to a Source or Group.

Note

If a source is a multichannel one, there will be only one number and one set of parameters for the whole cluster.

Warning

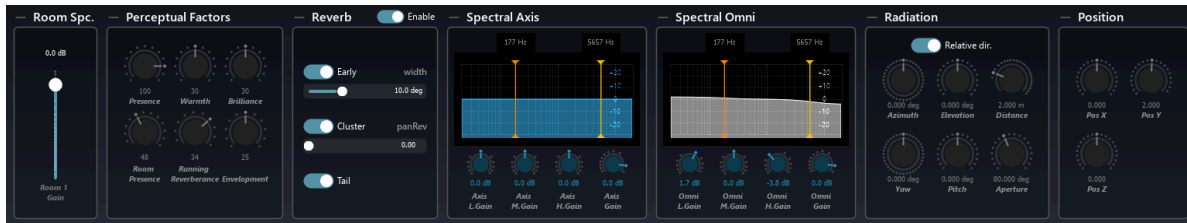
Note that the source number can also be used for OSC automation if the remote number is set to 0!

The source's parameters are detailed in the next section, Source Parameters [28](#).

27.1 Basic interactions

27.1.1 Reset to 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.



i Note

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

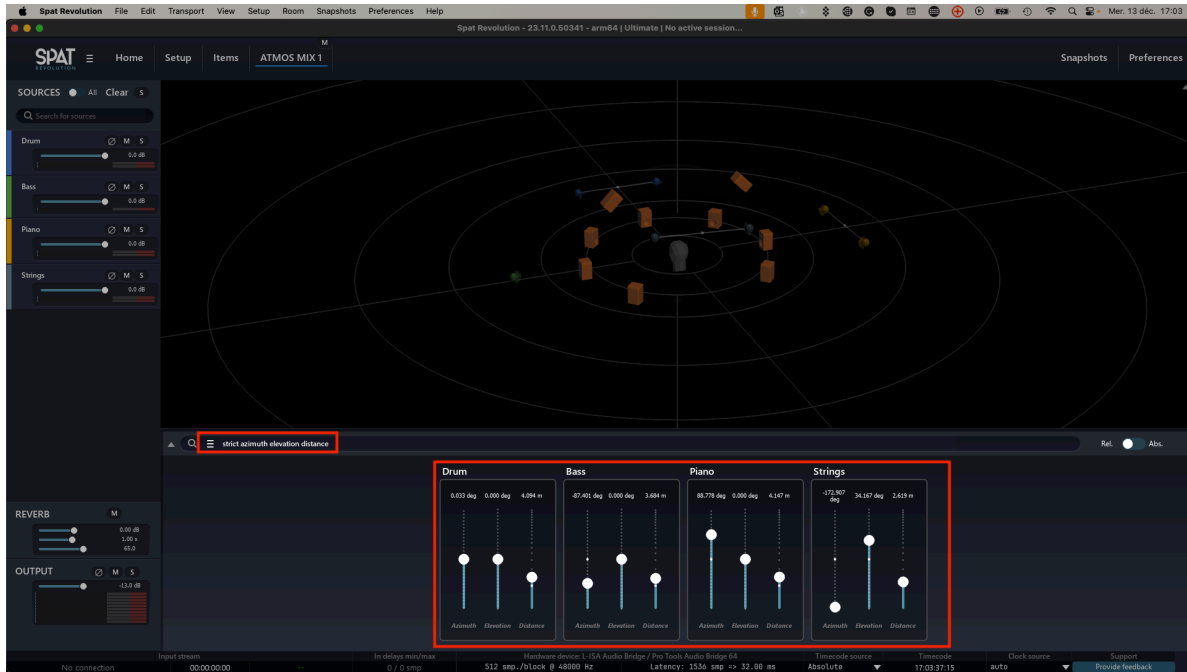
27.1.2 Preset Memories

Each parameter can 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 saved memory slot.



27.2 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.

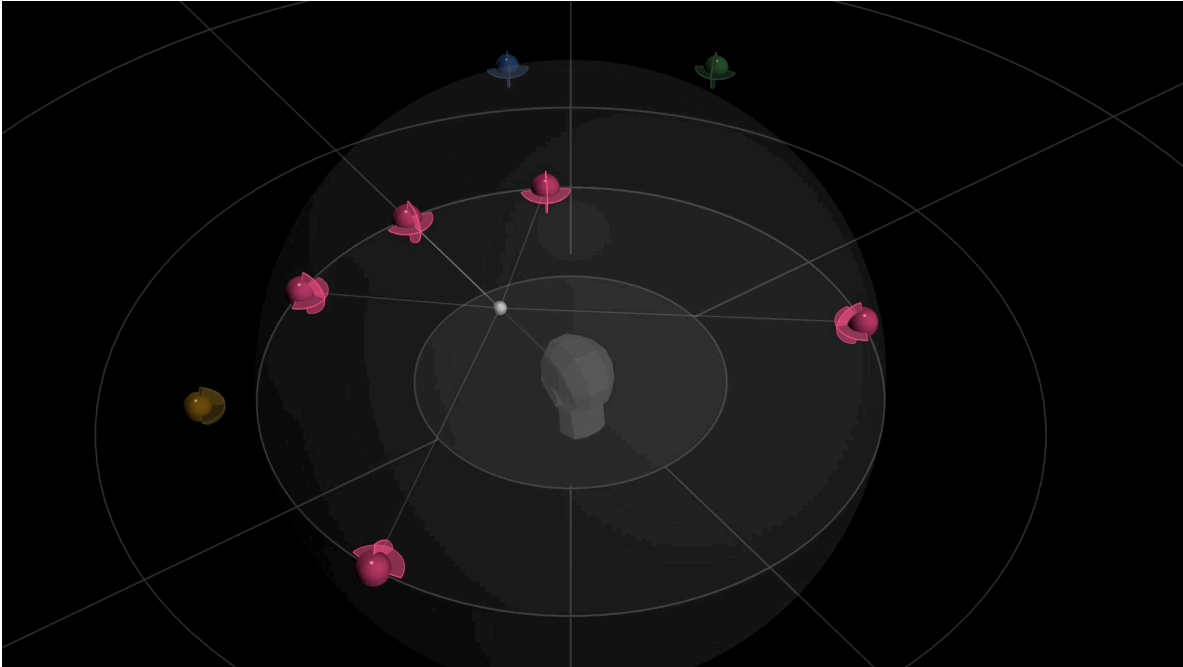


The key word **strict**: allows searching strictly one or several parameters. For example, **strict: gain presence** will display only the sources gain and presence, while **gain presence** will display the gain, the room-specific gain, the presence, the room presence, and all gains of the axis and omni filters.

Some pre-determined presets are available on a menu accessible on the left of this edit to give you some ideas.

27.3 Barycentric Groups

When a 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 visualized 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 about the absolute listener position in the Room (the Dummy Head). The dot at the center 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 focused on.

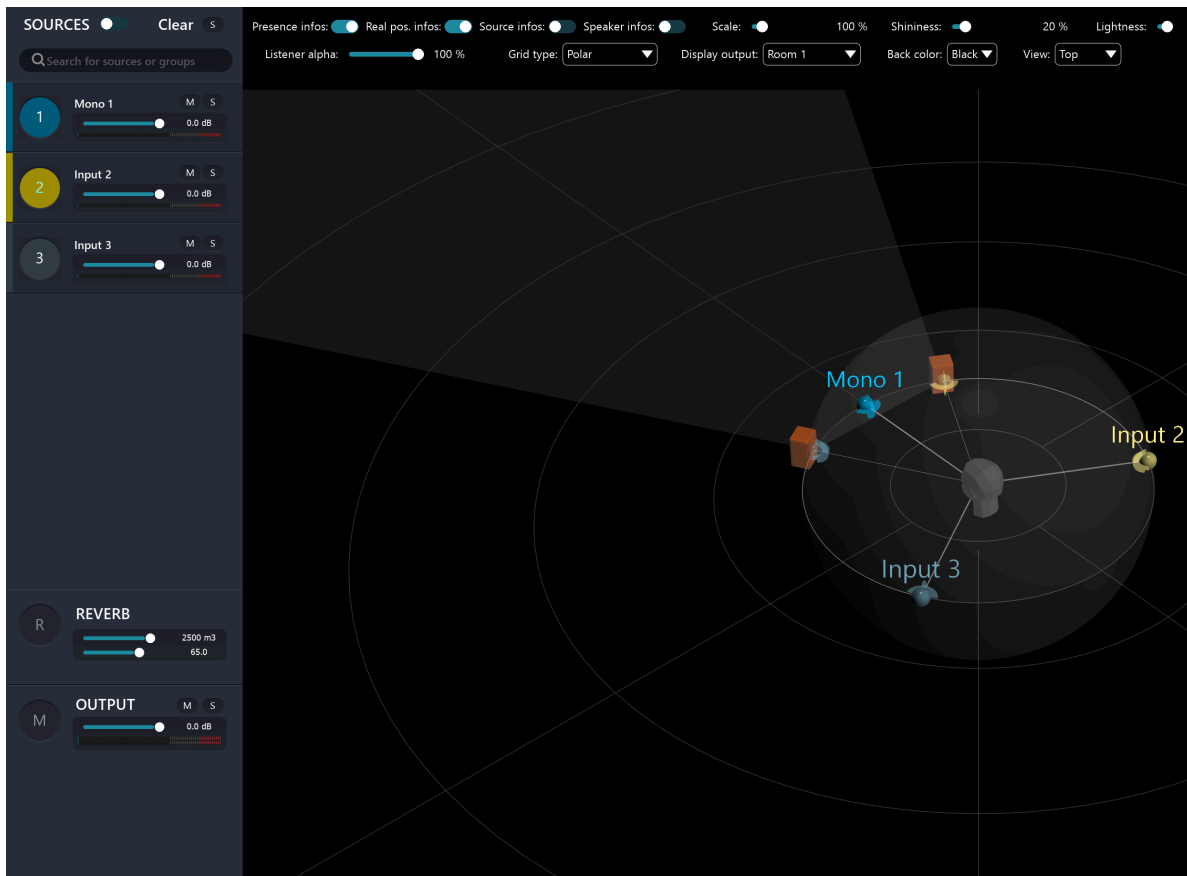
These complex spatial positioning algorithms are computed and controlled in real-time using *SPAT Revolution's* advanced *Barycentric* and relative direction source parameters. A group that may contain many elements can be transformed, scaled, moved, and manipulated in complex ways, through only one set of controls. See the dedicated section for a breakdown of the **Source Parameters** [28](#).

27.3.1 Multiple Source Selection

You can shift-click on the source lane of separate sources to create an ad hoc edit group. When you have group Sources in this way, you can perform several 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 with the Transform panel [19](#)
- generate different colors 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 open the *Source Parameter 28* panel by clicking on the property panel header ‘fold arrow’ as shown in the screenshot below.



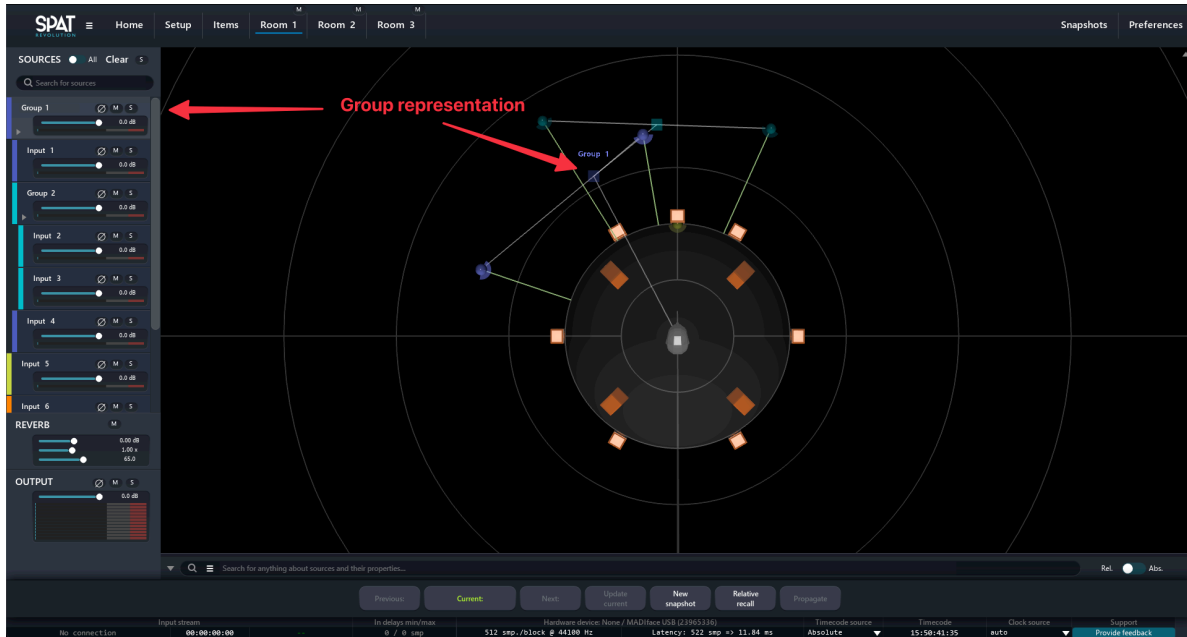
Any source parameter variable you adjust manually will assign that same setting to all selected sources in the group. A barycenter will then become practical to work from a center of mass perspective. For example, transformations like scaling, distance, rotation, and directivity of the group are managed by SPAT controlling each member of the group a barycentric relationship.

27.4 Groups of sources

Barycentric groups are a very convenient way of temporarily editing an ensemble of selected sources. But in some cases, we would prefer the barycentric group to be permanent. That's where groups come in.

A group works very closely with a barycentric group. You can create a group by selecting all the sources you want to include and using the **CMD/CTRL+G** shortcut. In that case, the group appears as a new source in the Inspector and also as a square in the 3D view. If you move

this square, all the sources in the group will move with it. If you move one of the sources in the group, the position of the barycentre will be updated.



Another important point is that **groups can be nested**. In other words, you can have groups inside other groups. The depth of the arborescence is indicated by an indentation in the inspector.

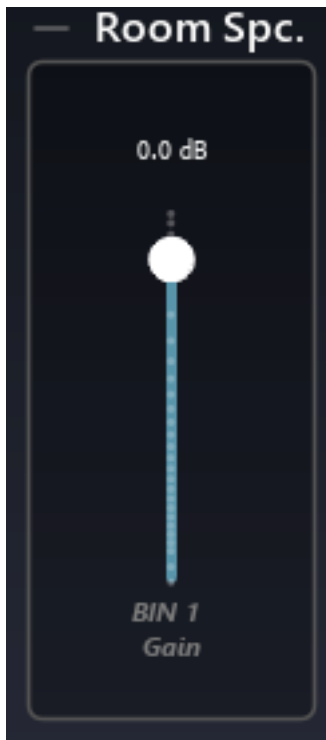
Like a source, a group has an OSC index that can be used for external control or automation. The easiest way to find this number is to look at the Item page.

Warning

It is recommended that you only record automation for sources and not for groups, otherwise, you may end up playing the same automation twice, causing all sorts of strange and unpredictable behavior.

28 Source Parameters

28.1 Room specific gain



New on the 22.02 build, this parameter allows trimming the source gain on each room. A source can so have a different level for each room, which can be useful for monitor, musician headphone monitor, N-1, ...

Only the parameter dedicated to the current room will be displayed, but it's possible to display all the specific gains using the filter and selecting "Room gain".

28.2 Perceptual Factors



This parameter group holds settings affecting the way the source’s 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 is used in the design, where a test panel of listeners is presented with a test set of sounds, constructed from several variations of the reverb engine’s inner parameters. The listeners are then asked to rate each set on a few different scales with perceptually and aesthetically meaningful names. Using principal component analysis (PCA) and optimization techniques, we then built an algorithm that 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.

28.2.1 Presence

The source's 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.

28.2.2 Warmth

Presence of the low-frequency content part of the source.

28.2.3 Brilliance

Presence of the high-frequency content part of the source.

28.2.4 Room Presence

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

28.2.5 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.

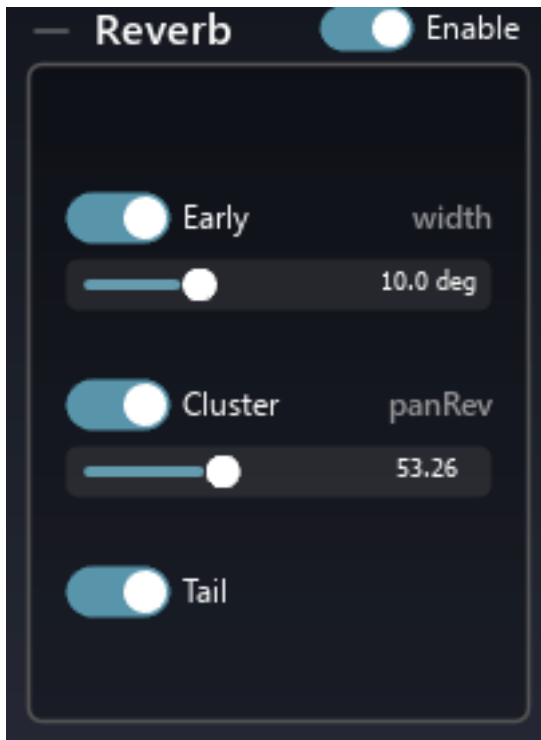
i Note

This setting is not the same as the 'reverberance' in the *Reverb Properties*.

28.2.6 Envelopment

Envelopment corresponds to 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.

28.3 Reverb Options



28.3.1 Reverb Enabled

Toggle whether a source will use the reverberation engine.

28.3.2 Early / Cluster / Tail

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

Early refers to the *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 localization.

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 localizability of a sound source in a space, but instead gives a sense of depth and ambiance.

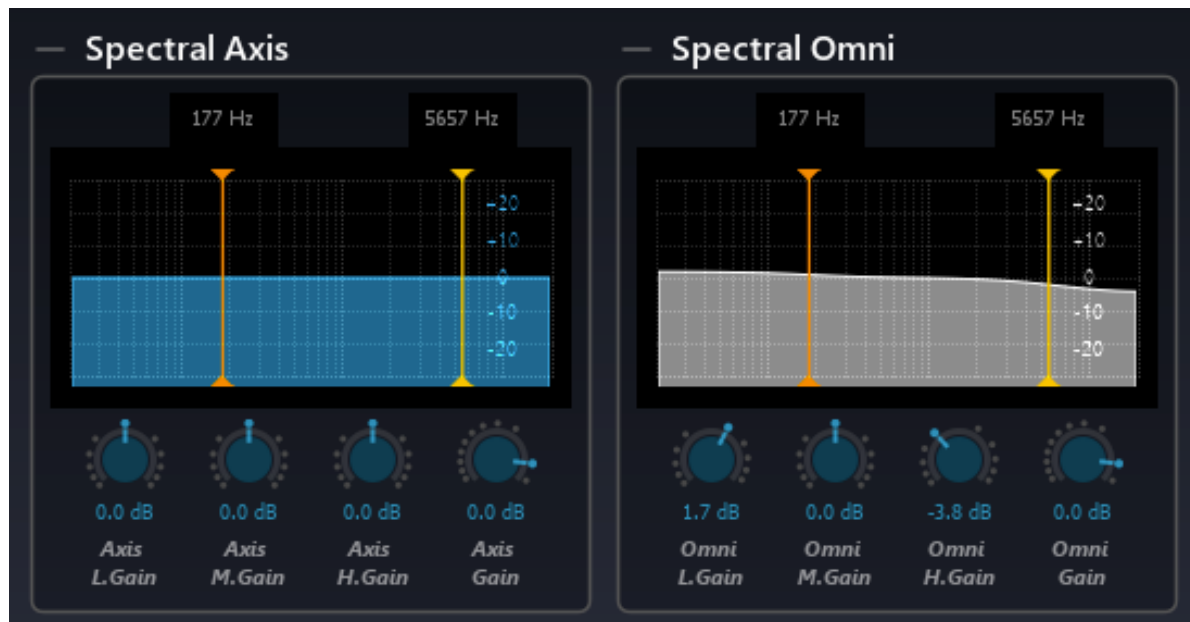
28.3.3 PanRev

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.

28.3.4 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 omnidirectional.

28.4 Axis / Omni Filters



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

28.4.1 Spectral Omni

This filter section is for equalizing the omnidirectional part of the sound radiated by the virtual source. This equalizer mimics the global frequency response of the source, similar to how a loudspeaker colors the sound.

28.4.2 Spectral Axis

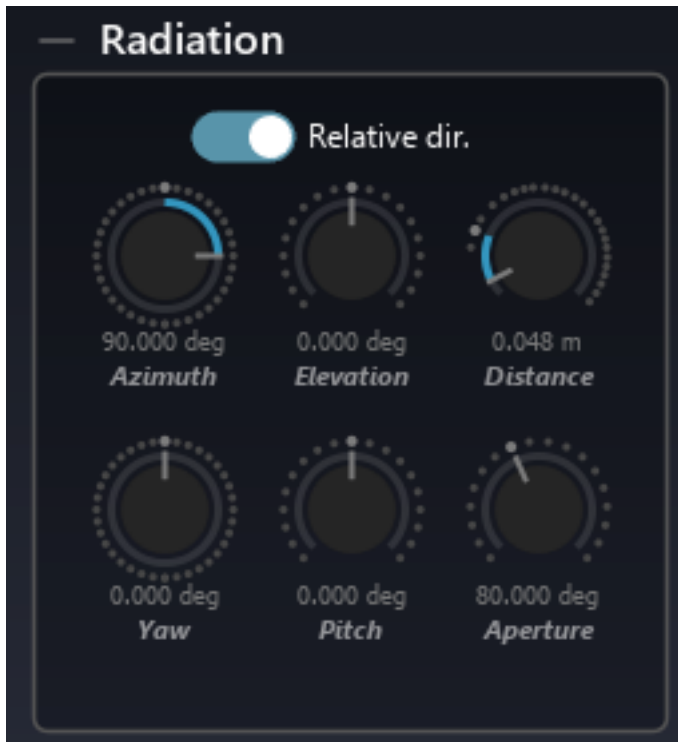
This filter section is for equalizing the on-axis part of the sound radiated by the virtual source. Most, if not all commercially available loudspeakers do exhibit a radically different frequency response whether a listener or microphone is right in front of it or to the side.

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.

i Note

Spectral Axis performs more like conventional mixer EQ

28.5 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.

28.5.1 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 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.

28.5.2 Distance

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

28.5.3 Azimuth

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

28.5.4 Elevation

Elevation angle, in degrees.

28.5.5 Yaw

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

28.5.6 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.

i Note

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

28.5.7 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.

i Note

Aperture can make a source ‘activate’ more of the acoustic space.

28.6 Send LFE



This panel will only be available when the room is a Channel-Based Format which features one or several LFE speakers in its configuration, such as DTU7.1 / 5.1 / 22.2, AURO, and ATMOS for example.

It will send an amount of the source into the dedicated LFE speaker channels of the output channel-based configuration.

i Note

Automate the LFE sends for dynamic low-frequency effects.

28.7 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.

28.7.1 Rotation XYZ

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

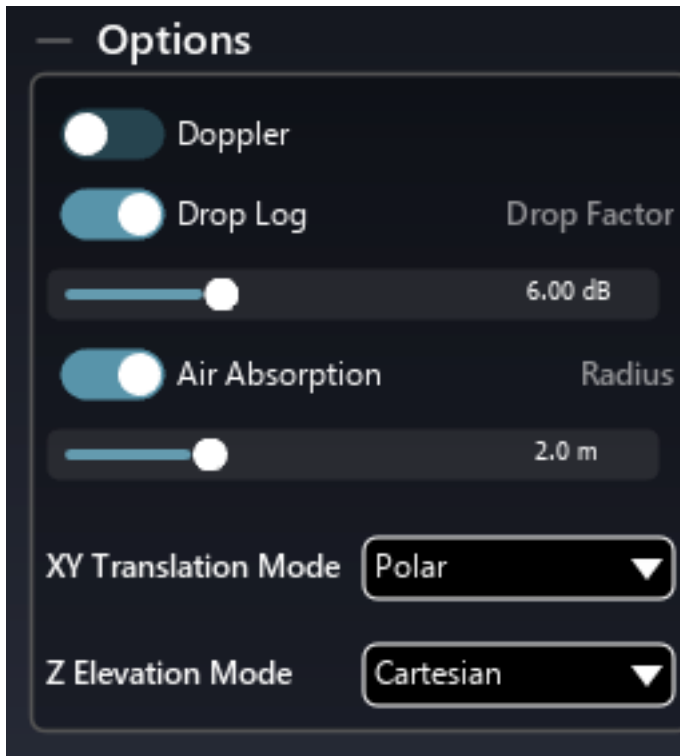
28.7.2 Scale

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

28.7.3 Relative Direction

The barycentric transformations will continue to orient their on-axis energy towards the listener position if the relative direction algorithm is enabled.

28.8 Options



Finally, there are some options available for each source. It is possible to edit a global parameter on the room panel on the preferences page.

28.8.1 Doppler

The Doppler effect is a well-known wave propagation phenomenon where the height of a sound perceived from a listener's standpoint rises when the source is accelerating and falls when decelerating. This is the fire siren pitch going up and then down when passing you. It will only be heard if you quickly move the source 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.

i Note

Careful with this effect: it adds the delay corresponding to the distance between the source and the listener point.

28.8.2 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 behavior). The default *Drop Factor* of 6 dB is also the acoustically accurate setting.

28.8.3 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.

28.8.4 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 regarding distance. This is not only useful to prevent any dramatic sound level peak when placing a source too close to the listener, but it also reflects real-world behavior 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.

i Note

The radius is now deprecated, except in rare cases where you really want a source to have a different behavior than others. Please prefer the room protection zone width to prevent the audio drop.

28.8.5 XY translation mode

Specifies the two different behaviors:

- For multichannel sources, when set to polar, the source orientation will follow the listener. When set to cartesian, it will face the X-axis.
- For snapshot recall, if set to cartesian, the source will move to the given point with the shape of a straight line, whereas the shape will be circular when set to polar.

28.8.6 Z translation mode

Changes the behavior of the translation mode regarding the Z-axis. The behavior is the same as in XY translation mode.

28.9 Spreading



28.9.1 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 focusing effect but will translate differently across certain channel-based speaker arrangements according to how many speakers are involved.

Warning

The spread factor is not available on binaural and WFS.

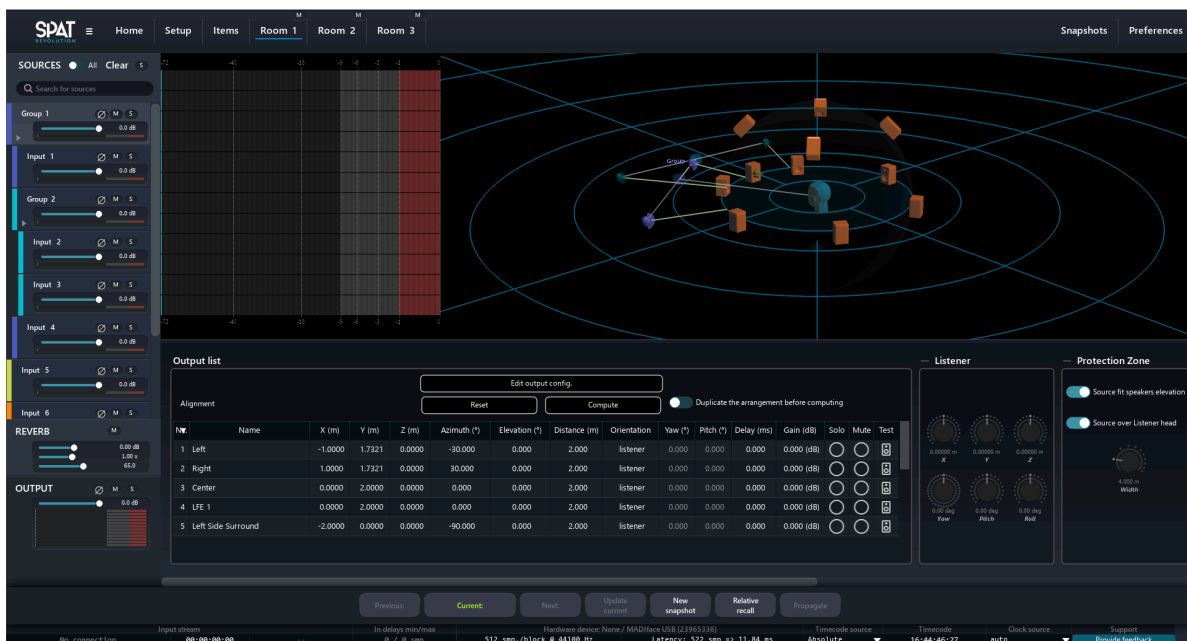
28.9.2 Nearest Neighbors

This parameter is only available to a source if the room it is simulated in has been specified to be using the [`_K Nearest Neighbor_` panning type `-@panType-KNN`]. It sets a maximum limit to the number of speakers that the algorithm can use as neighbors 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 the source. 11%-20% will be 2, and so forth.

29 Room

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

When we enter a Room, we will see the 3D view and all the connected sources. Options related to the 3D view display are located in the top menu. On the left side panel of the room editor, you get a list representation of each source. We can click on an empty part of the source lane, and the source parameters editor for that virtual source will appear.



Mutes, solos, and phase inversions are manageable for all sources and for the entire room output from this index list. A solo clear button is available at the top of the source list. Clicking on it will reset all the solo of all rooms.

Two particular items labeled as **REVERB** and **OUTPUT** appear fixed at the bottom of the left panel. By clicking on these, we then enter into two more parameter editors: one relating to the Artificial reverberation 30 and one relating to the room output configuration and Listener Position 4 editor.

When you have more than one 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 Rooms in the same editor - allowing edition, mix, solo and mute management of all sources from one Room view.

29.1 Room Graphic Engine

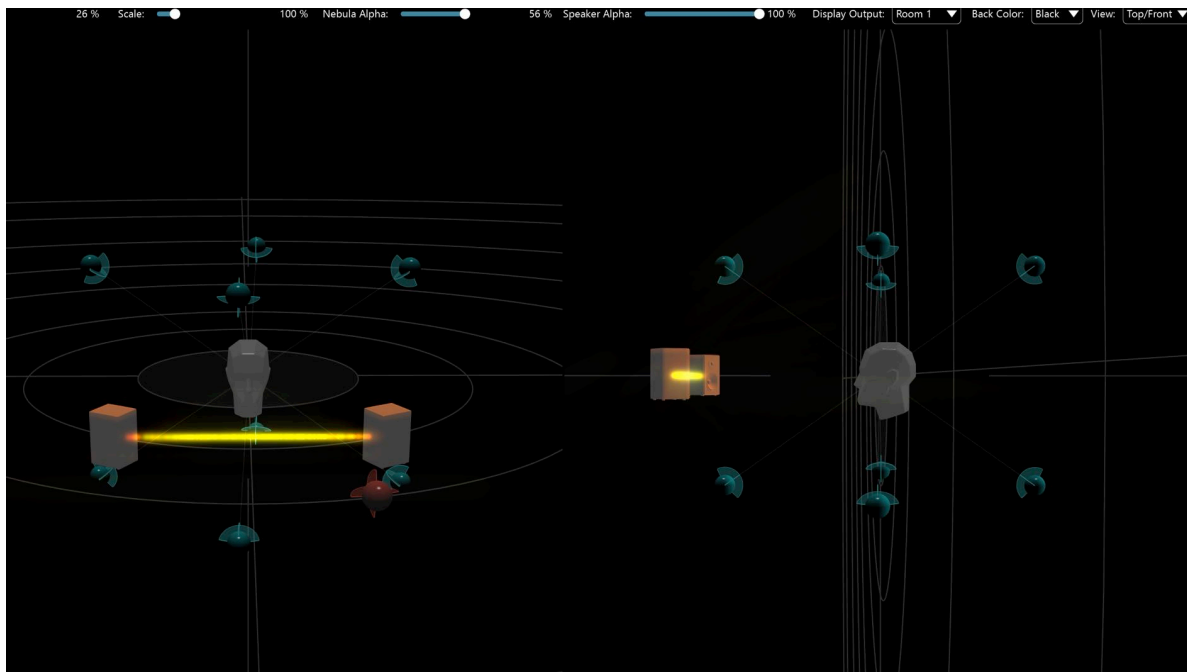
Along with the audio modeling engine, one of SPAT's key features is its ability to model a high-definition graphical representation of the virtual space inside each room. We can intuitively interact and move sources and “camera view” directly with our mouse. Move a source by grabbing its “emitter” object, 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 coordinate-related source parameter controls.

Source manipulation from the 3D view:

- One click to select.
- Drag to move them.

The camera angle can also be manipulated:

- The mouse wheel controls the zoom factor.
- `cmd/ctrl+drag` will rotate the scene.
- `shift+drag` will move the scene.



i Note

A split-screen Top-Front view is also available in the top bar menu. See the corresponding section below.

Selecting a source also displays its name over its 3D representation. At the top of the room, there is an option to display the names of all the sources or all the speakers.

29.2 Room top menu

This menu allows changing what elements and how they are displayed in the 3D view.

Presence Infos:

Display the presence factor as a green vector. The brighter it is, the more present the source is. When off, the green vector is no longer drawn.

Real Pos. Infos:

In some particular cases, the position of the source in the DSP may differ from the one you set up. When on, the DSP position is also displayed.

i Note

To understand better what these two first options do, consult the “Understanding the mixing zones [32](#)” section.

Source Infos:

Display the name of the source even if it is not selected.

Speaker Infos:

Display the name of the speakers.

Auto-hide sources

Automatically hide sources depending on their input level.

Scale:

Make the elements bigger or smaller to adjust ease of sight.

Shininess:

Change the shininess aspect of the graphical elements.

Lightness:

Change the brightness of the graphical elements.

Nebula Alpha:

Change the transparency of the Nebula spectral analyzer.

i Note

When set to 0%, Nebula does not take any resources.

Consult the “Nebula Spatial Spectrogram [20.2](#)” section for more information.

Nebula Quality:

Configure the quality factor of the Nebula.

i Note

Changing this factor will significantly change the performance.

Nebula Scale:

Allows to scale the Nebula “behind” the speakers.

Speaker Alpha:

Change the transparency of the speakers.

Listener Alpha:

Change the transparency of the listener’s head.

Zone Alpha:

Change the transparency of the efficiency zone.

Grid type:

Toggle between polar or cartesian grids.

Display Output:

Allows to display the Nebula of a different setup module connected to this room. It can be helpful if you want to see the decoding of an ambisonic stream inside an ambisonic room, for example.

Background color:

Change the background color.

View:

Choose if the 3D view is seen from:

- The top
- The front
- Or a split view: top and front

29.2.1 Display output drop-down

Located on top of the 3D view, the “display output” drop-down allows choosing which point of the signal path to display.

For example, working in an HOA room creates a 3D view without a speaker. This is because of the very nature of how ambisonic work. But it also means you cannot use Nebula in that room. This is where this “display output” feature becomes handy. Instead of showing the actual HOA scene, it is possible to look at the sound scene at the transcoding stage to see what happens with Nebula on the speaker array.

29.3 Room output parameters

29.3.1 Output list

The screenshot shows a dark-themed 'Output list' panel. At the top, there is a button labeled 'Edit output config.'. Below it, there are two buttons: 'Reset' and 'Compute'. The main part of the panel is a table with a dropdown arrow on the left. The table has the following columns: Name, X (m), Y (m), Z (m), Azimuth (°), Elevation (°), Distance (...), Orientation, Yaw (°), Pitch (°), Delay (...), Gain (dB), Solo, Mute, and Test. There are five rows of speaker data.

	Name	X (m)	Y (m)	Z (m)	Azimuth (°)	Elevation (°)	Distance (...)	Orientation	Yaw (°)	Pitch (°)	Delay (...)	Gain (dB)	Solo	Mute	Test
1	Left	-1.0000	1.7321	0.0000	-30.000	0.000	2.000	listener	0.000	0.000	0.000	0.000	<input type="radio"/>	<input type="radio"/>	<input type="button" value="Test"/>
2	Center	0.0000	2.0000	0.0000	0.000	0.000	2.000	listener	0.000	0.000	0.000	0.000	<input type="radio"/>	<input type="radio"/>	<input type="button" value="Test"/>
3	Right	1.0000	1.7321	0.0000	30.000	0.000	2.000	listener	0.000	0.000	0.000	0.000	<input type="radio"/>	<input type="radio"/>	<input type="button" value="Test"/>
4	Left Si...	-2.0000	0.0000	0.0000	-90.000	0.000	2.000	listener	0.000	0.000	0.000	0.000	<input type="radio"/>	<input type="radio"/>	<input type="button" value="Test"/>
5	Right ...	2.0000	0.0000	0.0000	90.000	0.000	2.000	listener	0.000	0.000	0.000	0.000	<input type="radio"/>	<input type="radio"/>	<input type="button" value="Test"/>

This panel lists all the speakers used in the room (when set to channel-based). It allows quick access to the speaker arrangement editor and the compute function. Each output has a **test** button that sends the signal from the signal generator directly to the routed speaker. The signal generator type and level are set in the *Preferences page 21*.

29.3.2 Listener



This panel gives access to the listening point. We can change its position using the X , Y , Z parameters, and its rotations using Yaw , $Pitch$, $Roll$.

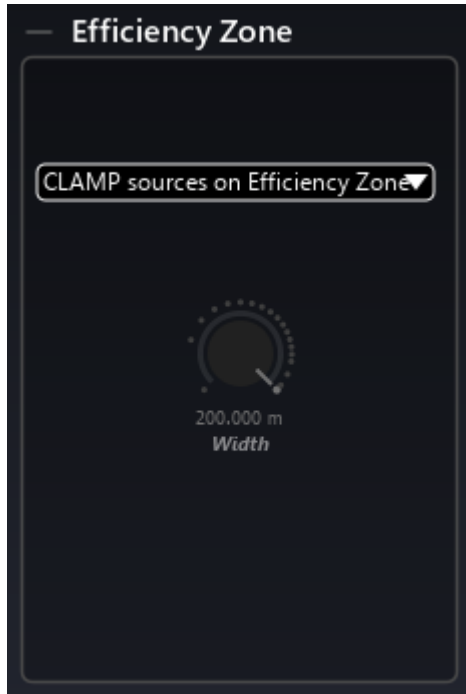
29.3.3 Protection Zone



This panel controls the behavior and size of the protection zone. By default, it is set to a diameter of four meters. Please check out the section named [Understanding the mixing zone 32](#) if you want more information about the protection zone. Note that the protection zone is attached to the listener position.

- Source fit speakers elevation
- Source over listener head
- Width

29.3.4 Efficiency Zone



This panel contains options related to the efficiency zone.

- Clamping behavior option (consult the Understanding the mixing zone [32](#) section for more information)
- Depth - change the depth of the efficiency zone
- Trunc (available only for non-surrounding 2D speakers' setup) - change the starting distance of the efficiency zone

29.3.5 Scaling



Distance

This parameter scales all the distance automation (OSC, plugins data, and snapshots) by a manual factor. This factor is adapted automatically when editing the arrangement of the room.

Tracking

This parameter changes the scale of RTTrPM protocol data.

29.3.6 Background Image



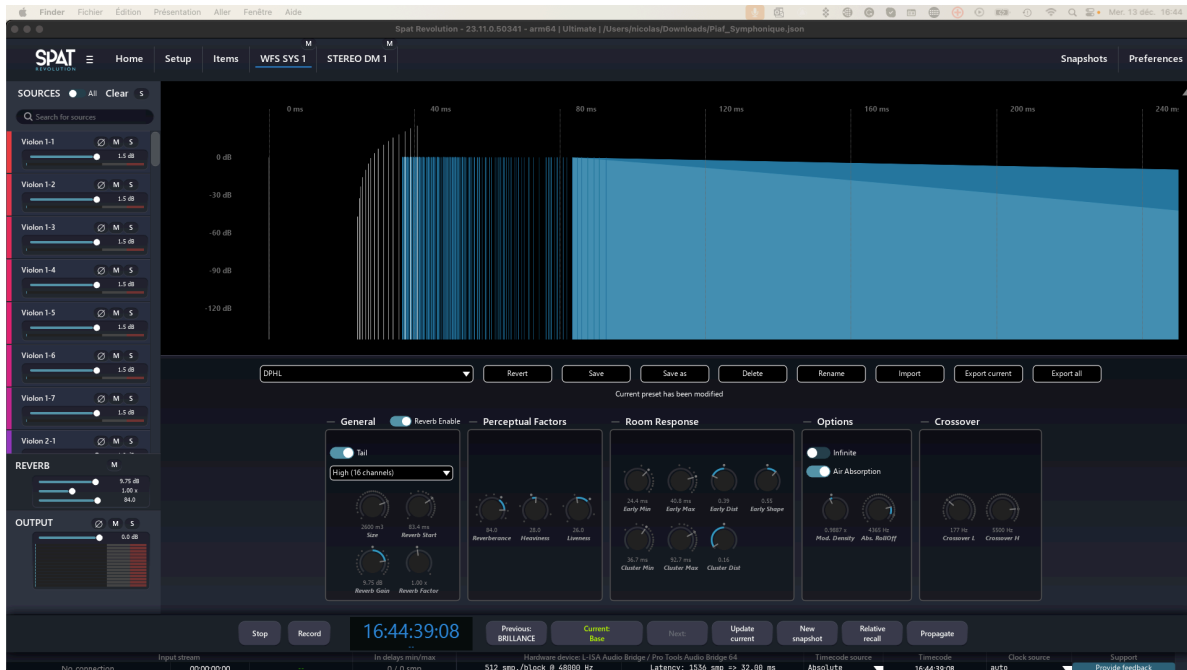
This panel allows you to import a background image in *SPAT Revolution* and to position it in the 3D view.

⚠ Warning

Make sure to have no special character in the path or file name.

30 Artificial Reverberation

Each *Virtual Room* in SPAT can have its own artificial reverberation. Reverberation is essential in the psycho-acoustic perception of localized sources and immersive sound fields. The reverb processor in SPAT is a multichannel algorithmic 3D reverb based on feedback delay networks. The SPAT reverberation engine is designed to synthesize 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 *Reverb* at the bottom of the list of sources on the left side panel of the Room.



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.

i Note

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* 28.2 of sources such as *warmth*, *envelopment* and *brilliance*. These can be found among the parameters for each virtual source.

The SPAT reverb designer can be used for much more than only simulating a “normal” acoustic space. For example, you could design an unreal space with continuously modulating acoustic properties or infinite reverberation.

i Note

Try to switch the Infinite option for an immediately impressive immersive effect.

31 Reverb Parameters

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

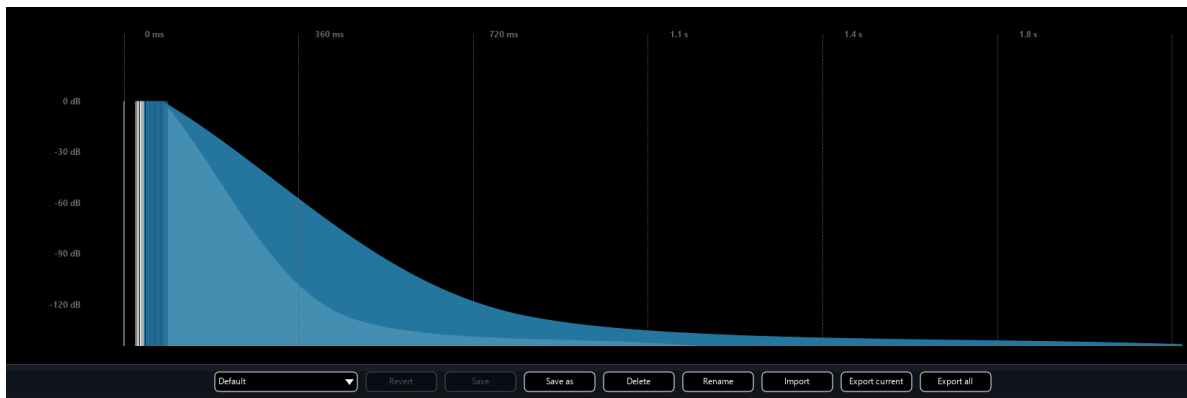
The reverb designer excels at creating static acoustic settings that add dimensionality and immersive depth to a virtual scene. It also invites more creative reverberation ideas. Remember, it works in 3D and interacts deeply with the parametric design of all virtual objects that are expressed through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

This is no ordinary reverb.

The SPAT Reverb is a true acoustic modeling multichannel 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 real-time.

31.1 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.

31.2 Reverb General



31.2.1 Reverb enable

Turn on/off the entire reverberation engine for the room. All sources' components (early, cluster, and tail) will be affected.

31.2.2 Tail

Turn on/off the tail of the reverberation engine for the room.

31.2.3 Reverb Density

Internally, spatial variations are computed using a kind of 3D network of reverb, 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. However, it must

be emphasized that the high-density setting consumes fairly more CPU and that the color of the reverb can be altered by this setting, particularly at some extreme parameter setting combinations.

31.2.4 Size

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

31.2.5 Reverb Start

Reverb start sets the duration between the direct, dry source signal, and the first late reflections, or beginning of the reverb tail.

i Note

Please note that its value can never go below the cluster minimum time, as the reverb tail is fed with a signal derived from the cluster section.

31.3 Perceptual Factors



31.3.1 Reverberance

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

31.3.2 Heaviness

The relative decay time of low-frequency content.

31.3.3 Liveness

The relative decay time of high-frequency content. Describes the liveness and movement associated with the reverb tail (late reflections).

31.4 Room Response Parameters



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

Cluster refers to a secondary iteration of room response reflections and is 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 localizability of a sound source in a space, but instead gives a sense of depth and ambiance.

31.4.1 Early Min

Early reflections minimum time, i.e., the time at which the early reflections appear, in milliseconds. This is a similar setting to the ubiquitous “pre-delay” found on most reverberation processors. It represents the time between the direct sound and the first early reflection.

31.4.2 Early Max

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

31.4.3 Early Dist

Early reflections’ distribution. Determines how 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.

31.4.4 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 acoustic space where reflective surfaces are located roughly the same distance from the listener. Below 0.5, early reflections decay with time; above 0.5, they rise with time. Decreasing levels of the early reflections would be typical of a space where most reflective surfaces are grouped at a range closest to the listener.

31.4.5 Cluster Min

See Early Min. Please remember that the cluster is fed with the input of the early reflections’ processor section, as shown on the display.

31.4.6 Cluster Max

See Early Max.

31.4.7 Cluster Distribution

See Early Distribution.

31.5 Options



31.5.1 Infinite

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

31.5.2 Air Absorb

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

31.5.3 Modal Density

Scales the modal density with respect to the current setting, which is internal to the application engine and depends on other parameters such as reverberation time. 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.

31.5.4 Abs. RollOff

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

31.6 Crossover



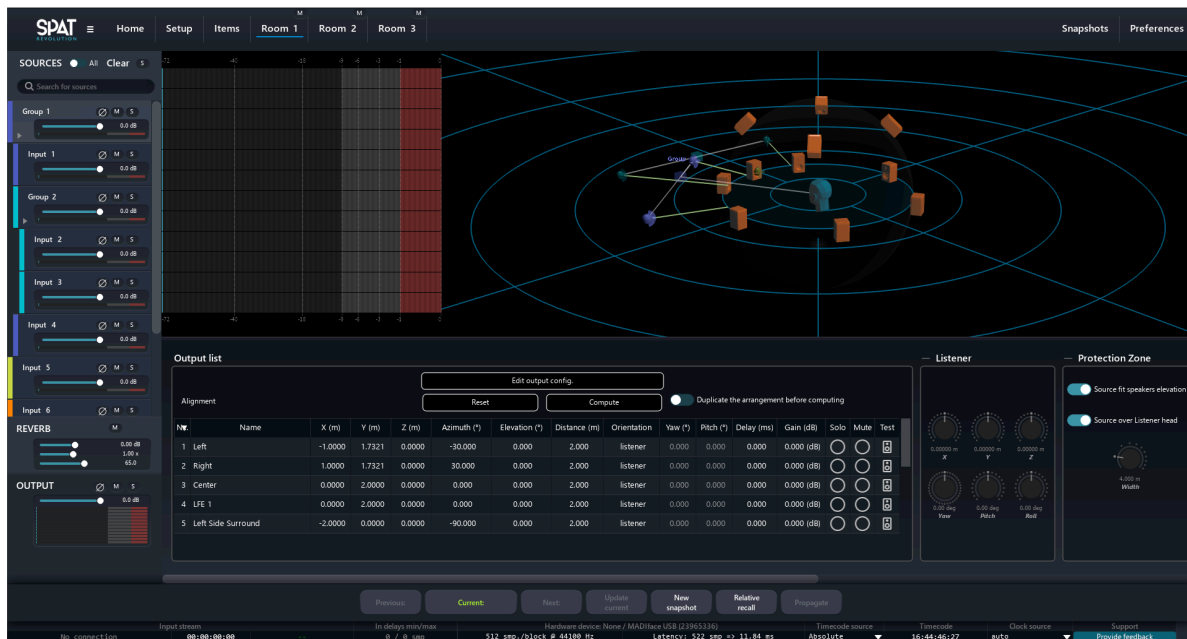
31.6.1 Crossover L

Sets the frequency below which decay time is determined by the heaviness setting, expressed in Hertz (Hz). Default value: 177 Hz

31.6.2 Crossover H

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

32 Understanding the mixing zones



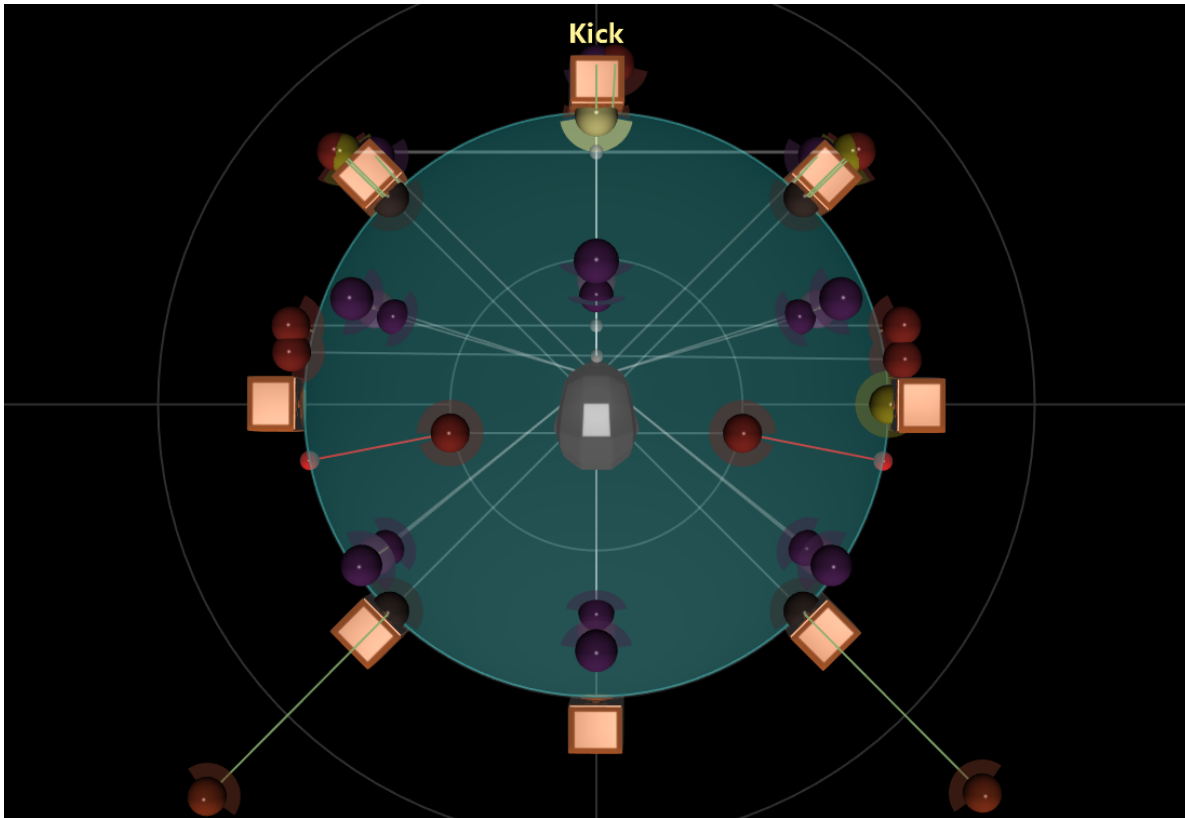
The whole mixing experience of *SPAT Revolution* takes place in the 3D view. Creating a mix this way is really simple: simply grab a source and move it where you want it to be. But there are a few subtleties that are important to get right.

The first important thing to understand is that there are two main zones in the *SPAT Revolution* 3D view:

- One inside which you should put sound sources in: the efficiency zone.
- One inside which you shouldn't: the protection zone.

Whether you are working with a room using channel-based or not, or depending on the speaker arrangement, these zones will have different shapes and usages, which we will discuss below.

32.1 The protection zone



The protection zone is an area of the 3D space represented as a sphere around the listener head. It is important to understand that as most of the pan law or spatialization techniques are incapable of creating the illusion that a sound source comes from in front of the speakers (some exceptions like WFS) **we should try to avoid putting source inside this zone**

When a source is placed inside the protection zone, **its distance will no longer induce a presence factor change**, meaning that the distance won't affect this zone.

When a source is inside the protection zone on a 3D capable stream type/speaker arrangement. The behaviour is set by the **Source over listener head** parameter, which is on by default. It is the legacy function where any source entering the zone follows the sphere of protection in elevation as it tries to enter. Thus, the source is processed at its elevated position. If disabled, the source will simply keep its elevation. In both cases, reaching the protection zone means the attenuation model based on the distance reaches its threshold. (same as for the Air absorption calculation)



Figure 32.1: ProtectionZonePanel

32.2 The presence of a source

A sound source in *SPAT Revolution* has a presence factor. It defines its overall level and brightness inside the virtual acoustic space. Many parameters can change this presence factor:

- The distance between the source and the **protection zone**
- The presence parameter
- The drop factor
- The Air absorption

Putting a source closer or further away from the protection zone will have the consequences of modifying the presence. The closer the source is, the more presence it has. The farther it gets, the less presence has a source.

The presence parameter directly affects the presence factor of the source.

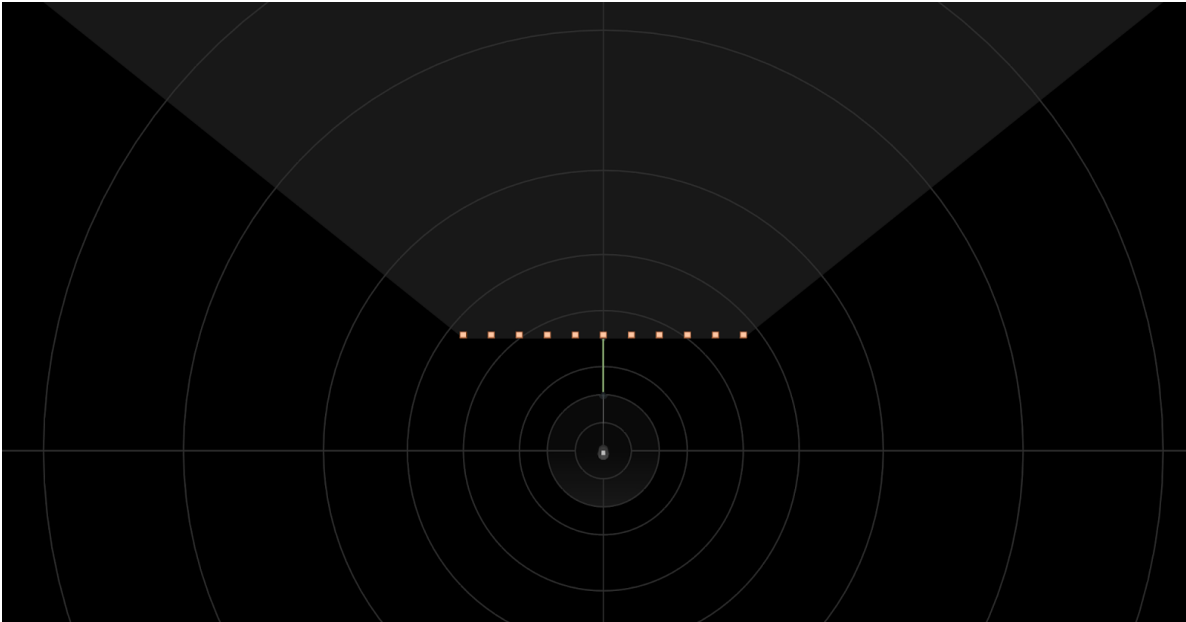
The drop factor defines the relation between the loss of presence and the distance between the source and the protection zone. It is set by default to follow the acoustic law of our world, where we lose 6 dB of presence each time we double the distance.

When **Presence infos** on the top bar of the 3D view is enabled, the overall presence of a source is displayed by a green vector, drawn between the source and the protection zone. The intensity of the green color is proportional to the presence factor. If the source is inside the protection zone, the vector will turn red and a small sphere of the same color will be drawn on the surface of the protection zone.

Warning

****While a source is in the protection zone, there is no variation in presence.**

32.3 The efficiency zone



By definition, the efficiency zone is where the virtual sound sources should be localized. According to the speaker arrangement and coverage capacity, an efficiency zone is calculated for all channel-based panning types. It gets displayed on the 3D view as a grey shadow, and can be considered as a safe source position zone.

One of the advantages of this mechanism is that it allows to change a speaker arrangements while providing a tool to manage the sources already positioned in the soundscape that may be outside the zone.

Inside a channel-based room, the efficiency zone is defined by the speaker arrangement used in a room:

- When using speaker setup that surrounds the listeners, the efficiency zone is a sphere (or circle in 2D), which spans from the border of the protection zone to the farthest distance you can put a source (100 m). A speaker array is considered surround if the angle between the foremost left and right speakers is over 180° . For readability reasons, the efficiency zone is not drawn in this case, but will be if its depth is define to a value inferior to its maximum (100 meters).
- When using a non-surrounding system (stereo, frontal line, etc.), the efficiency zone becomes a “piece of pie” displayed in the 3D view. The angle between the foremost left and the foremost right speaker defines its width. Visualizing the efficiency zone should help understand these systems’ limitations in terms of spatialization options. For

instance, placing a source behind the listener head in stereo will never produce the effect of a source coming from behind.

The “depth” parameter sets the range of the efficiency zone. It sets the maximum span. A use case for defining a maximum efficiency zone Depth in CLAMP mode is to limit the attenuation to the arc part of the zone.

Inside a non-channel-based room, the efficiency zone is a sphere because no speaker constrains the diffusion area. Still, it can be edited with the same “depth” parameters for multi-room and creative applications. More information below.

32.3.1 Behaviour of sources outside the efficiency zone

When a source is out of the efficiency zone, *SPAT Revolution* offers three behaviours:

1. The source is clamped to the efficiency zone (default for new session)
2. The source is muted
3. It does nothing (default for pre 22.02 release sessions)

This option can be set in the room’s output section. Check out the Room section [29](#) for more information.

i Note

By default, the source is clamped to the border of the efficiency zone.

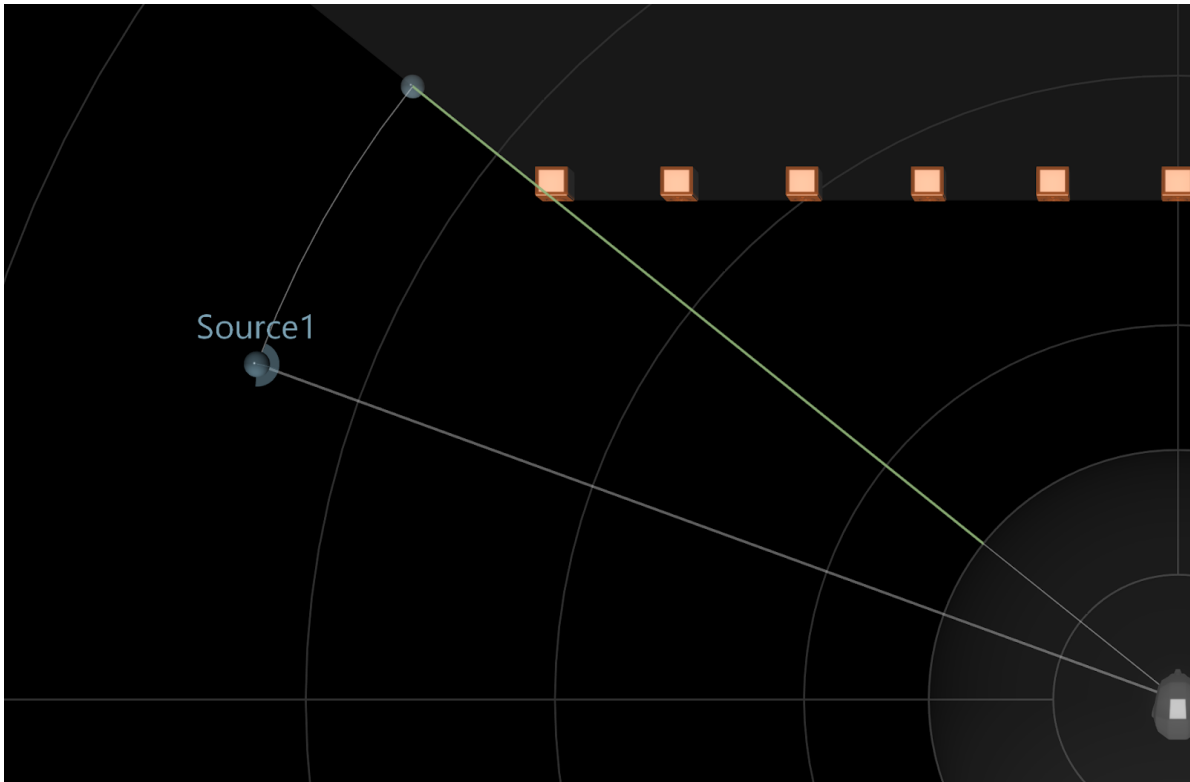
Clamping the source to the efficiency zone helps to keep coherent sound scene, while the “mute” mode helps to create ins and outs effect. The mute only occurs in the actual room.

i Note

When in mute mode, a slight fade out is applied to avoid clicks. The fade’s length can be adjusted by making the source go faster or slower.

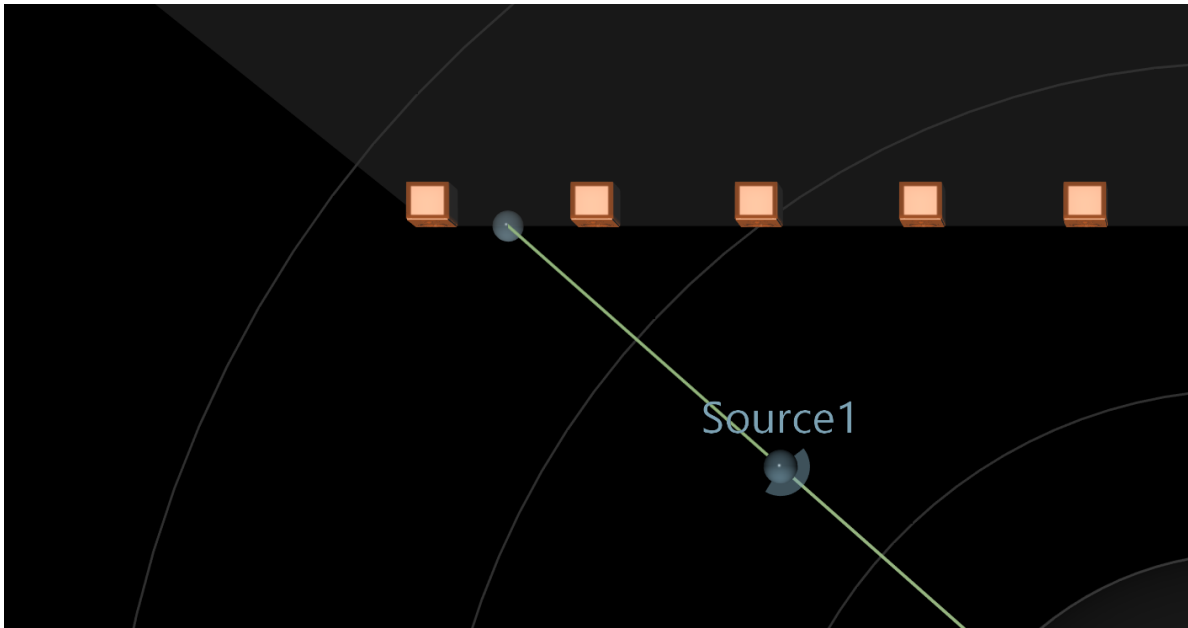
Note that the preferred behaviours are clamping or muting. **Clamping will prevent an aberrant render in sound and localization.**

The pictures below shown some key cases of clamping, on non-surround systems:



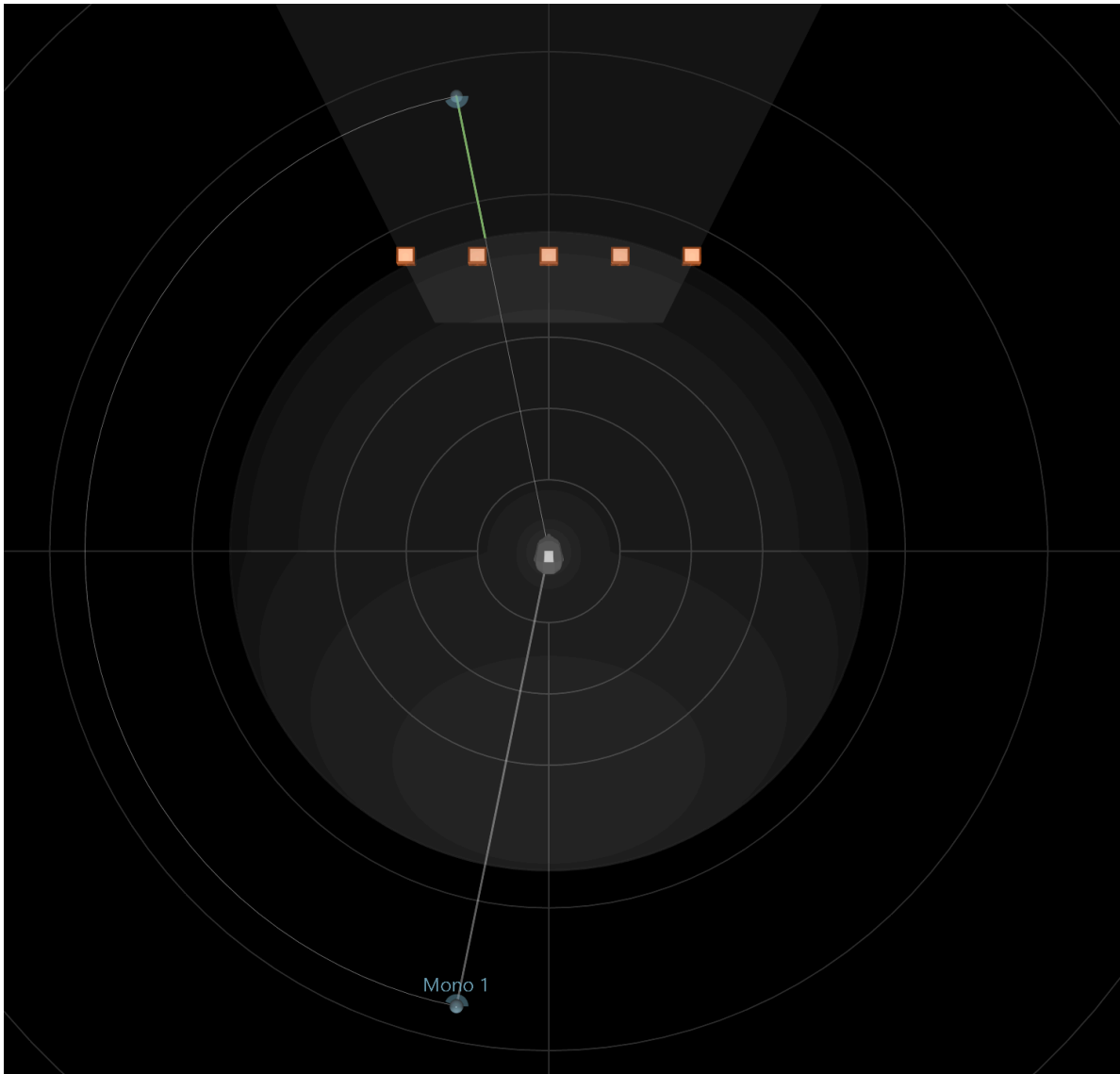
i Note

Trying to escape the efficiency zone on non-surrounding systems will result in the source being clamped to its foremost left or right.



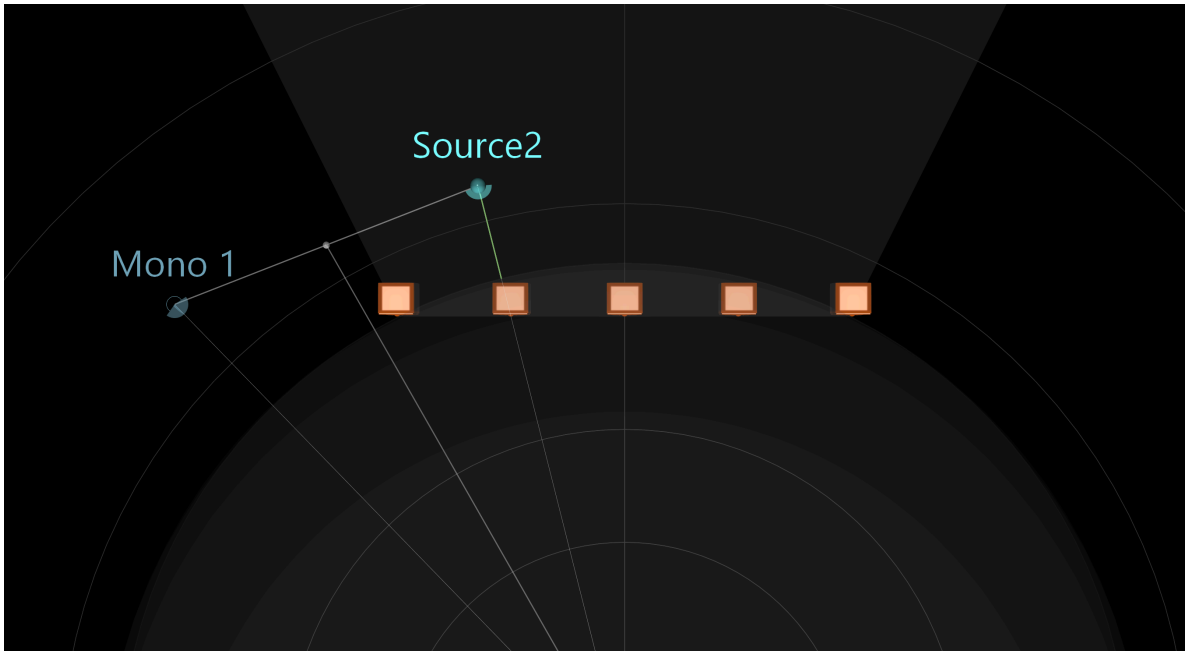
i Note

If a source is placed in front of the speakers, it will be clamped to the front line they formed while preserving the azimuth angle.



i Note

When a source is on the opposite side of a non-surround system, the projection of the virtual source will mirror the behavior of the actual source, thus, removing any jump or abrupt change in position of the source.



i Note

When a source is outside the efficiency zone and mute behaviour is selected, the source is represented differently.

32.3.2 Elevation clamping

The elevation clamping helps keep sources inside coherent space regarding the room speaker arrangement. It only happens with channel-based configurations.

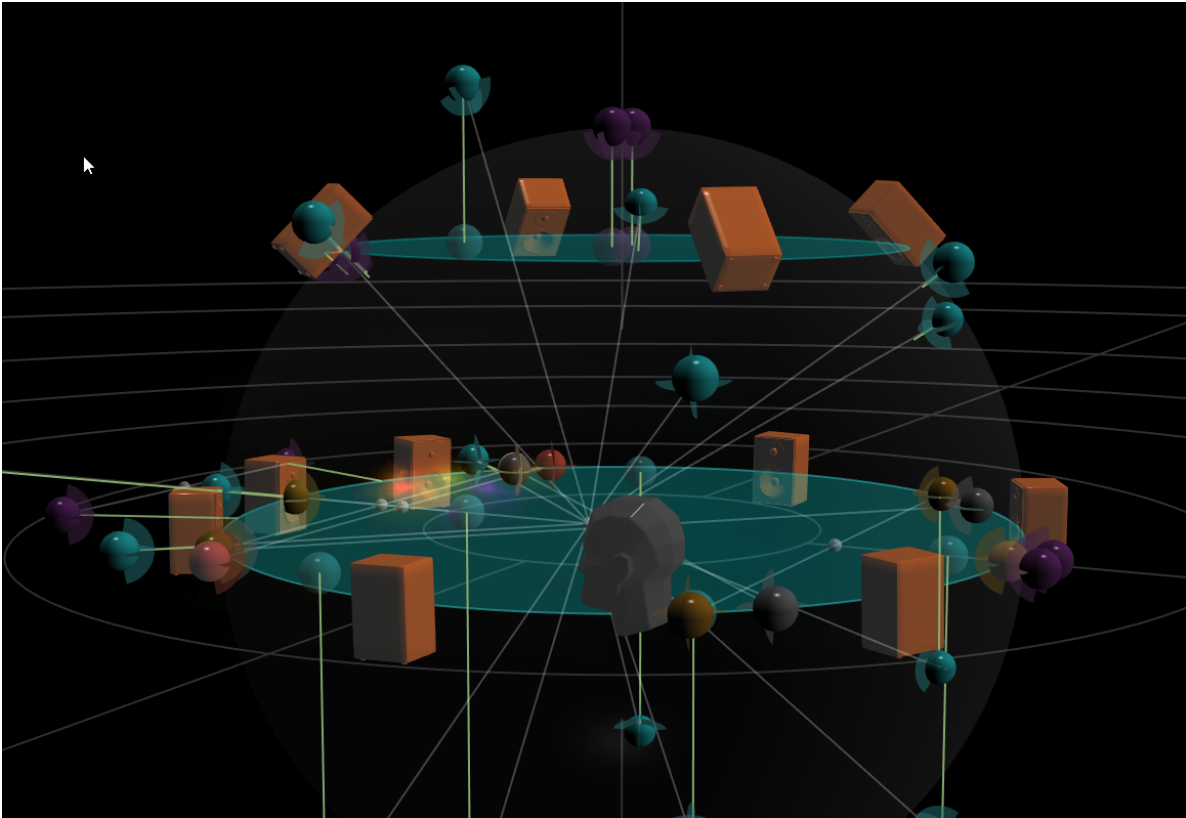
It can be activated by the **Source fit speakers elevation** parameter, in the room options. Elevation clamping works differently depending on if you are working with a 2D or 3D speaker array.

- **2D Speaker Array**

When dealing with a 2D speaker array, there is no point at placing a source above or below the horizontal plan. If you choose to do so, or use a 2D room to translate a 3D mix, you will see phantom sources that are the projection of each source on the horizontal plan. The DSP uses the position of the phantom source to render the actual source position in the virtual space, but, it will preserve its actual presence factor.

- **3D Speaker Array**

Most 3D speaker arrays have at least two speaker layers. Such layers are defined by three non-align speakers sharing the same height (z). Elevation clamping will handle sources as they exceed the extreme layers. Like with 2D speaker array, this clamping behavior is shown with phantom sources that indicate the position used inside the DSP stage of *SPAT Revolution*.



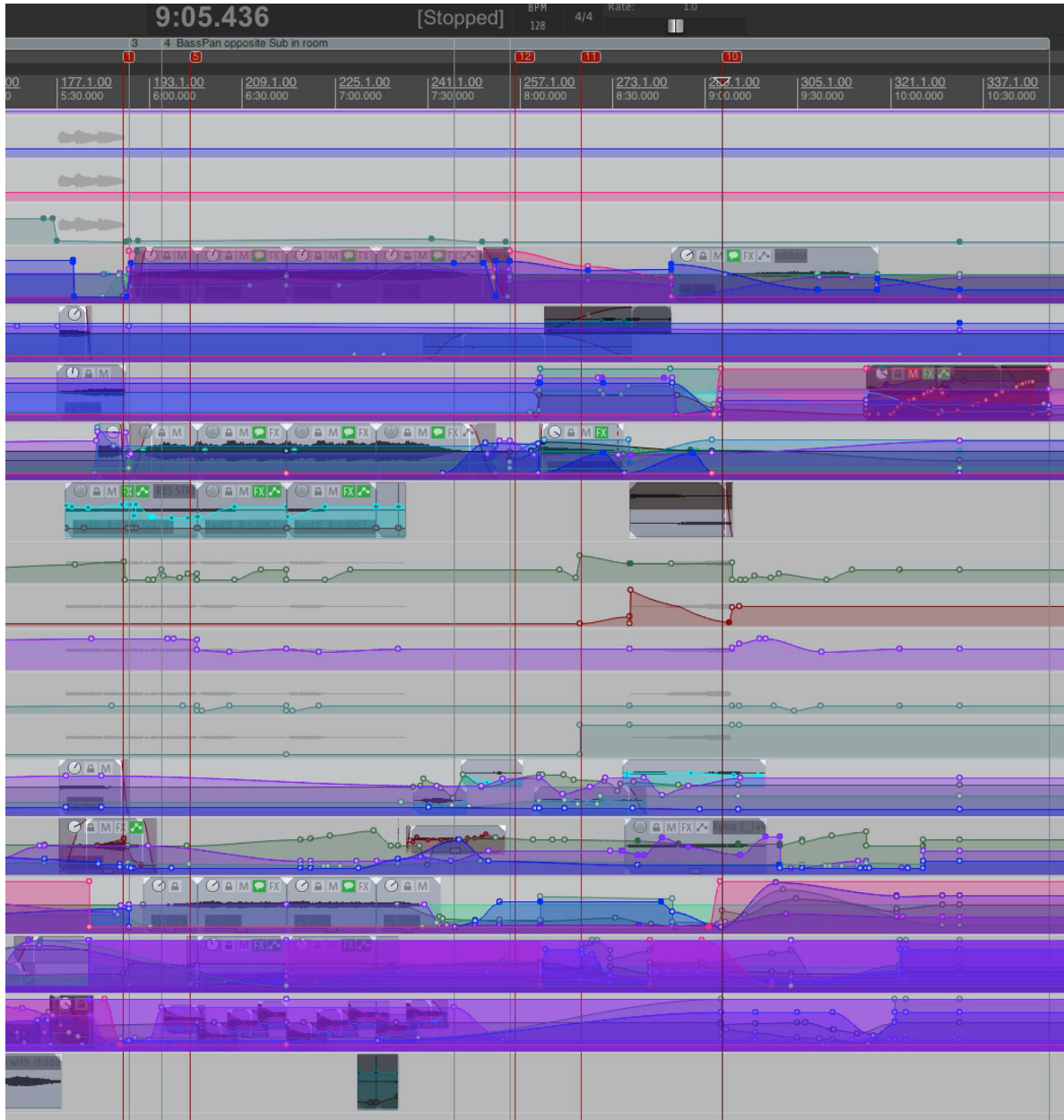
32.4 Limit cases

- Elevation clamping does not occur with distance-based pan law (KNN and WFS) when using 3D speaker array, except for the $Z=0$ plan. This prevents some jumps and aberrations in sound.
- When using a 2D non-surrounding speaker array, efficiency and elevation clamping are tied together. If efficiency clamping is activated, *SPAT Revolution* automatically switches on the elevation clamping. On the contrary, if elevation clamping is turned off, so is the efficiency clamping.
- WFS is the only case where it can make sense to put virtual sources in front of the speakers, to take advantages of the focus zone. This zone and the associated behaviour is detailed in the WFS Section 29 of this user guide.

Part VIII

Ecosystem & integration

Almost all *SPAT Revolution* parameters 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 Revolution* plug-ins in your **DAW**. Parameter control can be played back from precomposed timelines, or performed, generated and captured in various ways. One of the most straightforward ways is to set DAW automation on 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

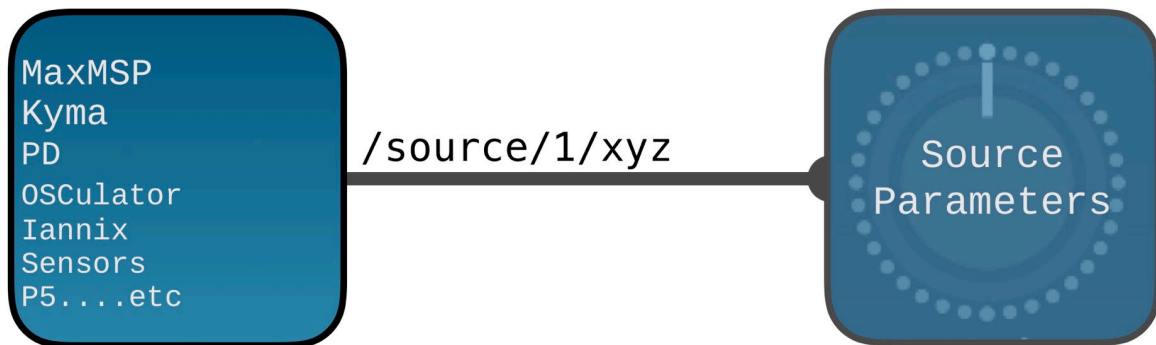
source parameter pots. The movements will be captured into the DAW timeline in precisely the same way you would capture automation from a conventional DAW plug-in, like Reaper 40, Reavolution 41, ProTools 42, Nuendo 45, Ableton Live 43, Bitwig 46, or other.

Alternatively, a wireless OSC control surface could be used to control the parameters of *SPAT Revolution* in real time, while you stand in the middle of the sound system away from the computer. There is already fully programmed remote for *SPAT Revolution*:

- the SPAT Revolution REMOTE 51 can be used on any device as a web-based controller.
- the legacy and discontinued *Lemur* template for *SPAT Revolution* (see Third party integration - Lemur 51).

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 “Third party integration” - QLab 47).

Suppose you are working with algorithmic gesture generators and modulators. In that case, your control signals can be easily sent into the SPAT Renderer via OSC to distribute and control spatial sound sources in real time using your own control programs.



Get creative with spatial sound design using OSC and Spat!

33 Using Hardware I/O

33.1 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 labeled as 'Hardware' IO in SPAT.



Hardware input connections could include:

- Mic
- Line
- MADI
- DANTE
- AVB
- ADAT

Warning

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 multichannel sources using only one input/source module.



33.2 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.

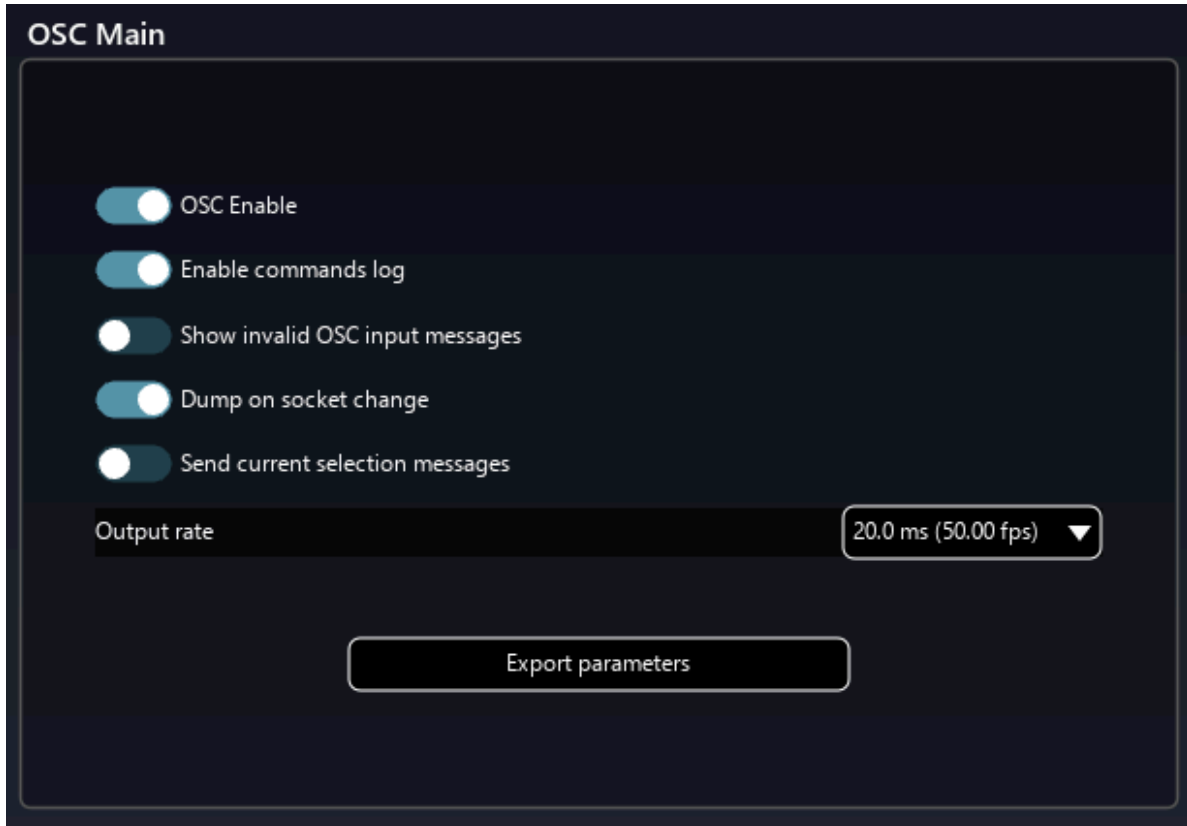
Furthermore, a single machine setup can be converted to a fully automated distributed using OSC to target the single or dual SPAT engine for a real time controllable and interactive distributed hardware setup.

Warning

Enable bidirectional OSC communication in SPAT preferences to send automation over the network to another machine. Contact FLUX Support if you need expert support on this advanced distributed setup.

34 Open Sound Control

34.1 Introduction to Generic OSC



Suppose you are developing your own control systems to integrate with SPAT. In that case, 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. You will also find a complete OSC table [D](#) in the appendix.

i Note

Enable `commands log` will display the received and emitted OSC messages in the log windows to confirm you are receiving data. **Shift + F7** will open the log window.

In general, SPAT OSC patterns have the form of:

```
/source/[remote number]/x
```

where `remote number` represents the Remote number of the Virtual Source, or the index (i.e. the room number) of the 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/[remote number]/xyz Cartesian co-ordinates in meters
```

```
/source/[remote number]/aed Polar co-ordinates (azimuth, distance and elevation)
```

! Warning

Take care to automate EITHER Cartesian OR Polar, not both.

Sometimes, it is more convenient to have the remote number 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 ifff
```

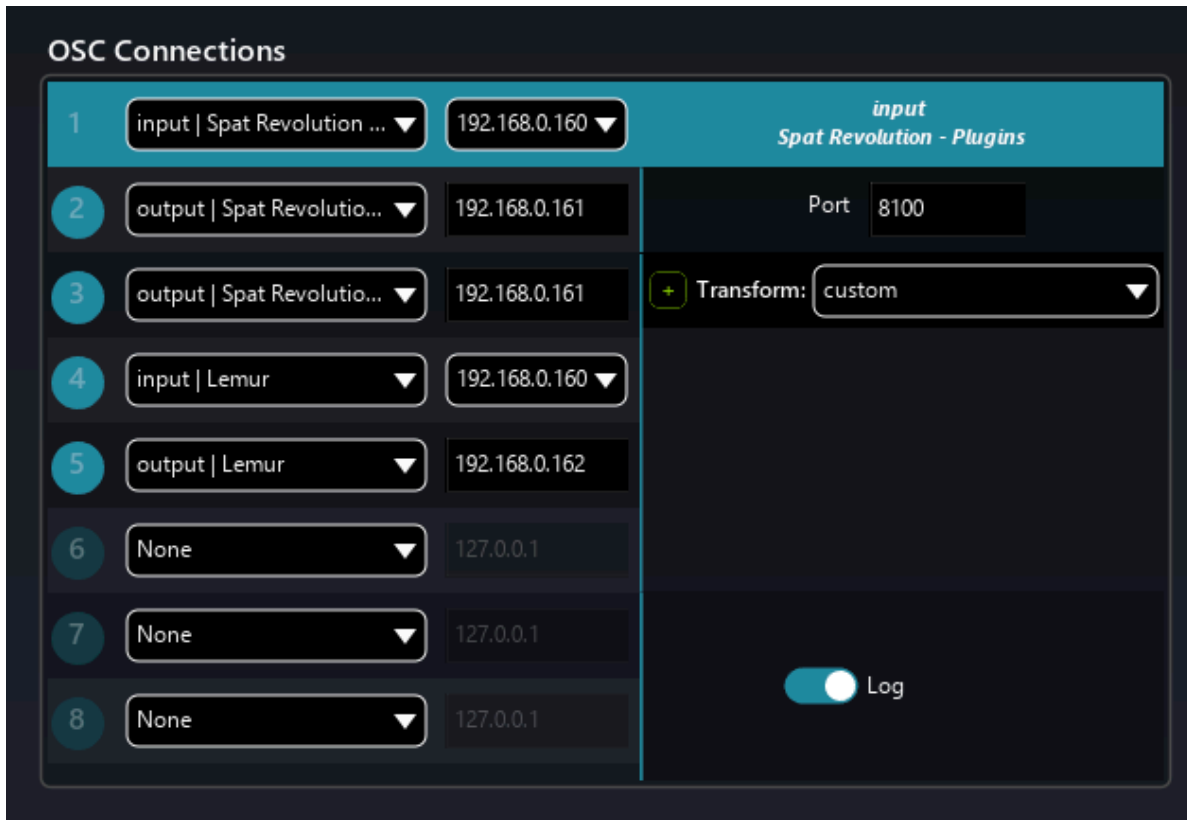
where `i` is an integer denoting the remote number of the target, and `f` according to convention is a float denoting the values of the message.

For more details about the SPAT Revolution OSC dictionary and its syntax, please refer to [Appendix C D](#).

The **output rate** allows changing how fast the engine will send OSC messages. The lower the value is, the more SPAT Revolution will send OSC message and the remote will display the changes, but it will also increase the stress on the CPU. If you experienced some CPU overload when moving sources through OSC command, you can try to increase the value. By default, it is set to 5.0 ms.

34.2 OSC Connections Menu

There is a lot of flexibility in the OSC connection menu. We find it in the SPAT Preferences page.



Eight different connections can be set up as input or output. It is displayed as a table where each connection is a line.

The first parameter is the **connection type**, which offers many input and output presets. These presets come with dedicated network ports, dedicated option and dedicated transform if needed.

i Note

OSC presets are: - SPAT plugins - SPAT Remote server - ADM-OSC - Lemur - Avid S6L - Digico - SSL Live - SpaceMap Go

Then the **IP address** needs to be set. If the OSC connection is established locally, on the same computer, the port 127.0.0.1 is dedicated to local network usage. For other configurations, we need to inform the IP of the targeted device.

Lastly, the **port number** can be adjusted to a free one. If a “custom” preset is loaded, we must most likely edit this parameter to establish a network connection.

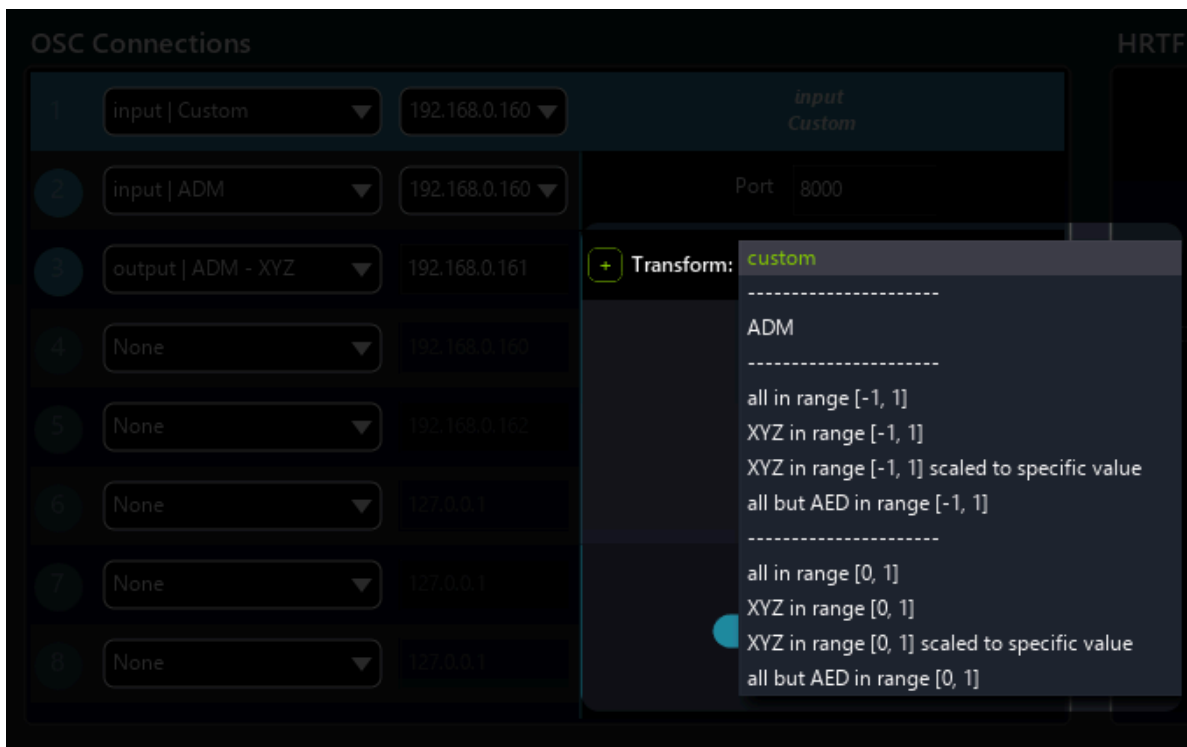
And finally, it is possible to **name the OSC slot**, clicking on the name of the top of the right panel.

If we want to deactivate an OSC socket without losing the parameters, it is possible to **deactivate an OSC socket**. To do this, click right on the OSC socket number and click on **Activate/Deactivate**, or shift-click on the OSC socket number. As it, the input or output will be deactivated.

If we loose the sync between the remote and SPAT, it is also possible to **dump all the project properties** on a particular socket. To do so, click right on the OSC socket number and click on **Dump project**. Note that the project is always dumped when the OSC socket is activated, and when the OSC IP address or port changes.

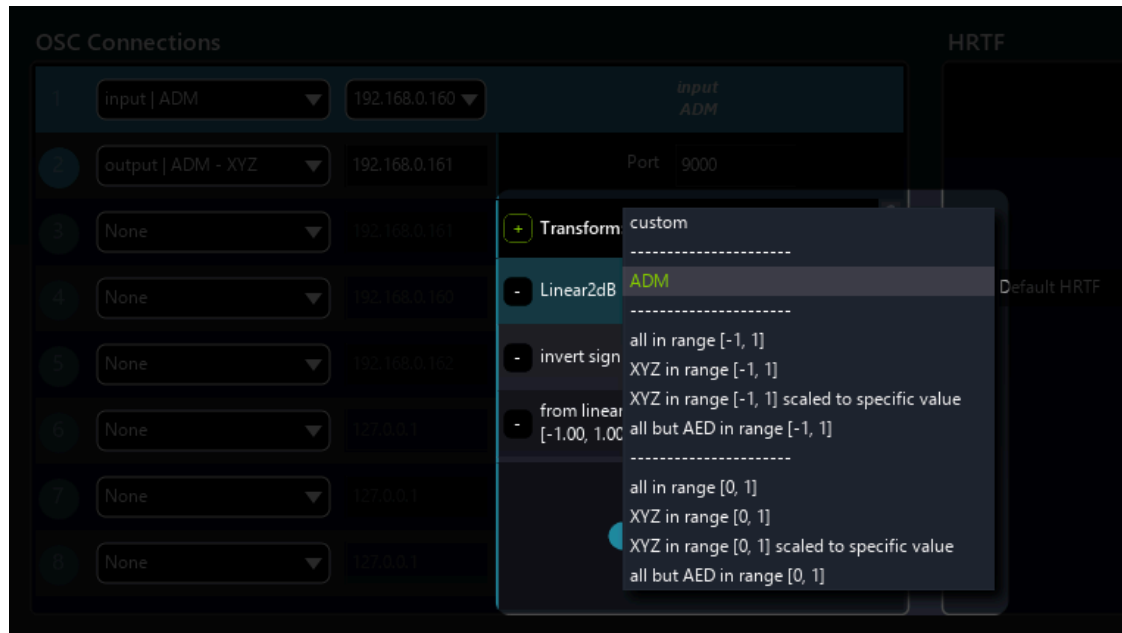
34.3 OSC Transform

Interfacing different devices and software in OSC can be problematic as each piece of equipment can have its own value scale. To overcome these difficulties, some OSC transform presets have been implemented.



Transform presets are accessible for each OSC connection and allow quick re-scaling of the values.

i Note



If the included transform preset does not fit your needs, you can click on the + button to open the custom OSC transform menu. You can choose how to scale your input or output values in this menu. You can also choose to exclude specific OSC command from the scaling rules.

35 SPAT PI : Automation

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.

A multichannel audio stream flowing in and out of SPAT may not be coming from hardware but 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.

35.1 DAW Automation - Manual setup

You must set up automation manually when you are not using the local audio path workflow and instead working directly with a connected hardware interface.

The first step is to attend the *SPAT Preferences* and *Enable OSC*.

You must then set up an OSC connection to connect with SPAT plug-ins via OSC. In one of the *SPAT Revolution* OSC connections, you can choose an input preset for a SPAT plug-in and do the same for the output preset configuration.

The port and host should match that of the SPAT plug-ins. By default, the plug-in ports are set according to the *SPAT Revolution* preset, requiring you simply to choose / set the IP address.

If you are running SPAT and the DAW software on the same machine, the IP address is always the so-called *LocalHost* (127.0.0.1).

Remember that the index number of each SPAT plug-in links it to a virtual source with the same remote number - or in the case of a SPAT Room plug-in - it is used to identify which Room you wish to control.

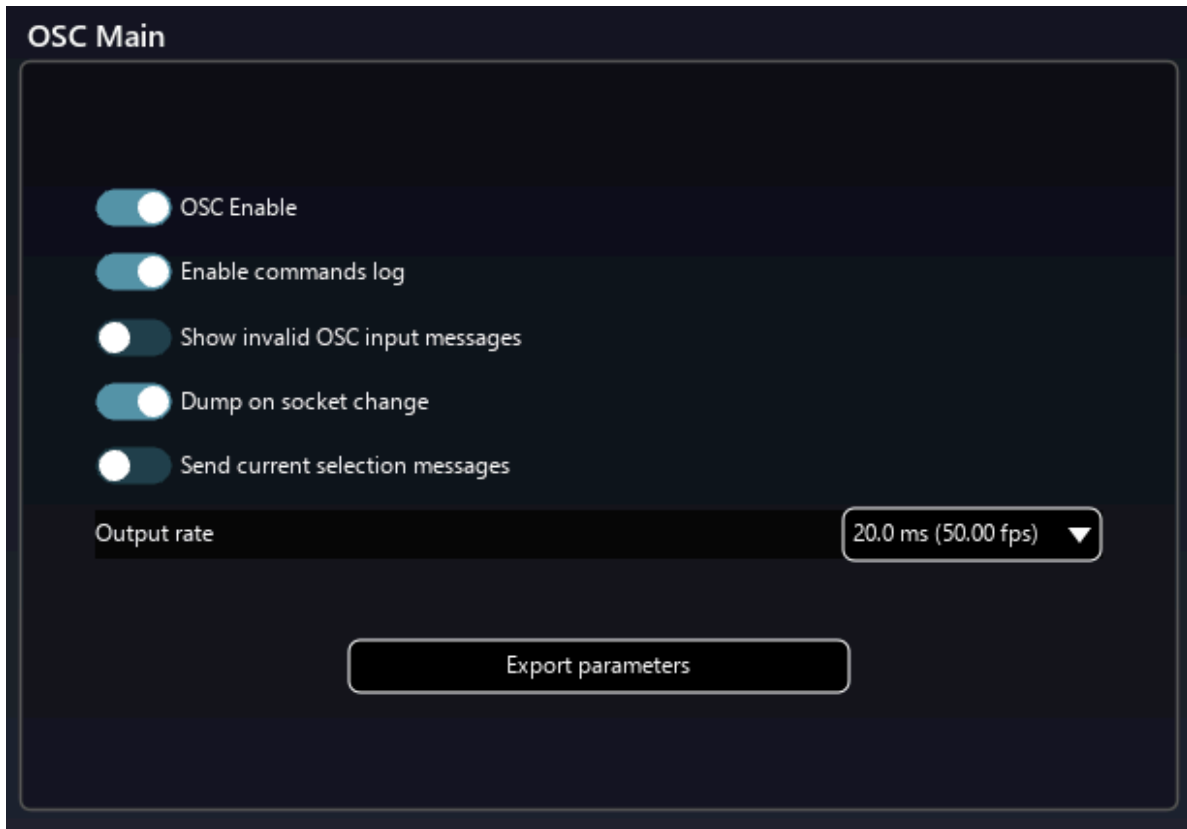


Figure 35.1: OSC Main

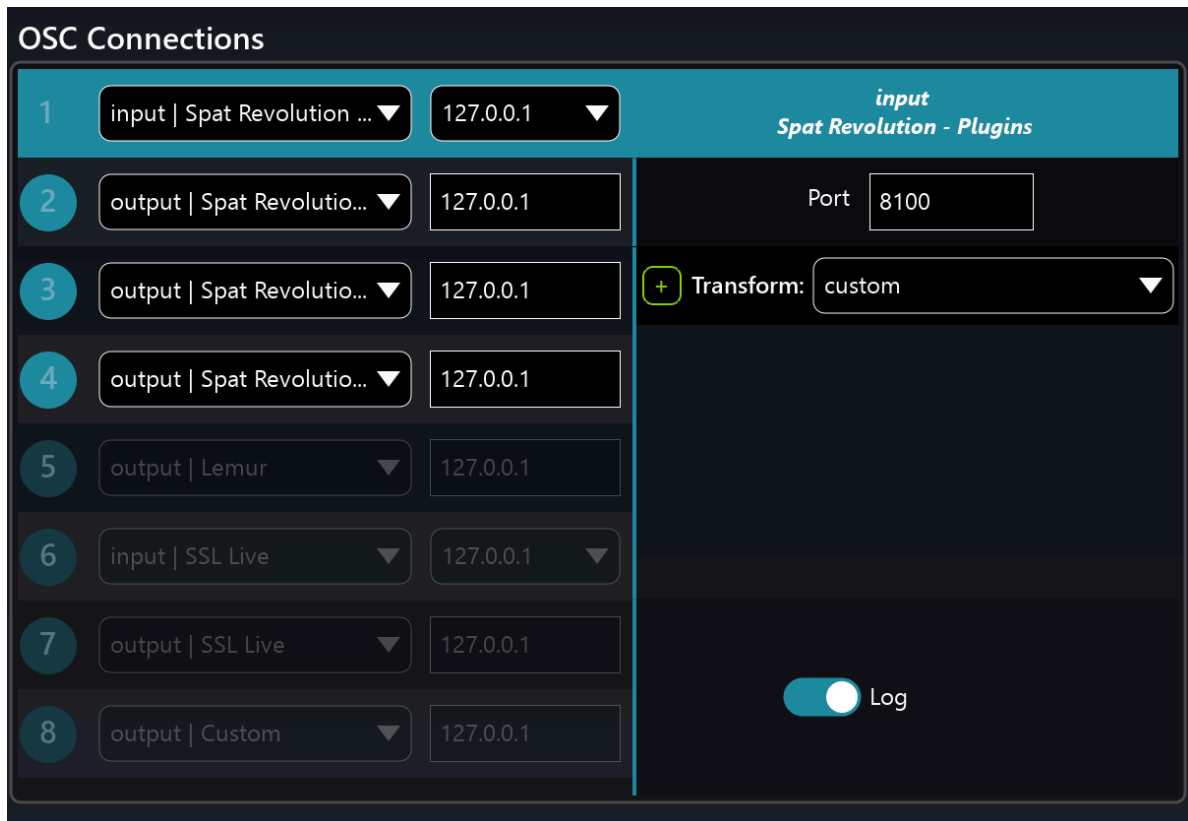
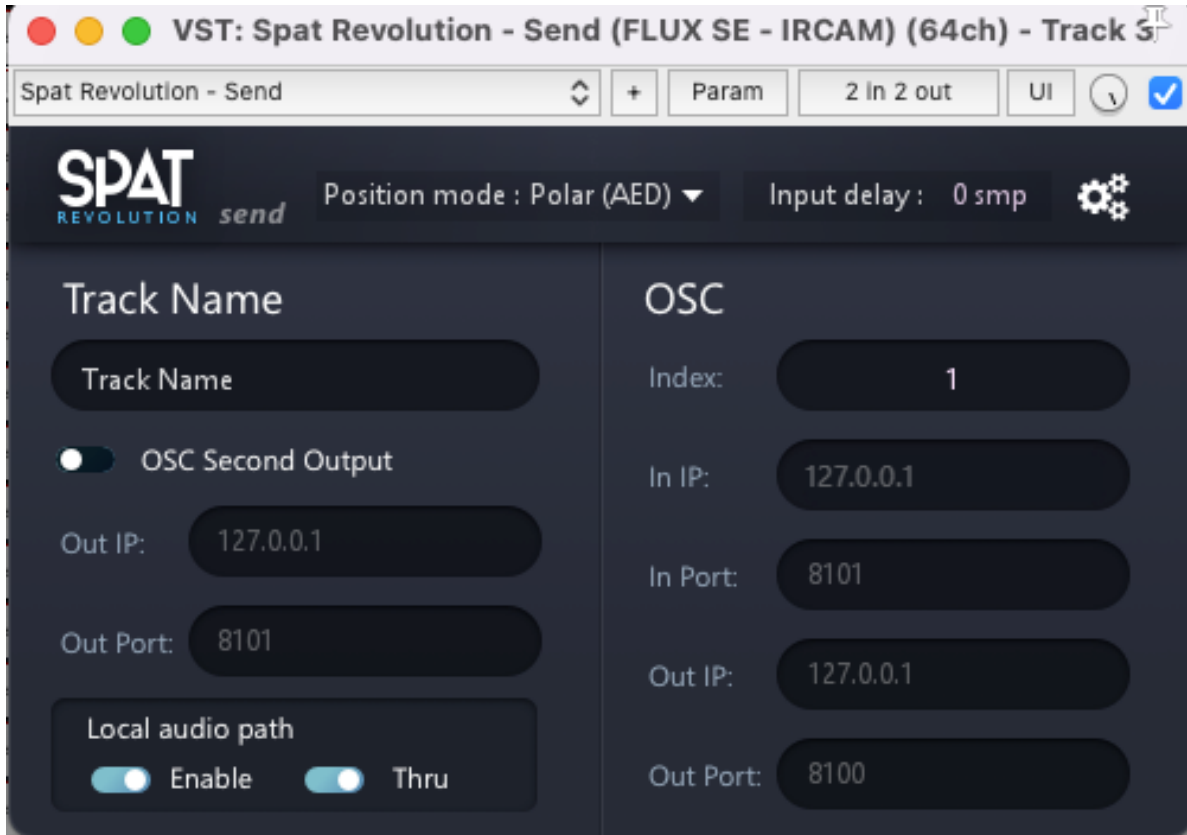


Figure 35.2: OSC Connection SPAT Plug-ins

Warning

Some systems require you to press **Tab** key and not **Return** to edit a field in the Plug-in.



Note

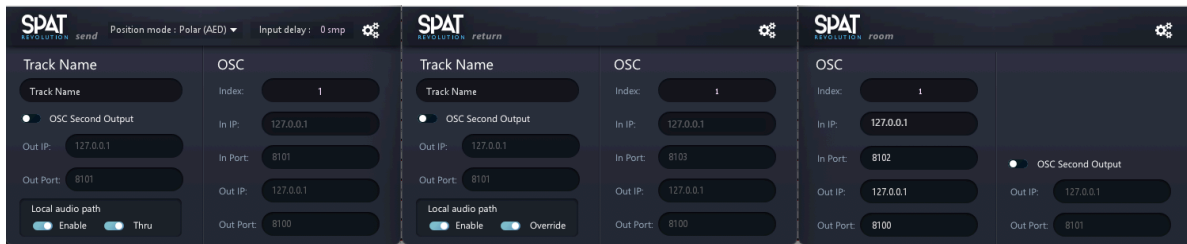
All send plug-in instances in one DAW will have the same IP and port number in the DAW, but different and unique Index numbers.

36 SPAT PI : Local audio path

Suppose you are deploying a setup for a single computer. In that case, you can use a software method to move multiple audio channels from the DAW running on the same machine as the *SPAT Revolution* spatialization environment. This is not a trivial task, and many previous solutions have been prone to drop-outs and other problems.

To answer this challenge, *FLUX:: immersive* has developed an audio pipe technology. Called “Local Audio Path” (LAP), it is available in the two (2) of the three (3) plug-ins available in AU, VST and AAX format.

The SPAT plug-ins offer a straightforward way to integrate the *SPAT Revolution* spatialization 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 modules
- Enable DAW automation of all Source-related parameters
- Index selects which Virtual Source to Automate

SPAT Return

- Return multichannel audio from a SPAT Return output module to a DAW track
- Enable DAW automation for Return parameters
- Index selects which Master to automate

SPAT Room

- Enable DAW automation of all Room-related parameters

- Index selects which Virtual Room to Automate

36.1 Setting up a local audio path connection

For the audio software integration to function correctly, the user needs to take into account certain principles of configurations.

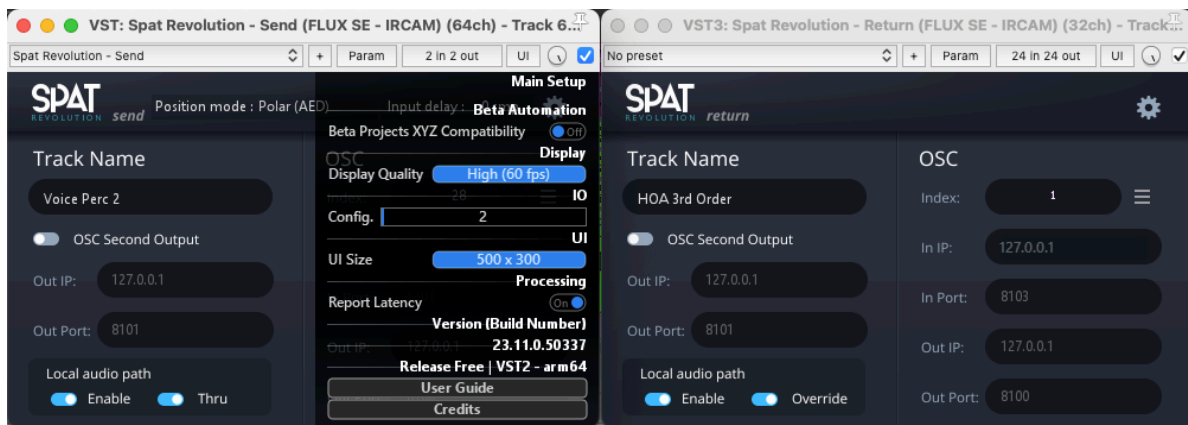
Warning

Sample Rate and buffer size must match in both *SPAT Revolution* and the Plug-in Host

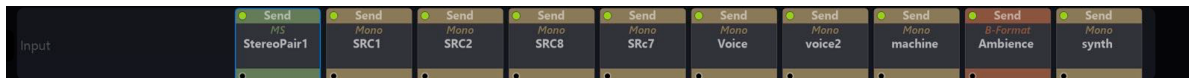
You can configure these settings in the *SPAT Revolution* Preferences, and matching settings must also be configured in the host DAW Preferences.

Note

If the sample rate and the buffer size don't match in *SPAT Revolution*, double-clicking over the "Sync section" at left bottom will automatically adjust them.



Additionally, each plug-in has an IO configuration setting, 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 Revolution*. Once you have selected the channel count, *Enable* the software routing using the *Local audio path* switch.



If *SPAT Revolution* is running, then a Send or Return IO module will automatically appear in the *Environment Setup* labeled 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 Return modules will have a small green indicator.

i Note

On some machines, you must use the **Tab** key to register a new Track Name or IP address change in a SPAT Plug in.

36.2 Plug-ins parameters

Index - Relates the plug-in automation to a virtual source.

- **Index** is assigned automatically and can only be changed manually to an index number not yet used by another SPAT Plug-in.
- On *SPAT Revolution*, Index is designed as **Remote number**. Initially set to the object number, you can customize it by object.

Position mode - *In Send only* - Choose the recorded and read coordinate mode.

Input delay - Use this text field to enter the latency induced by the previous plug-ins in the signal chain. For example, if you use a compressor with a look ahead above in the audio stream that delays the signal by 100 samples, you should enter 100 in the SPAT plug-in's **input delay** field. This value will delay every other incoming streams. In more complex cases, where many input latency is set up in different SPAT send instances, the lowest possible delay is applied.

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

Thru - In Send only, choose if you do not want to mute the sent audio through the plug-in.

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

Report Latency - Activate latency compensation reporting for the DAW.

Override - In Return only, override the DAW input path.

36.3 Implications of using the Local Audio Path

36.3.1 DAW routing priority

To make sure that no sync error can happen between *SPAT Revolution* and the DAW of your choice, it is obligatory to make sure that **each track containing a Spat Send plug-in have to be routed to every and each track containing a Spat Return plug-in.**

This routing forced digital audio workstations to process *SPAT Send* plug-ins **before** *SPAT Return* ones. Without this prioritization, you can end up in a situation where the DAW calls for SPAT samples that have never been sent. This led to sync errors, and most likely to audio drop. This is an absolute golden rule when working with *SPAT Revolution* and a DAW on the same computer.

 Warning

Always be sure that each track containing a SPAT Send plug-in is routed to every and each track containing a Spat Return plug-in.

For more DAW specific information, please consult our third-party integration section.

36.3.2 Local audio path and delay compensation

Taking the signal inside the FX insert blocks and sending it to SPAT Revolution has many implications on how we integrate with DAWs. For example, many DAWs compensate for processing latency after the entire processing chain (or even at the last possible summing point). This means that audio sent to SPAT Revolution with LAP is not latency compensated by default.

We have implemented a delay compensation algorithm directly in SPAT Revolution to work around this. To use it, we must manually enter the latency caused by the processing chain before each LAP plug-in. Each DAW has different ways of reporting this latency. Please consult our third party section to find out how the DAW you are using can display it. If your DAW does not appear here, please consult its user manual.

36.3.3 Dealing with DAW CPU optimizations

Some DAWs, like REAPER, use some technic to reduce the load of VST plug-ins on the CPU. One common trick is to process an audio track ahead of time and then delay the buffer to play it back at the intended moment. It is called anticipative processing. This is often very efficient and can drastically reduce the CPU load (up to 30-50% !)

Other DAWs, like Ableton live, put plug-ins that do not receive or send audio in an off-line state. Although it is not necessarily displayed to the user, it can create major issues when working with *SPAT Revolution* plug-ins.

 Warning

Be sure to consult our third-party integration section for more DAW specific information.

36.4 Alternative to local audio path

36.4.1 Pros and Cons of the Local Audio Path

The local audio path is designed to easily create audio connections between an audio application that can host plug-ins and SPAT Revolution, while also providing the automation data.

However, LAP may be limited by the DAW used in some cases. For example, both Logic and Ableton Live have an internal strategy to save CPU usage, which can cause problems with our SPAT send and return plug-ins.

Also, as most DAWs only allow the use of pre-fader and pre-mute plug-ins, this can either restrict the mixing workflow or make the routing very complex to allow post-fader send to SPAT Revolution.

When dealing with such issues, looking at virtual loopback devices is interesting.

36.4.2 Virtual Loopback Devices

Virtual loopback devices allow the audio output of one application to be routed to another. In this case, we use the direct output of our DAW console to feed the input of SPAT Revolution. If we want to record our mix, we have to send the output of SPAT Revolution back to our DAW.

Virtual loopback in MacOS

Within the MacOS environment, you must download a virtual driver such as [Blackhole](#) to send audio from one application to another.

For example, suppose you are using an application that supports multiple input and output devices (such as SPAT Revolution from Ableton Live). In that case, you can set your DAW to use an audio interface as input, set its output to Blackhole, and do the opposite in SPAT Revolution.

If your application does not support different I/O devices, you could try to use aggregate audio, but be aware that the latency may increase significantly.

Virtual loopback in Windows

Unfortunately, the Windows audio environment is less responsive to multi-client audio connections. So there aren't many solutions for software loopback within the Microsoft OS.

However, some audio interfaces, such as RME's, allow loopback of an output. This can be a good solution if your audio interface has enough audio outputs.

37 Snapshot

Snapshots are an important part of a live show. It can be used to capture an audio scene into *SPAT Revolution*. Sources, rooms and master properties will be stored in a snapshot. It allows a complete transformation of the soundscape, with interpolation.

Snapshots could be controlled on the snapshot page 17, the dedicated menu, and with OSC messages.

It is possible to exclude some sources from the snapshot recall:

- on the **Source Page**, disabling “Snapshot” on the wanted sources.
- on a **Room Page**, right-clicking on the wanted source and disabling the “Snapshot recall enable” option.

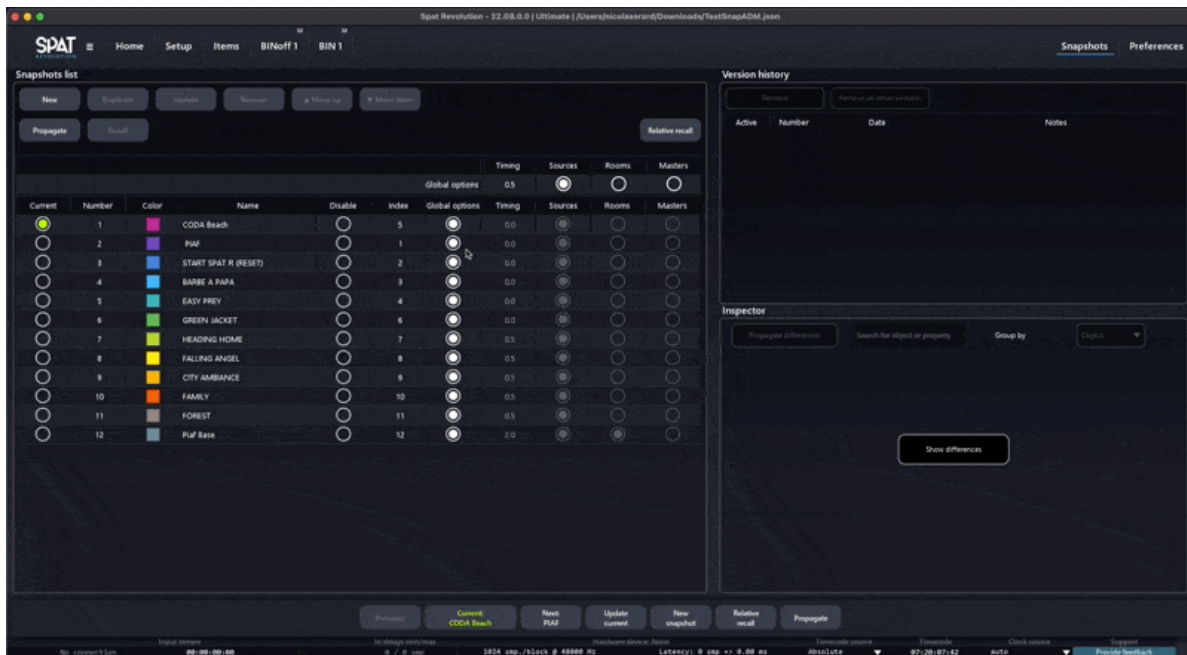


Figure 37.1: *SPAT Revolution* snapshot recall

37.1 Create a snapshot

You can create a snapshot: - by using the **Create** action into the “Snapshots” menu, or the **Create snapshot** available on the snapshot bar. In this case, the snapshot will be added, following the current snapshot (if there is any). The shortcut **Alt + Space** could also be used to capture a snapshot (currently not working on Windows).

- by using the “Insert Before” action. This option is available only if another snapshot exists. The snapshot will be inserted before the selected snapshot, and all the snapshots will be renumbered.
- by using the “Insert After” action. This option is available only if another snapshot exists. The snapshot will be inserted after the selected snapshot, and all the snapshots will be renumbered.

37.2 Recall a snapshot

Recalling a snapshot will interpolate all the current properties with the stored values of the snapshot. Different options located on the **Snapshots page** could alter the snapshot running:

- Recall relative

This option allows to keep differences between the current state and the current snapshot. More information about it on the snapshot page section [17](#).

- Global

This option enables to follow the global snapshot recall settings. It is also possible to setting up a particular one, disabling this option on the snapshot page.

- Recall time

This option is available in the menu “Snapshots/Options recall.” The values are ranged between 0 and 3600 seconds.

- Recall Sources / Room / Master

As all the properties are stored into snapshots, this option allows us to enable or disable the recall of some properties. The recall of each section could be separately activated.

Warning

Be careful with the Room properties recall, changing some properties like *Reverberation Density* or *Size* causes reverb reconstruction (and audio drops).

If **Ask for update before recalling** option is checked and changes are detected between the current snapshot and current state, a window will propose to update the current snapshot if there are differences between the current state and the snapshot.

37.3 Snapshots handling

Different actions could be executed with snapshots:

- updating a snapshot
- renaming a snapshot
- removing a snapshot
- remove all snapshots

38 Handle snapshots with OSC

Different snapshots actions could be launched by OSC messages. The complete list is available on the OSC Table [D](#).

i Note

Like sources and rooms, the index could be part of the message (`/snapshot/3/update`), or send as an argument (`/snapshot/update, 3`). All the example below will use it as part of the message.

If the index is part of the message, it can be replaced by “next”, “current” or “previous”: `/snapshot/next/recall` will then recall the next snapshot on the list using the default options set it.

- Create a snapshot: `/snapshot/create [*snapshot Name]`

The snapshot name could be added in argument (optional).

- Recall a snapshot: `/snapshot/[index]/recall [*time, *Recall Effective Selection, *Recall Actual Selection, *Enable sources recall, *Enable rooms recall, *Enable masters recall]`

Index: the snapshot index to recall. Used as an argument, the snapshot name can replace it.

Time: optional, it will define the recall time. If not given, the default value is 0s.

Recall Effective Selection: optional, if the value is **True**, the sources' selection on the snapshot creation will be recalled. It's the default value. If **False**, selection will not be recalled.

Recall Actual Selection: optional, if the value is **True**, only the parameters of selected sources' will be recalled. Else, all the sources will be recalled (default behavior).

Enable sources recall, Enable rooms recall, Enable masters recall: optional, enable to define if sources, rooms and masters parameters will be recalled. If not given, the default set value will be used.

- Update a snapshot: `/snapshot/[index]/update`

index: the snapshot index to update. The snapshot name can replace it.

- List all the snapshots: `/snapshot/list`

This will return the list of the snapshot, index and name.

- Len the number of snapshots: `/snapshot/len`

This will return the total number of snapshots.

- Rename the snapshot: `/snapshot/[index]/rename [name]`

index: the snapshot index to rename. *Name*: the new name of the snapshot.

- Remove a snapshot: `/snapshot/[index]/remove`

index: the snapshot index to remove.

 Warning

The Index can be replaced by the snapshot name.
Be careful: there isn't any confirmation.

- Change the recall options: `/snapshot/options/recall/sources [State], /snapshot/options/recall/[State], /snapshot/options/recall/masters [State]`.

State: 0 will disable the recall of related objects, 1 will enable it.

- Change the timing recall option: `/snapshot/options/recall/timing [Timing]`

Timing: Timing in seconds.

`/snapshot/options/recall [State] [State] [State]` takes the three options (sources, rooms and masters)

 Warning

Remove all the snapshots: `/snapshot/removeall`
Be careful: there isn't any confirmation.

39 ADM-OSC

ADM-OSC is an industry initiative developed by **FLUX:: Immersive**, L-Acoustics and Radio France (the later leading the case study and specific application) to facilitate the sharing of audio objects metadata between a live ecosystem and a broadcast or studio ecosystem. It tries to define a basic interoperability layer between object editors and object renderers in a live production workflow. It does so with **OSC** a communication protocol widely used in the live industry. At the base the ADM-OSC is a specific grammar and definition.

Immersive audio is gaining ground in different industries, from music streaming to gaming, live sound to broadcast. **ADM** or *Audio Definition Model*, is becoming a popular standard metadata model in some industries, with serial ADM used in broadcast or ADM xml files used in the studio.

A first implementation of ADM-OSC is now included with the latest release of *SPAT Revolution*. Other industry peers have implemented early versions of ADM-OSC such as L-Acoustics (L-ISA), Merging Technologies (Ovation) and Yamaha Steinberg Media (Nuendo). It is supported by default on OSC input and as an option on OSC output.

More information and specifications on the **ADM-OSC** initiative can be found on the GitHub repository [immersive-audio-live/ADM-OSC](https://github.com/immersive-audio-live/ADM-OSC)

39.1 ADM-OSC in SPAT Revolution

SPAT Revolution supports **ADM-OSC** in input as an alternate grammar, and on output as an option. The specification calls for normalized (linear) data value to provide interoperability and tend to align with the ADM protocol. Object-based mixing renderers will typically handle the scaling based on the system configuration.

Specifications calls for:

- Linear 0.00,1.00 for Gain and LFE (aux send) messages
- Invert Azimuth to clockwise
- Distance normalized to 0.00,1.00
- XYZ to a normalized cube -1.00,+1.00 (square in 2D, XY)

To configure ADM-OSC, ensure OSC is enabled and go to the OSC Connection section in the software preferences.



Figure 39.1: *SPAT Revolution* ADM OSC

- Choose an **input** | **ADM** preset to receive ADM-OSC data and select the receiving network interface.
- Choose an **output** | **ADM XYZ (or AED)** preset to send ADM-OSC data.
- Enter the IP address of the destination.

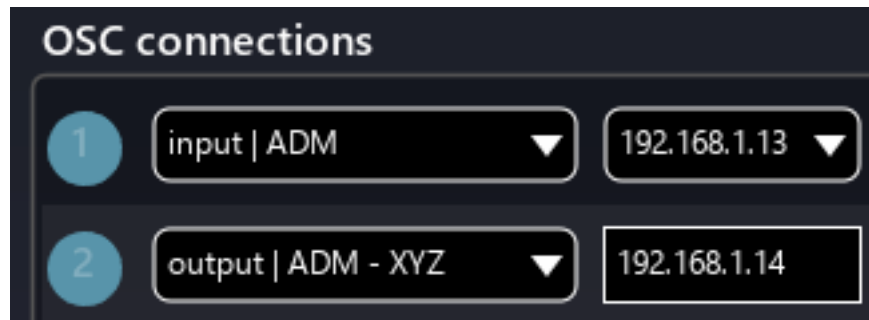


Figure 39.2: OSC I/O presets

On the OSC input connection, you can see that Port #9000 is our default incoming port and that and ADM Transformation preset is applied to match the specification. Enter your desired value to modify the incoming range (the automation zone). In this example, we are scaling to -3.00, 3.00.

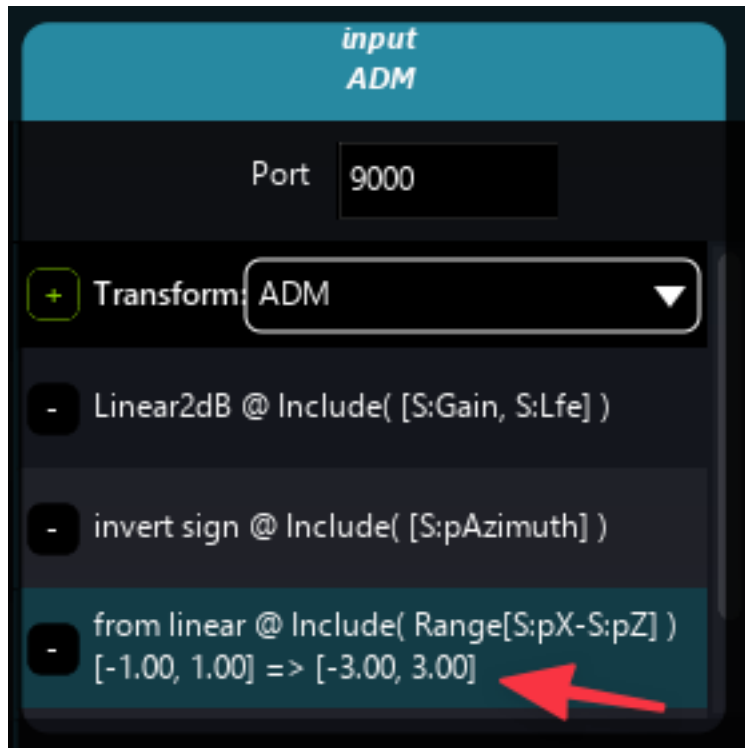


Figure 39.3: OSC I/O presets

On the OSC output connection, you can see that port #9001 is our default sending port and that an ADM Transformation preset is applied to match the specification. Enter your desired value to modify the incoming range (the automation zone). In this example, we send data from the -3.00, 3.00 zone. All output options are already set with the preset.

39.2 Third party example

Recently, L-Acoustics released their new L-ISA controller that can now output ADM-OSC as an alternate method (hardware required). For example, it is functional to be received by *SPAT Revolution*. OSC messages can be sent using the ADM-OSC format and be interpreted identically by any ADM-OSC compatible device.

Furthermore, Nuendo V11 adds the support of external OSC renderers, by mapping bidirectionally the multi-panner.

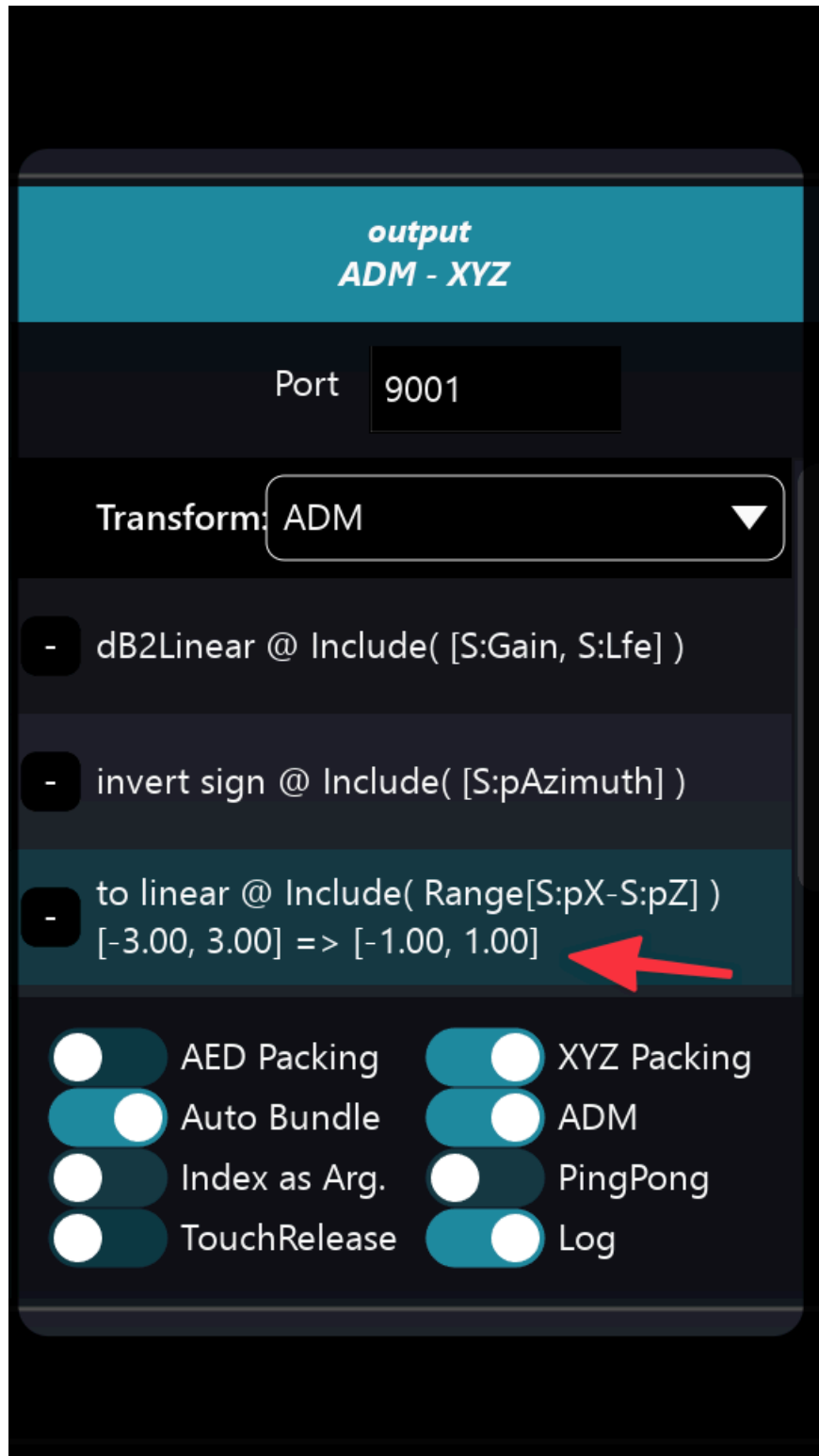


Figure 39.4: OSC I/O presets

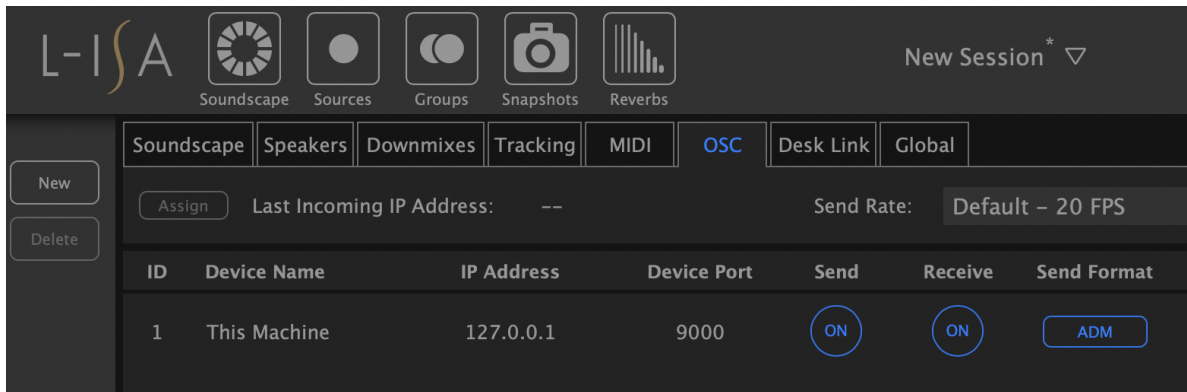


Figure 39.5: L-ISA



Figure 39.6: Nuendo

Part IX

Third Party Integration

This final section 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 that should generally apply 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.

40 Cockos Reaper

Reaper, a DAW developed by Cockos, is a reliable, fully-featured, 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 multichannel 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, has native OSC support for navigation, transport and more. It is one of the most complete flexible DAW regarding immersive sound production.

Reaper integrates well with the SPAT plug-ins and the Local Audio Path workflow described in section **SPAT PI : Local audio path 36**.

If you are using the Local Audio Path, it is also recommended reviewing the **Appendix C - Troubleshooting C** of this guide.

40.1 Templates

New recent templates are available for use with Reaper. They are start sessions and can be used as examples to see how to integrate *SPAT Revolution* using the SPAT plugin suite SEND, RETURN and ROOM.

A tutorial on [How to set up SPAT Revolution with Reaper](#) is available here for a quick dive into this integration.

You can download the following session templates:

Project Sessions and Templates downloads.

- **Reaper Tutorial Template** is the template used in the above video tutorial and includes send and return setup for rendering Binaural, 5.1 and NHK 22.2 output formats.
- **Basic music REAPER** is a basic template using send and return to render Stereo, 5.1, Atmos 5.1.4 output formats with Binaural monitoring.
- **Advanced multi-format REAPER** is an advance template using send and return to render Atmos 7.1.2, NHK 22.2 and multiple binaural output.
- **Ambisonic HOA mixing REAPER** is a template for using 3rd order HOA and binaural monitoring output formats.

40.2 ReaVolution

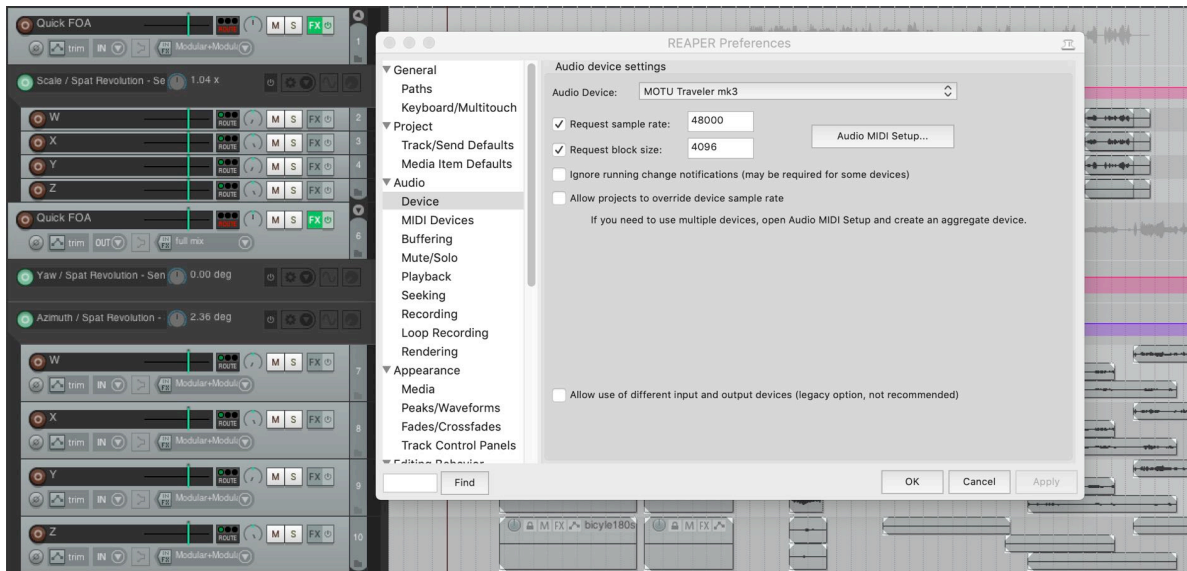
Release in 20.12 is ReaVolution. ReaVolution is an attempt to customize Reaper standard configuration to simplify its integration to *SPAT Revolution* and provide a complete simple immersive audio creation package.

The ReaVolution package includes macros (scripts), customized toolbars, system behaviour preference enhancements and makes the integration of *SPAT Revolution* to Reaper a charm.

Please follow this link for more information on ReaVolution [41](#).

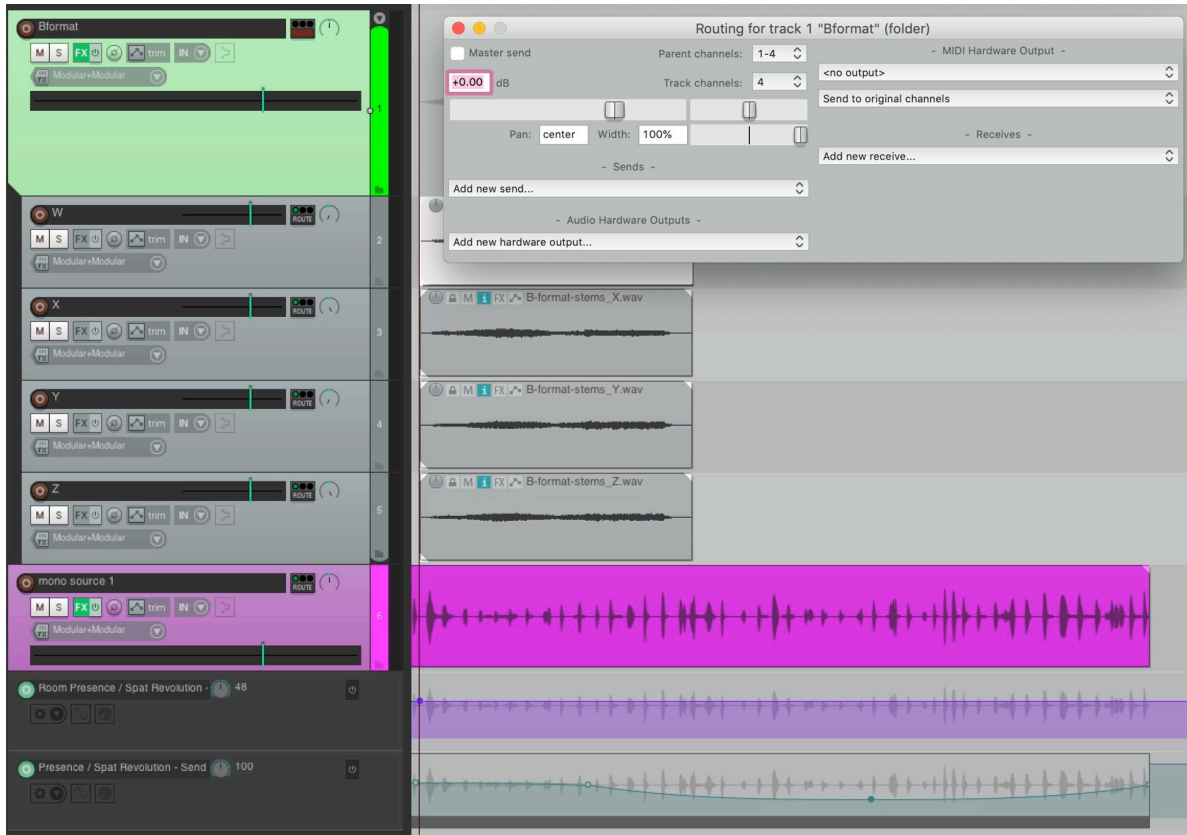
40.3 Setting Up Sync

As described in section **SPAT PI : Local audio path 36**, when 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.



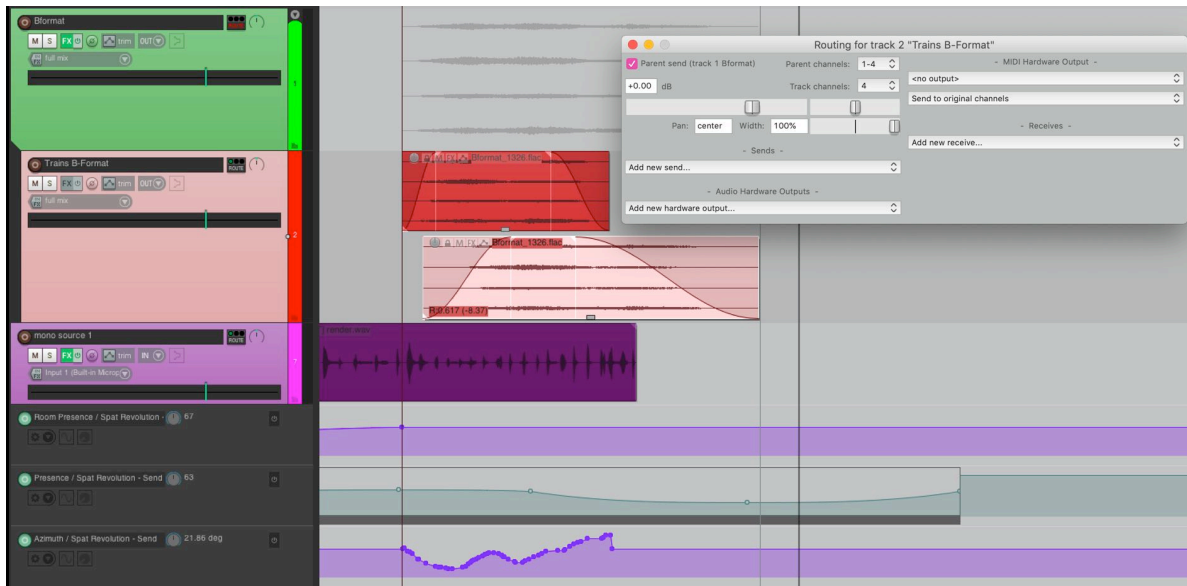
40.4 Setting Up Tracks

Working with Track Folder structure for your organisation is a good idea.



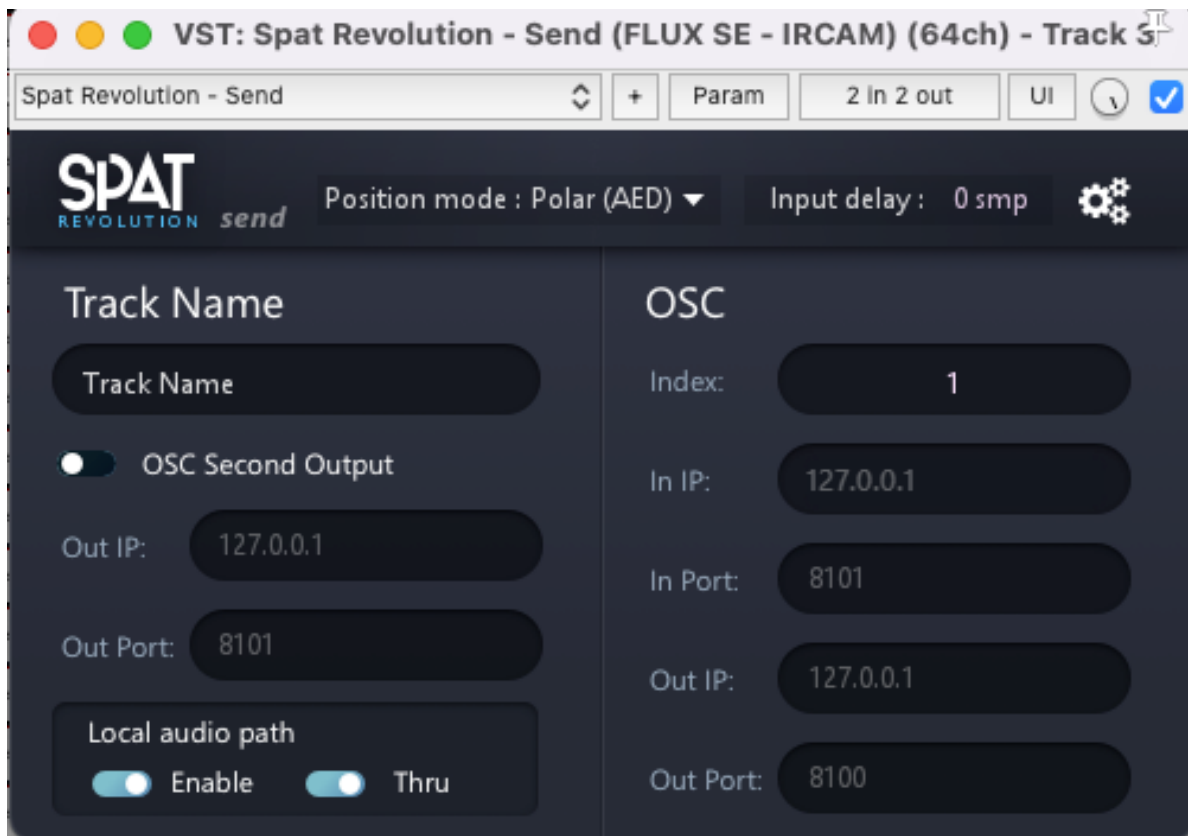
In the above screenshot, the B-Format Master has been set to be a Folder Parent with 4-Track Channel. Reaper channel routing is set on a track by track basis, using the *TrackIO Route* button of each Track. All the Child tracks routed to the parent can be assigned to one of the four receiving channels 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 4-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 the same virtual source/object in SPAT. All automation for that source should happen in the Envelope Lanes of the Parent Folder track.

40.5 Setting up SPAT SEND



Simply insert the SPAT SEND plug-in on a Parent Track. This means that as the composition grows, multiple Child tracks may send different audio material to the same virtual source/object 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 the automation for the spatialization is managed in one place even though source material may be different at different stages of the composition. It may not always be necessary to have a different SPAT SEND source/objects for every audio file. The source automation can change the envelope lanes of the Parent Folder, so child tracks share one SPAT SEND 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 preference 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.

In the main page, enable **Local Audio Path**. If all is well, you will see the SEND appear as an input at the top of the setup environment in your *SPAT Revolution* application. If you do

not see it, you can try clicking on *Get send/return* on the top actions or in *Setup/Pipes/Get send return* to force SPAT to look for virtual IO.



In the above screenshot, the Orange 'BFormat' Send input is the one being routed by the SEND plug on the Parent Track on the previous example. At first, it will appear in SPAT as a Channel Based input by default - in this case it appeared as a 4.0 QUAD. If dealing with ambisonic 1st order such as AmbiX or B-Format, the input stream type in SPAT needs to be assigned manually.

40.6 Accessing the latency of the FX chain

To access the latency information in REAPER, open the FX windows by clicking on the FX button.

In this window, REAPER will show you two latency values. The latency of the currently selected plug-in and the actual latency added to the signal chain. REAPER always rounds up to a multiple of blocks. You should report the sum of all the plug-in latency in the SPAT Plugin (ergo, not the latency actually added by REAPER).

Warning

If you use “post-fader” sends to SPAT Revolution, you don’t need to worry about this step.

Since REAPER 6.20, the latency is compensated per FX chain. One solution is to revert to the legacy behavior, which compensates per FX. To do so, go to the option of the FX chain, then to “Chain PDC mode” and select “Per FX compensation”. You must do this for each track hosting a SPAT send plugin and prior FX that add latency to the signal chain.

40.7 Optimization

There are a couple of things to make sure to optimize the integration and avoid inconsistent behaviour:

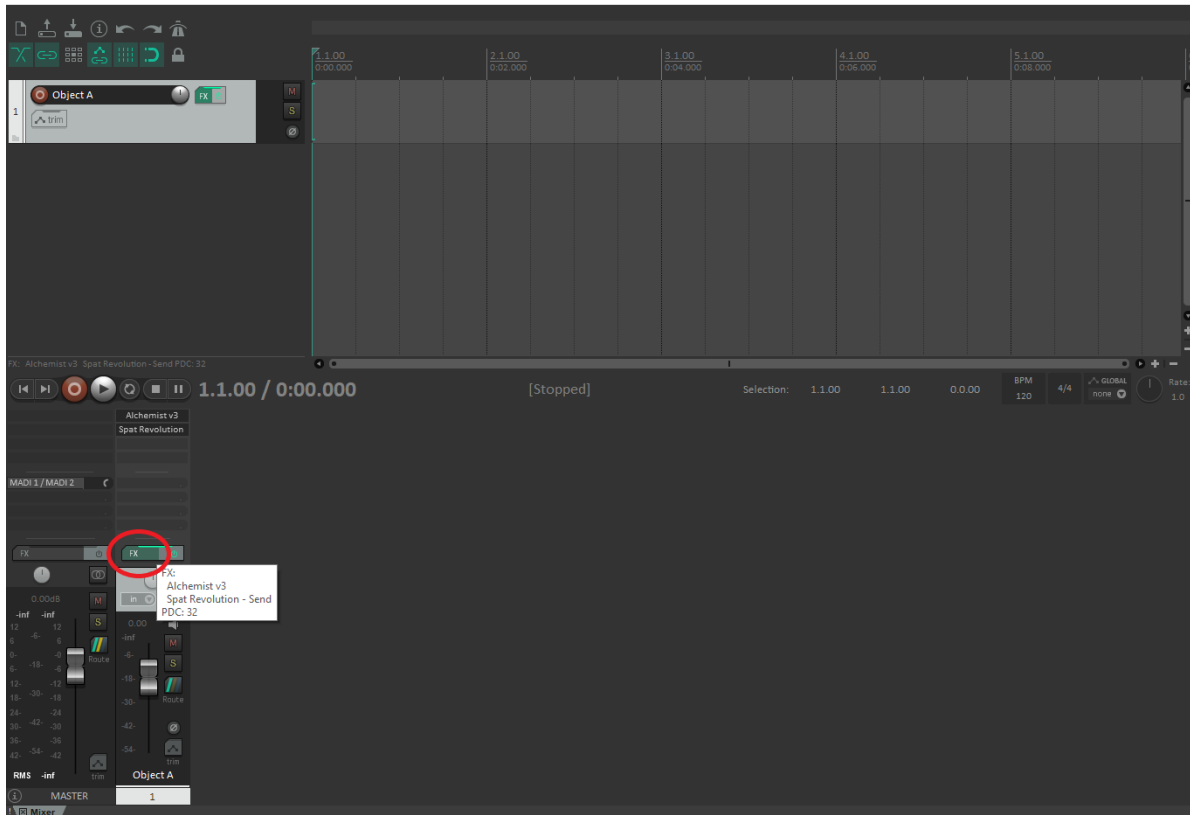


Figure 40.1: FX chain button

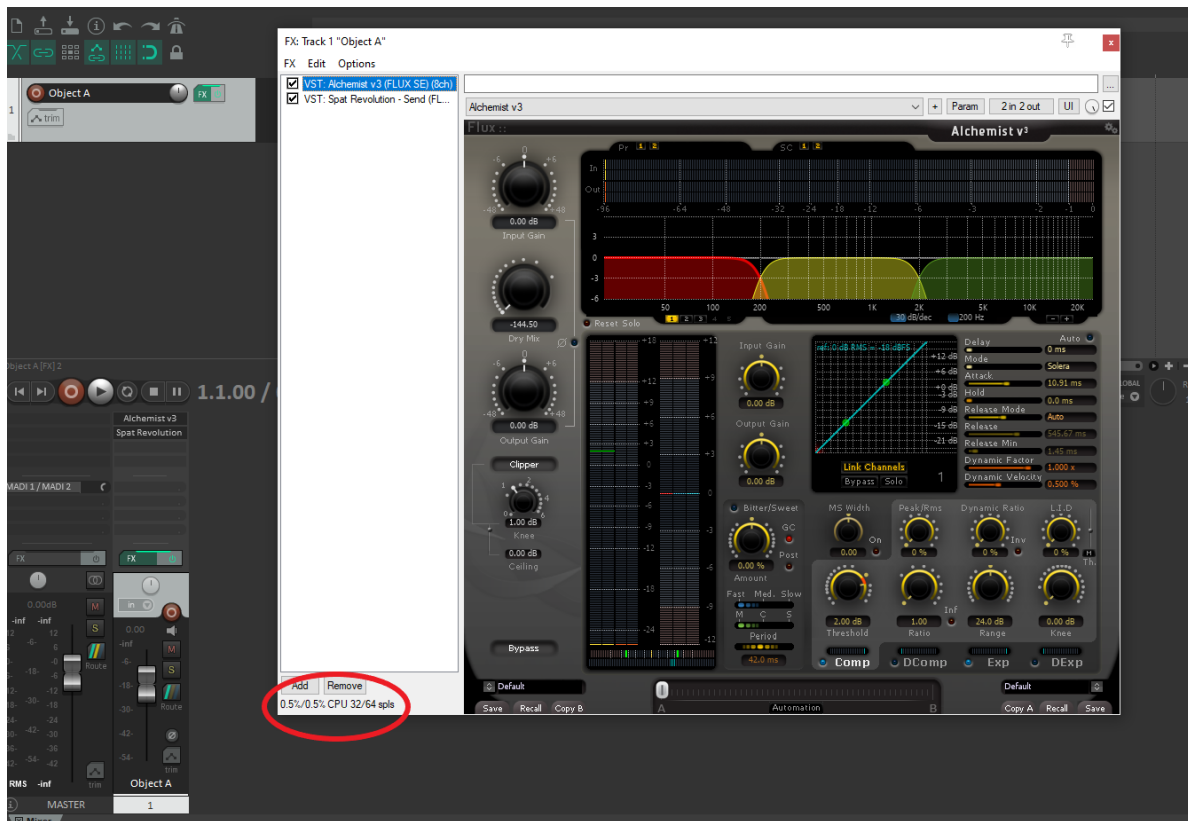


Figure 40.2: FX delay info

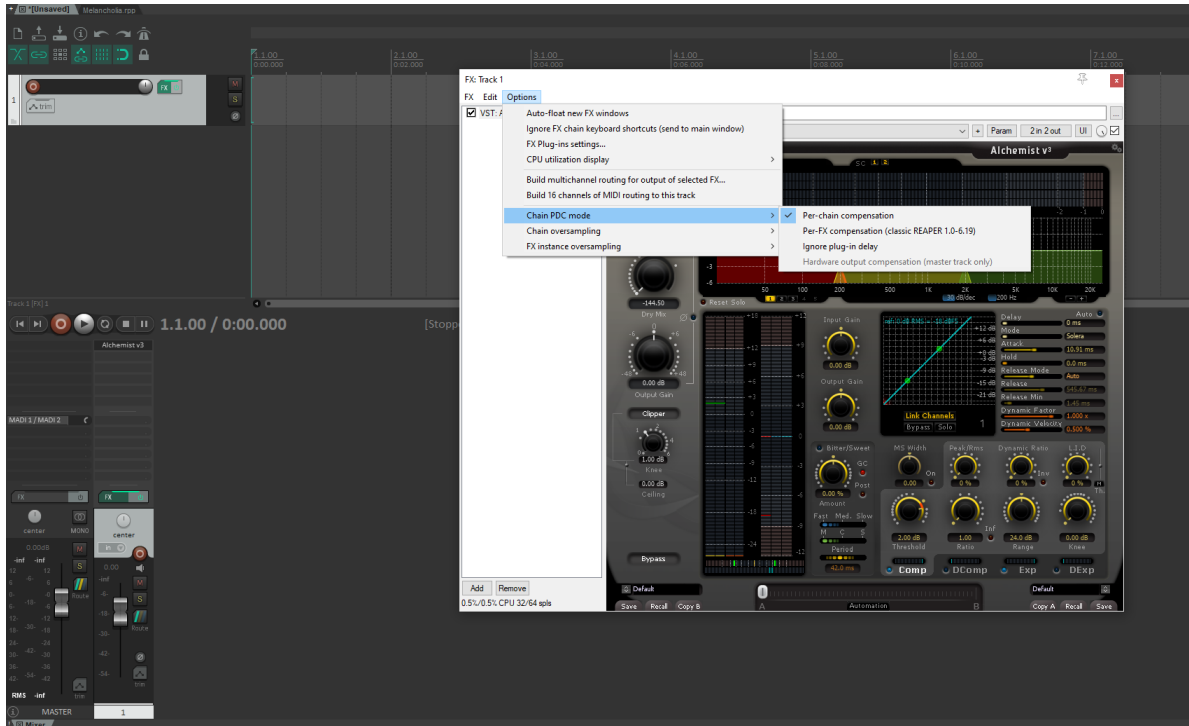
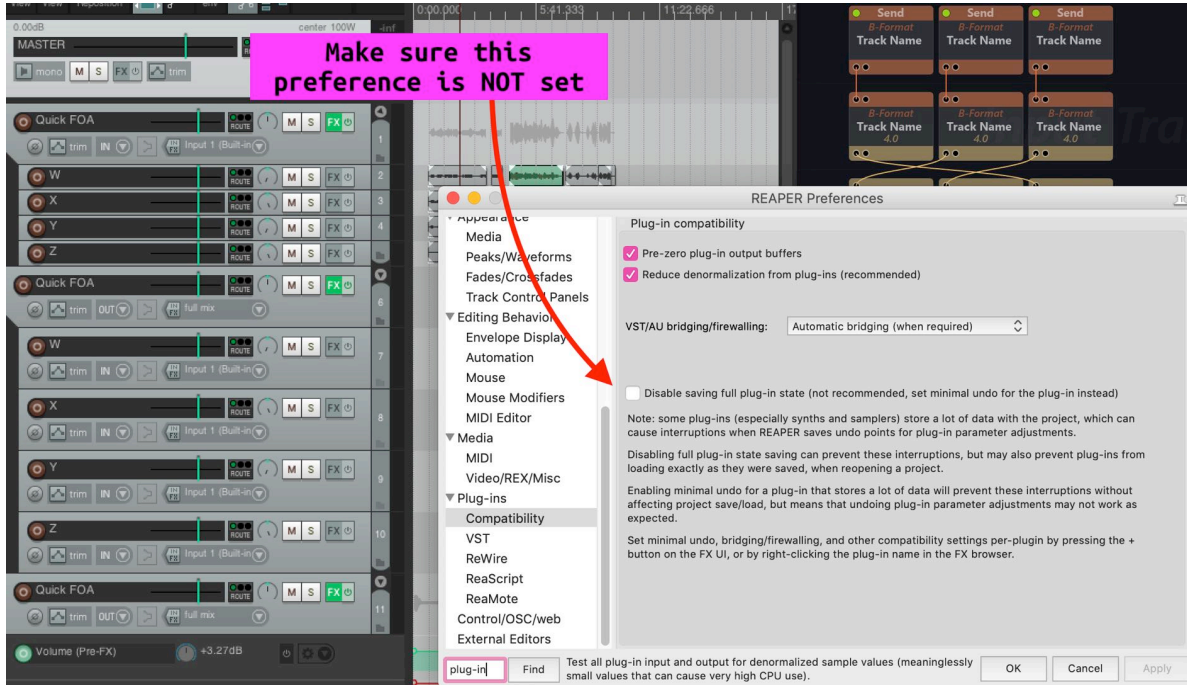


Figure 40.3: FX chain PDC mode

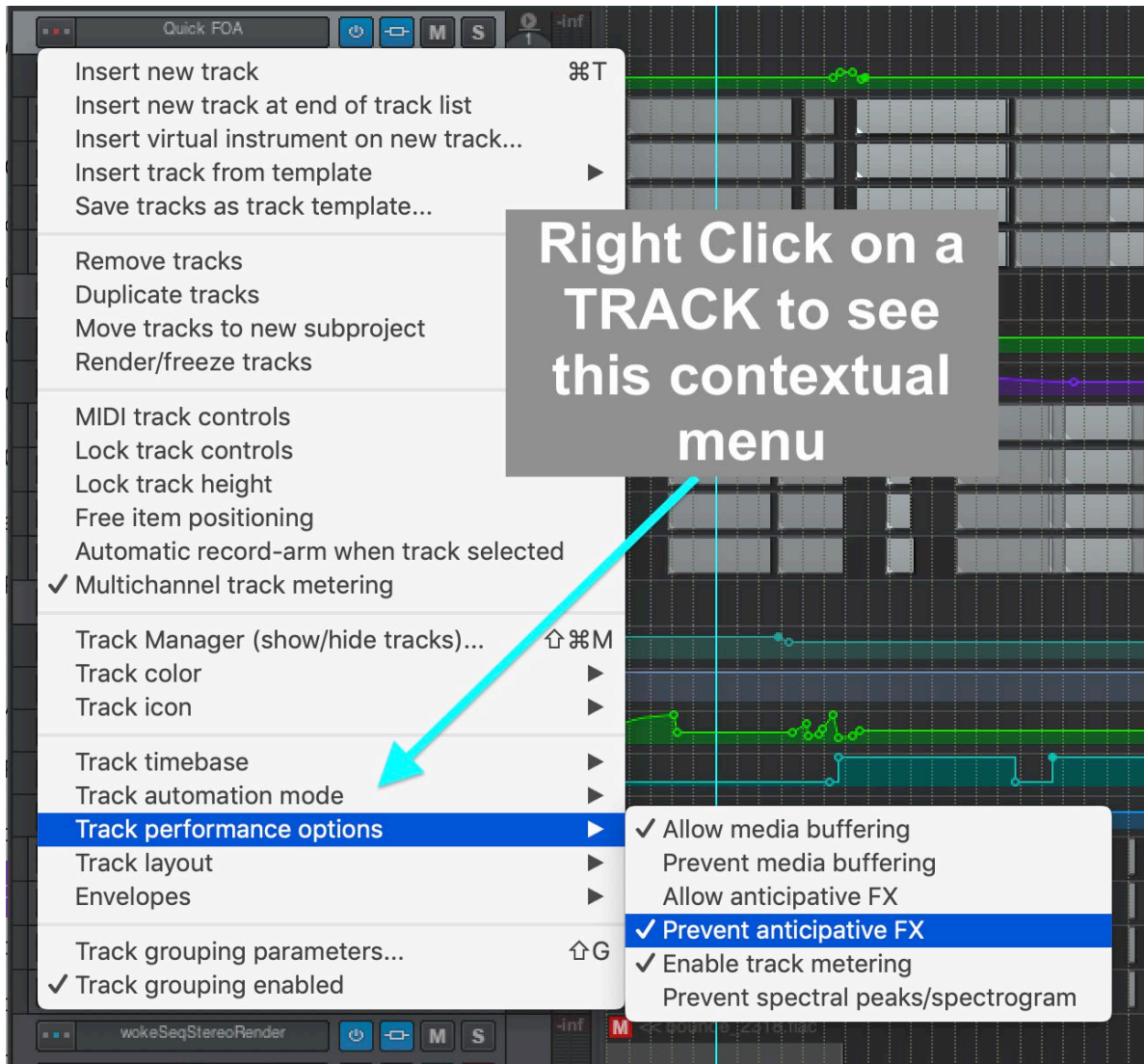
Warning

Disable **Full Plug-in State** set by default in Plug-in Compatibility Preferences.

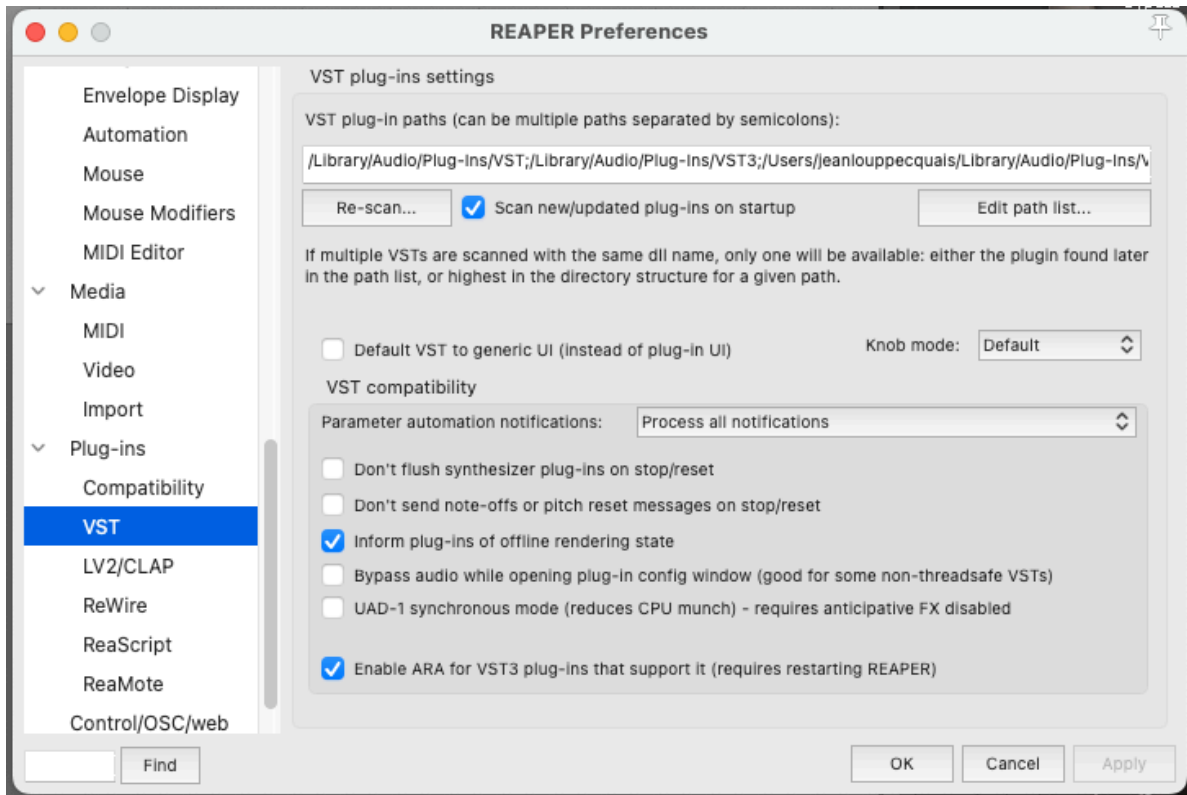


Warning

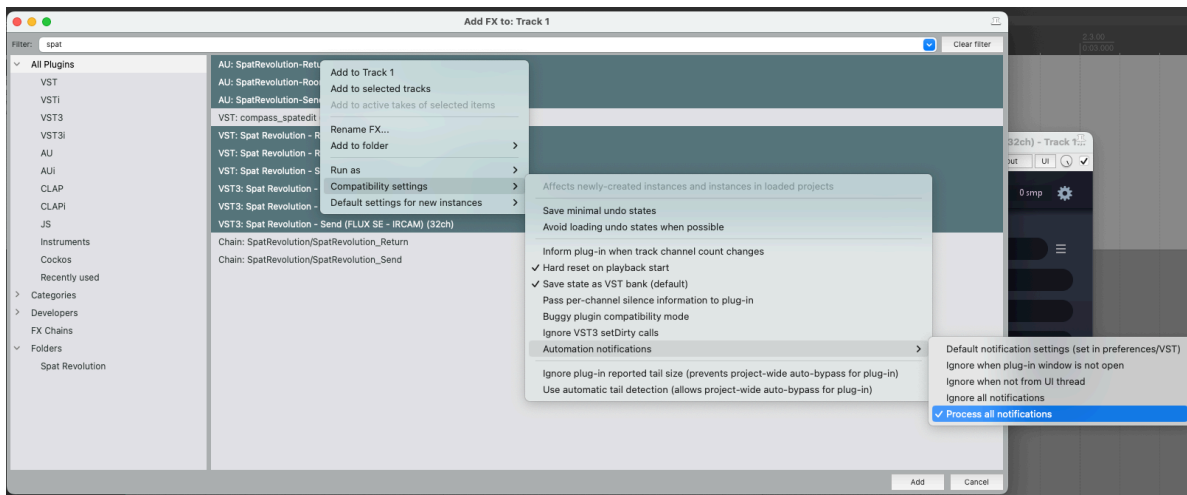
“Prevent Anticipative FX” should be enabled on tracks where SEND / RETURN are inserted.



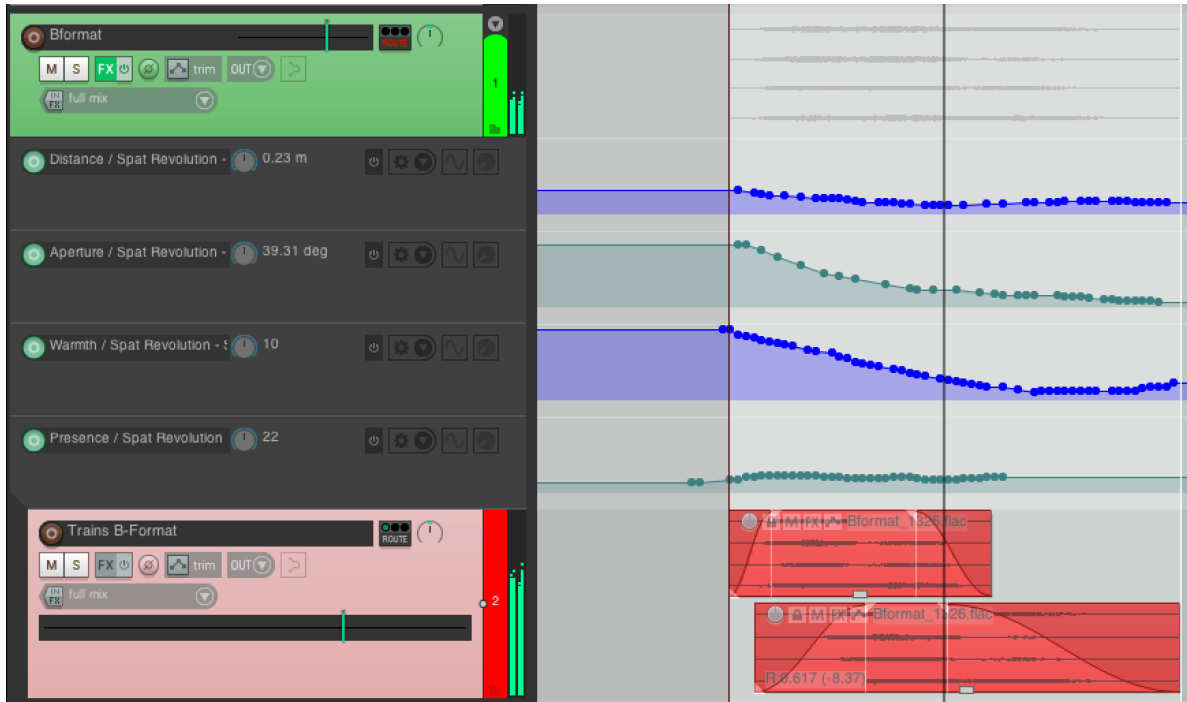
Also, by default, REAPER does not process automation messages when the plug-in GUI is not open. You can change this behaviour globally by going to the Preferences menu, in Plug-ins, VST. In the “Parameter notification automation” line, select “Process all notifications”.



If you don't want this behavior to be global, you can enable it for specific plug-ins. To do this, open the FX Browser, search for SPAT plug-ins and select all three. Then right click and go to "Compatibility settings", "Automation Notifications" and select "Process all notifications".



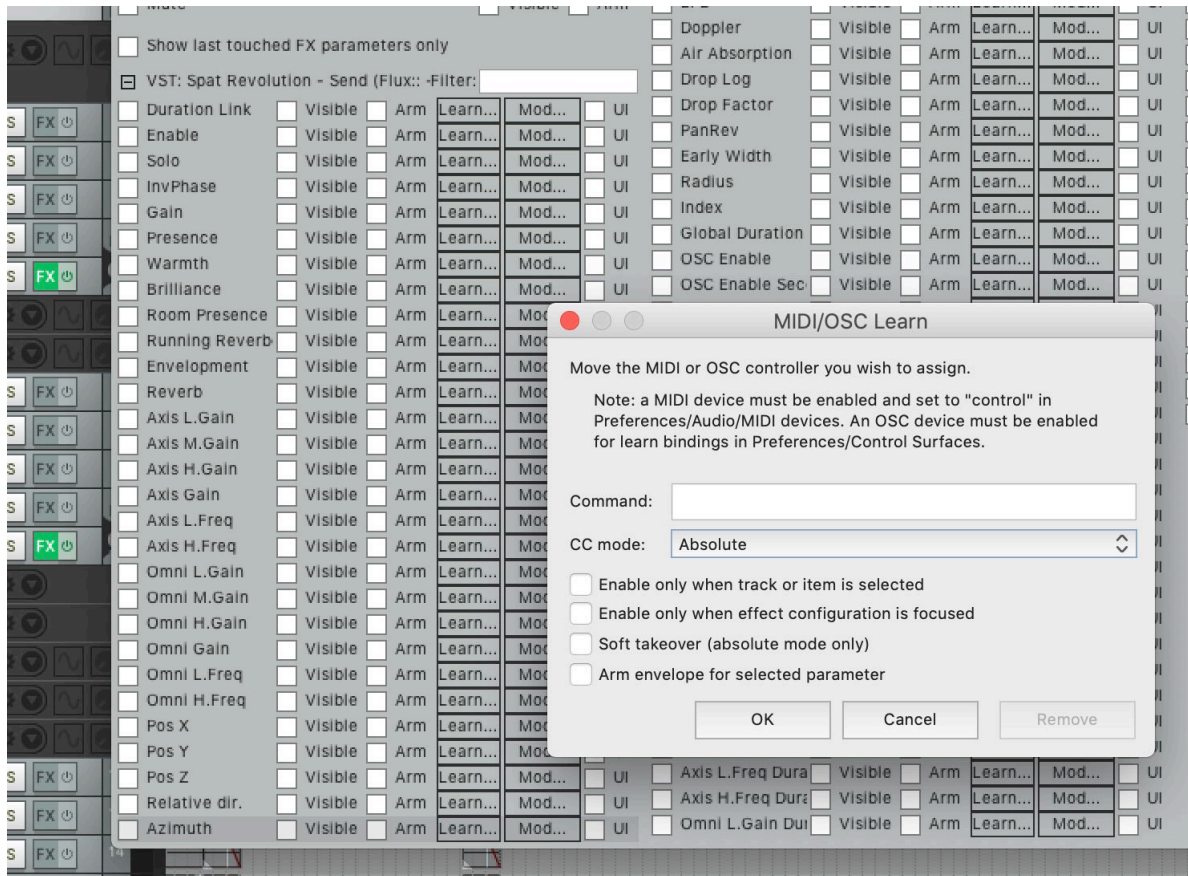
40.8 Source/Object Automation



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*.

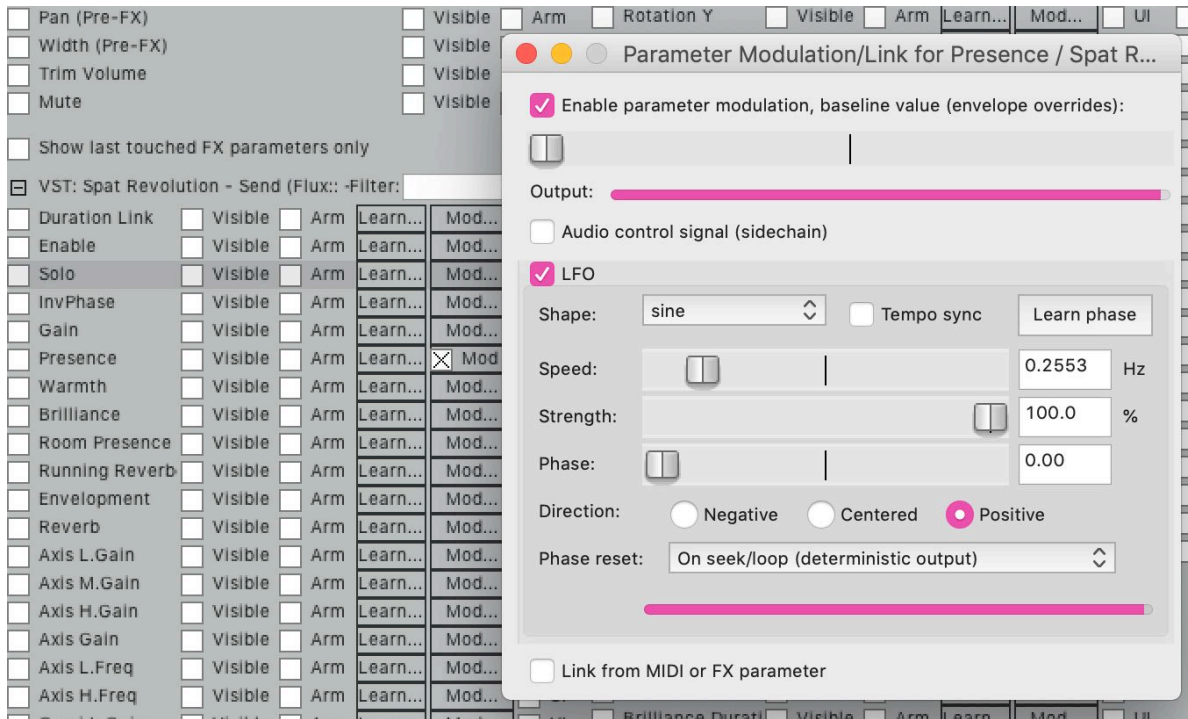
40.9 Setting up controllers and LFO

In Reaper, mapping any MIDI controllers (including 14-bit) to a Plug-In parameter is easy. This is a great way to control source properties like *Azimuth* or *Yaw* using external MIDI controllers, so you can control some sources live by hand during a performance or while mixing for example.



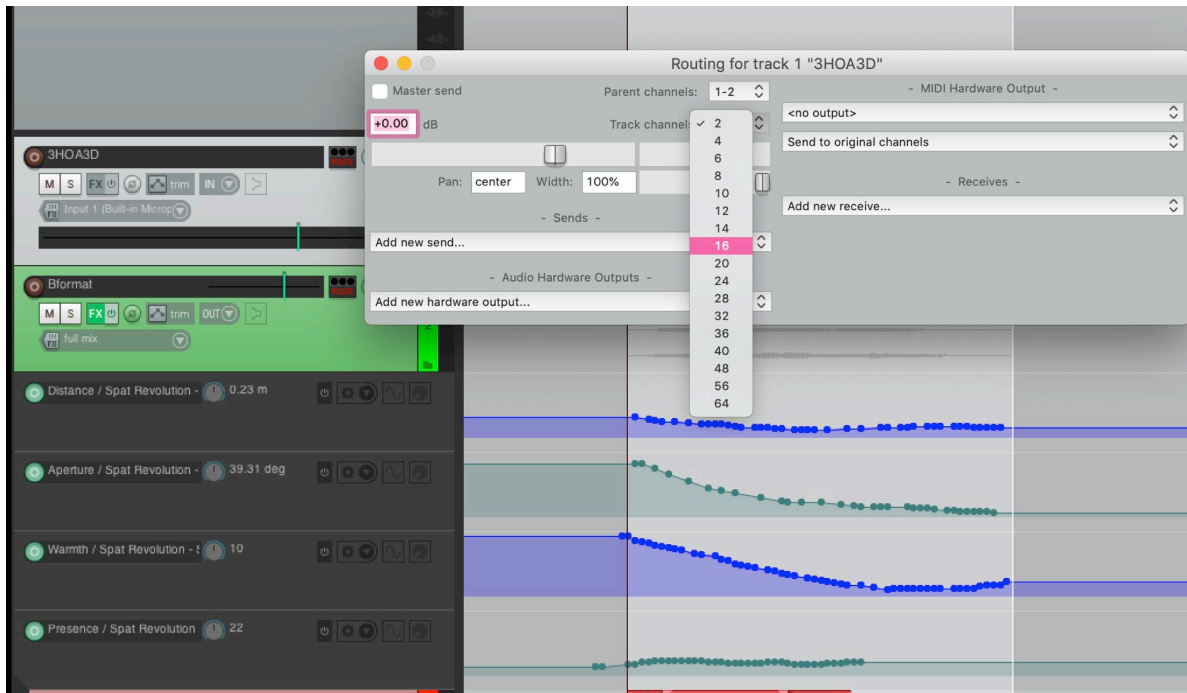
In the Automation parameter list for SPAT SEND just click on the **LEARN...** button to manually assign an external controller that you have set up in the Reaper Controller preferences.

From this page by clicking on the **MOD...** button of an automatable parameter, it is also 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.

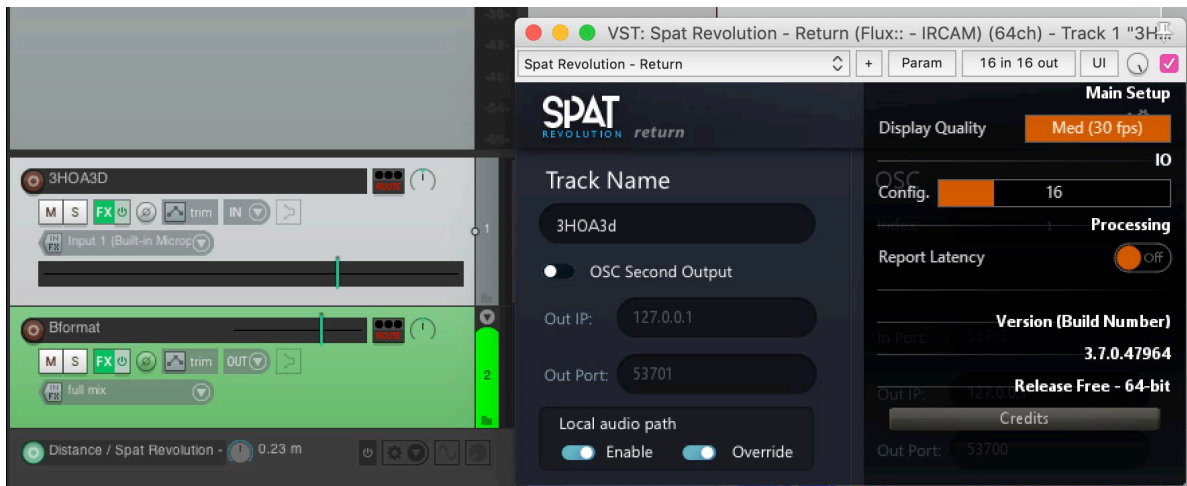


40.10 Setting up SPAT RETURN

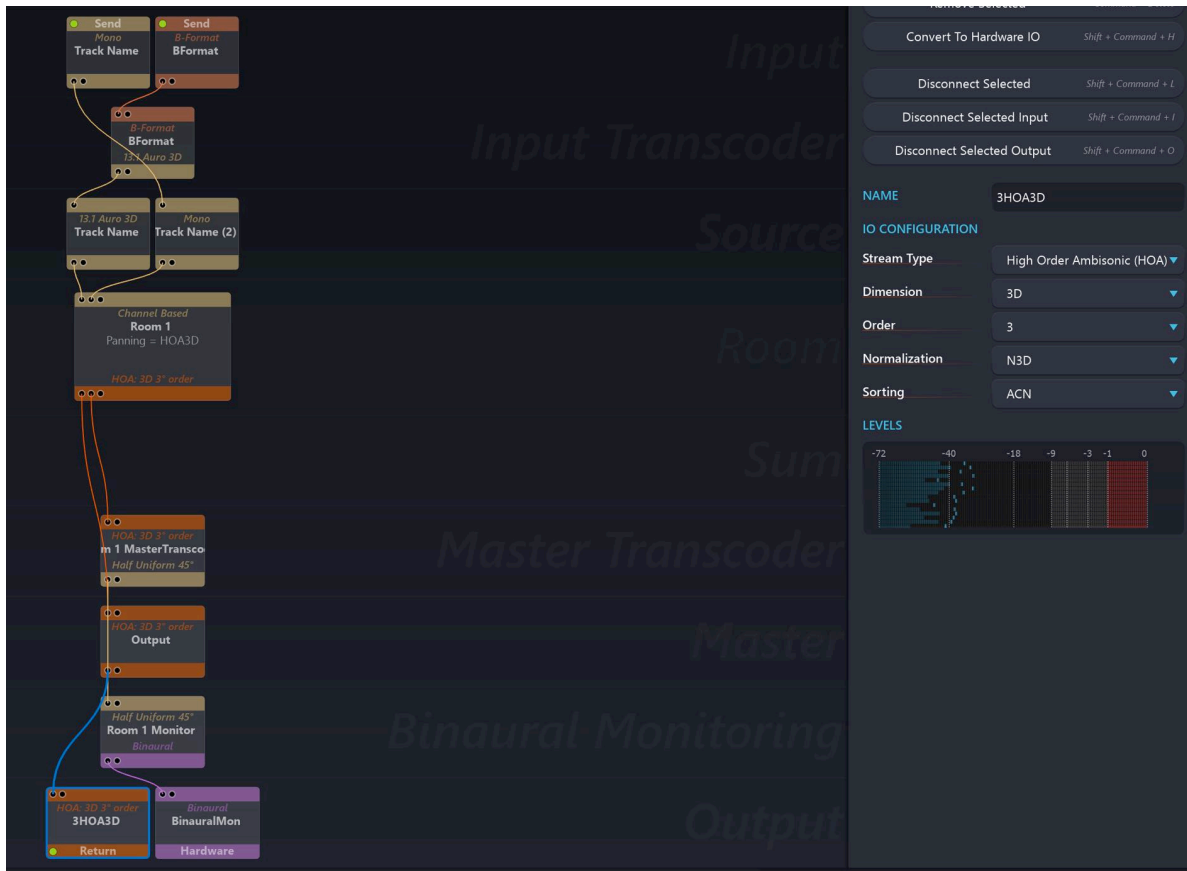
Now, to render the scene from the multichannel room output to disk as an interleaved 3rd Order HOA, for example, we need to add a RETURN plug-in. We can calculate that 3HOA3D needs 16 channels.



First, set up a Track in Reaper that can handle 16 channels simultaneously. Then add a SPAT RETURN plug-in on this 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.



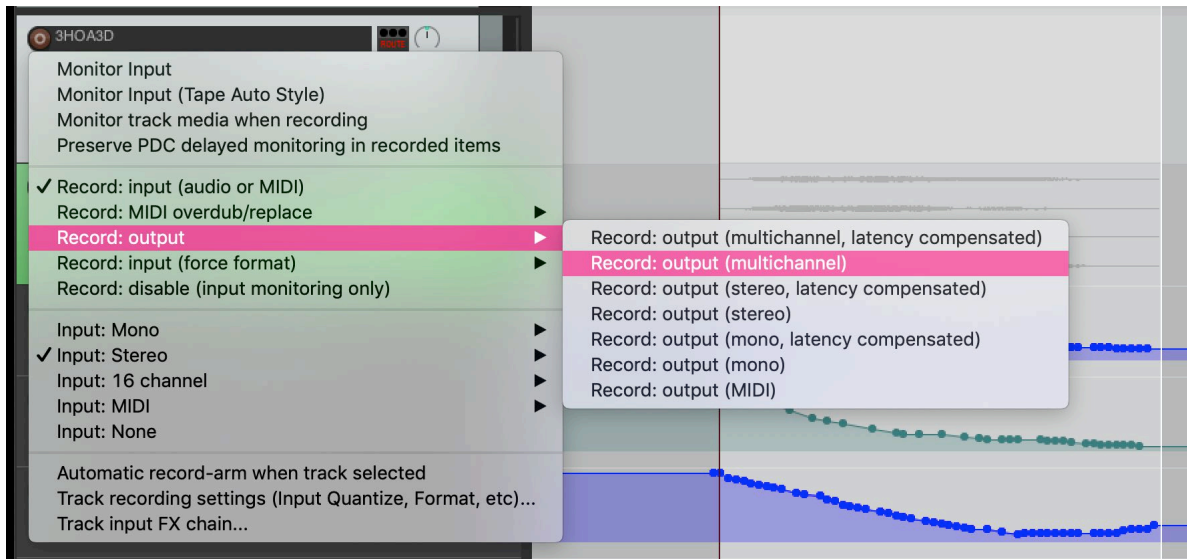
Enable the Local Audio Path: a Return output module should appear in the SPAT Setup Environment. Connecting it to the 3HOA3D stream output from the Room (or Rooms in a mixer/transcoder) and it should inherit the format.



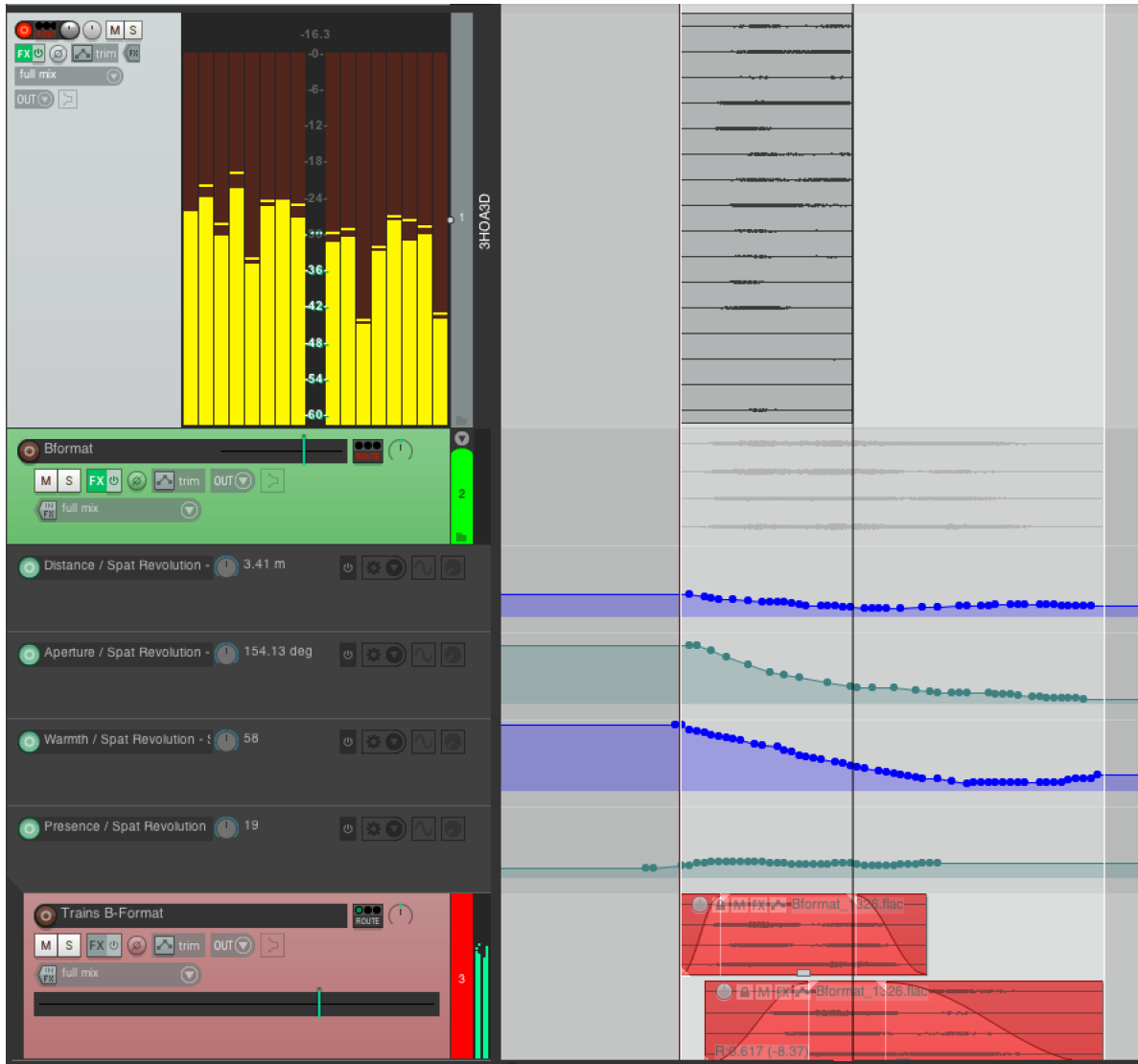
40.11 Recording Immersive creation.

To do this, 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 track's record arm button, 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). This is done in realtime.



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

41 ReaVolution

41.1 Credits & acknowledgements

ReaVolution was developed by Jean-Loup Pecquais, between Orléans and Paris, from 2019 to 2020.

The author would like to acknowledge a few peoples:

- Gael Martinet, FLUX:: Immersive CEO, to support and believe in this project
- Hugo Larin, *SPAT Revolution* product manager & business Development at FLUX:: Immersive, for his help and interest
- Nicolas Erard, to have tested almost every corner of ReaVolution and cramped my todo list with too many bugs report
- Sylvain Lambinet, which has followed this idea from the very start and always inputted some great feedback
- Justin Frankel and the whole Cockos team to have put on the market such an incredible tool.

ReaVolution is built upon the following Reaper extensions:

- SWS extension : <https://www.sws-extension.org/>
- JS API extension : <https://forum.cockos.com/showthread.php?t=212174>

ReaVolution theme uses graphical elements provided to the Reaper community by the creator of the original Reaper 6 theme, White Tie.

41.2 Introduction

What is ReaVolution?

ReaVolution is an attempt to provide a customization to Reaper standard configuration in order to simplify its integration to *SPAT Revolution* and provide a complete simple immersive audio creation package.

At the base of ReaVolution is Reaper, a DAW developed by Cockos. It is a very stable and fully-featured DAW with lots of customization possibilities. With its high track channel count

capacities on support for any speaker arrangements, it is one of the most complete flexible DAW regarding immersive sound production.

As the default configuration of Reaper is somehow of a blank slate to create your immersive sound workflow, ReaVolution provides a start point and customization that makes any first approach to Reaper and Spat work like a charm...

ReaVolution's goal is to solve these common problems and smooth out the transition gap for users that comes from other DAWs.

This package includes macros (scripts), customized toolbars, system behaviour preference enhancements. The ambition of ReaVolution is to give a more pleasant starting point for users that are coming from other DAW while giving a new set function dedicated to immersive audio production.

41.3 Table of content

1. **Installation**
2. **Presentation**
3. **Use cases**

41.4 Installation

If you already are a Reaper user, don't worry, ReaVolution will install itself next to your existing Reaper install. You will not loose your current customization and user data. If you wish to take some features from ReaVolution to your current Reaper install, simply use the user configuration import/export feature of Reaper.

To install ReaVolution, you need to follow this steps:

- Download the latest version of Reaper on reaper.fm (currently v6.15)
- Download the latest version of ReaVolution on the Flux center
- Locate the place of the ReaVolution folder. For MacOS users, go to your **Application** folder, and found the **ReaVolution** folder. For Windows users, go to **C:\Program Files\Flux\ReaVolution**.
 - On MacOS, launch the install package and drag and drop the application into the ReaVolution folder.
 - On Win10, launch the install executable, choose “install as portable installation” and choose the ReaVolution folder as the install folder.

First is the play mode. When off, the play cursor comes back to its initial state when the playback is stopped. When on, the play cursor stays at its position when the playback is stopped.

Next, there is a command to show/hide all the automation lane.

i Note

This action is also available with the shortcut “A”

In third, we’ll find the “Show takes lane” button. As its name implies, it shows/hides the different takes. It is very useful to organize its session and only see the various take when we actually need to edit them. It is also safer to turn it off to remove any risk to change a take by clicking on a wrong spot in the arrange view.

In the middle group of icons, we’ll first a toggle to activate the metronome. A right-click on it will open a complete metronome configuration window. Next, there is the Snap toggle. It allows snapping the items we move to the grid or other items. Then, we have something less common. This icon allows locking the time selection. The last button in this group is the ripple editing mode.

The third button group contains four actions. The first one automatically group newly recorded items when on. The second one links the automation point to the item position. This means that if the item is moved, the automation point will move with it. Then there is the master group toggle. When on the item’s groups are effective. When off, all item’s groups are set to off. Lastly, the foremost right icon activates the auto-crossfade option.



Figure 41.3: toolbar2.jpg

On the left side of the arrangement view, above the track panel control (TCP), there are two buttons. The red “M” one is a master mute, the other one is a master solo. If at any point one of the track is muted or soloed, the respective toggle will display on. By setting it off, it clears, respectively, all the mute or all the solo.

i Note

To display the mixer, use CTRL/CMD + =
To display the midi editor, use CTRL/CMD + M

41.5.2 SPAT Revolution library

ReaVolution is designed to work seamlessly with the software *SPAT Revolution* developed by FLUX:: Immersive. *SPAT Revolution* is a stand-alone application to create immersive sound content and support almost any surround sound technologies. FLUX:: has also developed audio pipe technologies which allow sending audio from a DAW to *SPAT Revolution* using a VST/AAX/AU plug-ins. This is a very powerful tool, but it can be quite time-consuming to setup.

ReaVolution comes with a set of action that automates all the things that needs to be done prior to sending audio from Reaper to *SPAT Revolution* and work in pairs with the **Audio Stream library** to support any channel configuration.

41.5.3 Audio Stream library

This library is the core of ReaVolution. Its goal is to provide a way to overcome the limitation that Reaper imposes on track format. By default, tracks and channels can only be even numbers. But in the real world, numerous audio streams come in odd channel numbers, mono being the really obvious one.

How does it work? the library uses the combination of a JSFX called “Audio Stream” and some scripts. This JSFX is present on every track and is used to store and display the stream type to the user. It supports the same streams as *SPAT Revolution*: channels based, HOA, Ambisonics, binaural and transaural.

41.5.4 Various library

Routing library

ReaVolution comes with some powerful macro to quickly manage your routing task: sending tracks, creating busses, folders, you name it. It uses the **Audio Stream library** to support any kind of channel arrangement and be fully compatible with an immersive sound workflow.

Editing library

One of the concerns of ReaVolution is to provide an editing experience closer to the industries standard. Editing is certainly with what most of the new Reaper’s user are struggling with. This library takes its inspiration in softwares that have made their proof in editing task: Pro Tools, Pyramix & Live.

DDP library

This one is a little bonus. It actually was the first one done to satisfy some frustration of the author. Reaper comes with the full support of DDP files, but it is not very easy to create markers and editing them. This DDP library aimed to solve this with a simple set of scripts.

41.6 Use cases

41.6.1 Creating sessions

Each new created track comes with a JSFX Audio Stream plug-in inserted. It does not process audio : it is a simple pass thru. If we open it, we will find no parameters to set and only GUI that display a number of channels/order and a color code that represent the stream type. This GUI is by default shown in the Reaper mixer (also called MCP).

Note

To create a new track, you can either double-click in an empty part of the MCP, or of the TCP. You can also hit CTRL/CMD + N or CTRL/CMD + Shift + N to insert multiple tracks.

All this new tracks are by default *mono* tracks. We can insert track to a precise format by right-clicking in an empty area of the MCP or TCP and by searching the “**Audio stream::insert track**” menu. Inside you’ll find all audio format supported by the Audio Stream library.

In the same vein, we can set the audio stream of an already created track by right-click on it and going to the menu “**Audio stream::set to**”. This can also be done for a selection of tracks.

If you are working on a session that has not been created with ReaVolution, you still can use the “**Audio stream::set to**” menu. It will insert an Audio stream plug-in on each selected tracks.

Warning

Inserting an Audio Stream plug-in will not be sufficient to display the stream type in the MCP. You have to right click on the instance and select “Show embedded UI”.

Note

When defining a channel based audio stream, you can either enter a number of channels (12 for example) or enter a speaker arrangement (5.1, 7.1, 7.1.4, 22.2 etc...).

You can also right click in an empty space of the mixer and select “**Audio Stream::Insert track**”. A pop up window will ask you how many tracks of the selected audio stream type you want to insert in your session.



Figure 41.4: createSession.gif

41.6.2 Dealing with routing

Creating auxiliary tracks

To send tracks to another one, we have three shortcuts : one for post-fader, one for pre-fader and one for pre-fx.

A new auxiliary track gets his audio stream info from its source track. In other words, if you create an auxiliary track for a A-Format track, the destination track will also be A-Format.

i Note

Shortcuts:

ALT + S → Create auxiliary track (post-fader) **ALT + Shift + S** → Create auxiliary track (pre-fader) **ALT + Shift + CTRL/CMD + S** → Create auxiliary track (pre-FX)

Creating busses

In Reaper, the concept of busses is merged with the concept of folders. A folder track has children tracks which are, by default, routed to their parent.

Note

To create a bus, select the desired tracks to group and hit the command CTRL/CMD + G

A bus created with this command checks the audio stream of each track and adjusts its number of channels to fit them all. It also looks at the parent channel of each track. For example, if you have one mono track going to 3-4 channels and another one going to 5-6 channels, it will create a bus routed to 3-4 and reroute the child track to 1-2 and to 3-4.

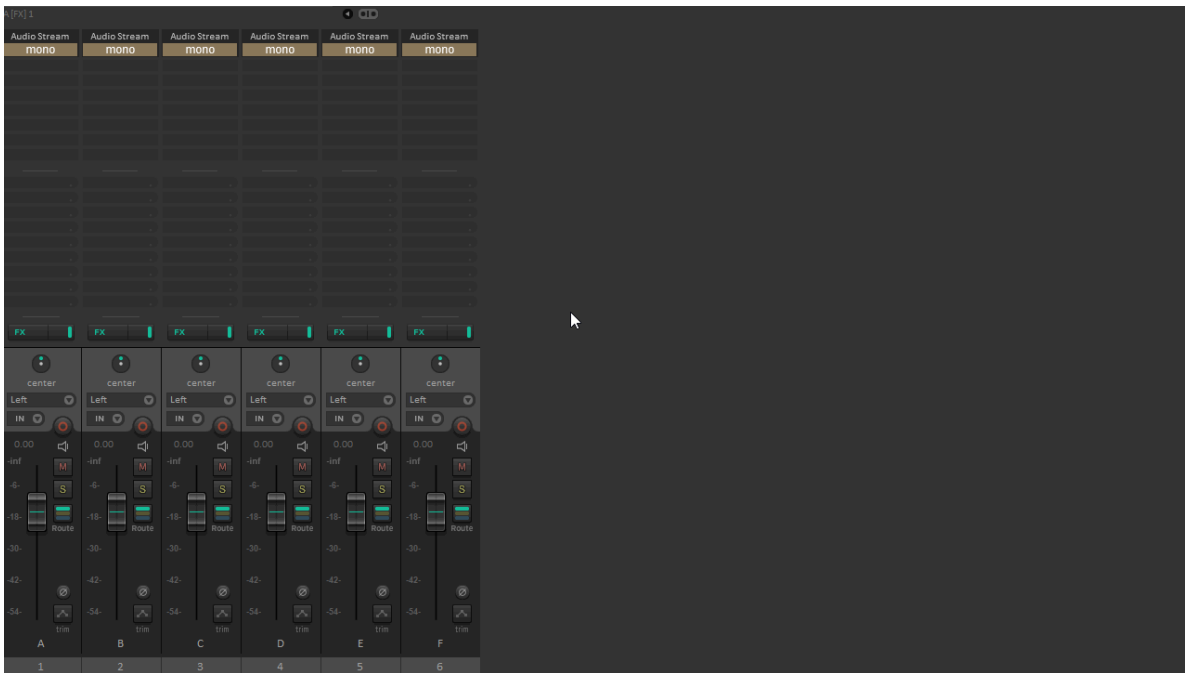


Figure 41.5: bus.gif

Creating multi-busses

This is a new concept brought by ReaVolution.

When working with software like *SPAT Revolution*, we often encounter one problem. Let say we have a drum kit recorded with four microphones: kick, snare & stereo overheads. A common mixing workflow is to create a summing bus and do some processing on it (compression and/or EQing). But then, we can only feed a summed stream to SPAT Revolution and we cannot spatialized each element independently.

Here comes the multi-busses. It is a simple trick actually. If we create a multi-bus from our drum kit, it will create a folder track where the kick is routed to channel 1, the snare is routed to channel 2 and the overhead are routed to channel 3-4. So each element has its channel so nothing is summed here.

Now the trick is to use multichannel compatible plug-ins such as the Flux:: Immersive suite. Their compressors allow to “link” channels to have the same detection signal for all the channels.

This way we can have “bus-like” processing without losing the possibility of spatializing each element.

i Note

To create a multi-bus, select the desired tracks to group and hit the command **CTRL/CMD + ALT + G**

41.6.3 Sending tracks to SPAT Revolution

As discussed below, it is possible to send and receive audio to and from SPAT Revolution with the help of SPAT Revolution plug-ins suite. The “Send” plug-in allows to send audio from the DAW to SPAT and the Return plug-in receives audio from SPAT to the DAW.

Without ReaVolution, there is a certain number of things to do prior to correctly use this plug-ins. First, each track hosting a Send plug-in has to be routed to each track hosting a Return. This has to be done to be sure that the DAW does not process a “return track” before a “send track”.

Then, each track hosting a SPAT plug-in has to be set to “prevent anticipative fx processing”.

When all this set-up is done, we can guarantee a good sync between Reaper and SPAT Revolution. But it can be quite tedious to do manually.

ReaVolution comes with an action that do all of this automatically. With the help of the Audio Stream libraries, it does all the routing automatically. It can even handle multi-bus and access to child tracks’ information.

This action is found by right clicking on a track, inside the menu “**SPAT Revolution**”. It is called “**Route selected tracks to SPAT Revolution**”. This action will create a new auxiliary for each selected track, with the right routing, the right configuration and a SPAT Revolution Send Plug-in inserted.

After running this action, you should be done at least at 95% of your configuration.

41.6.4 Receiving audio from SPAT Revolution

To receive audio from *SPAT Revolution*, simply create a track at the wanted format using “**Audio stream::insert track**”. Then, go back to the “**SPAT Revolution**” menu by right-clicking on the track we just created and choose “**Receive audio from SPAT Revolution on selected track**”.



Figure 41.6: sendToSpat.gif



Figure 41.7: receiveFromSpat.gif

You will find an option related to SPAT routing in the top bar menu, under **Extension>ReaVolution>When tracks are routed to SPAT**.

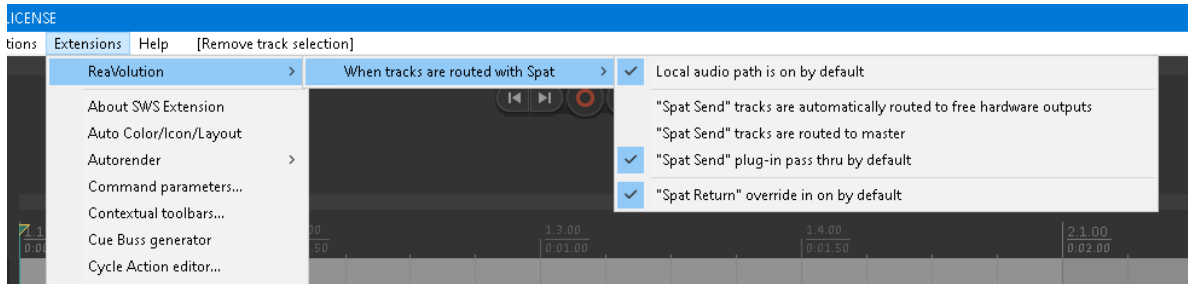


Figure 41.8: reavolutionNewExtensionMenu.png

This option allows to configure the default behavior of sending audio to and receiving audio from SPAT Revolution. By default, the local audio path, which allows sending audio from Reaper to SPAT using virtual audio pipe, is on, the same as the thru option of SPAT send plug-in and the override option of SPAT Return plug-in.

41.6.5 Editing audio

The editing workflow of ReaVolution has been thought to be simple and straightforward.

First thing first, we need to discuss briefly about the basic mouse behavior in ReaVolution :

i Note

Left click/left drag : select time & items **Right drag** : select only items **Middle click** : change edit/play cursor position

Using the mouse

In ReaVolution, some scripts try to simplify the editing behavior. It tooks its inspiration from various DAWs, such as Pro Tools, Pyramix and Cubase.

Items are now divided in two halves. The top half handles the selection action and the bottom half handles the movement action. It is also displayed to user : when the mouse is over the bottom half of an item, the mouse cursor change to a “hand” cursor.

By default, ReaVolution makes the time selection and the loop selection totally independent. The time selection is drawn by left dragging the mouse over the arrange view **or** the top half of items. The loop selection is drawn by left dragging the mouse in the ruler.

Regarding the time selection, it is now quite volatile and easy to clear. A simple click in the arrange view will clear it. The only exception is to click on the bottom half of an item **inside** the time selection.

You will also notice that the play/edit cursor is know very static. To move it, you can either left click in the ruler of the use the middle click of the mouse in the arrange view.

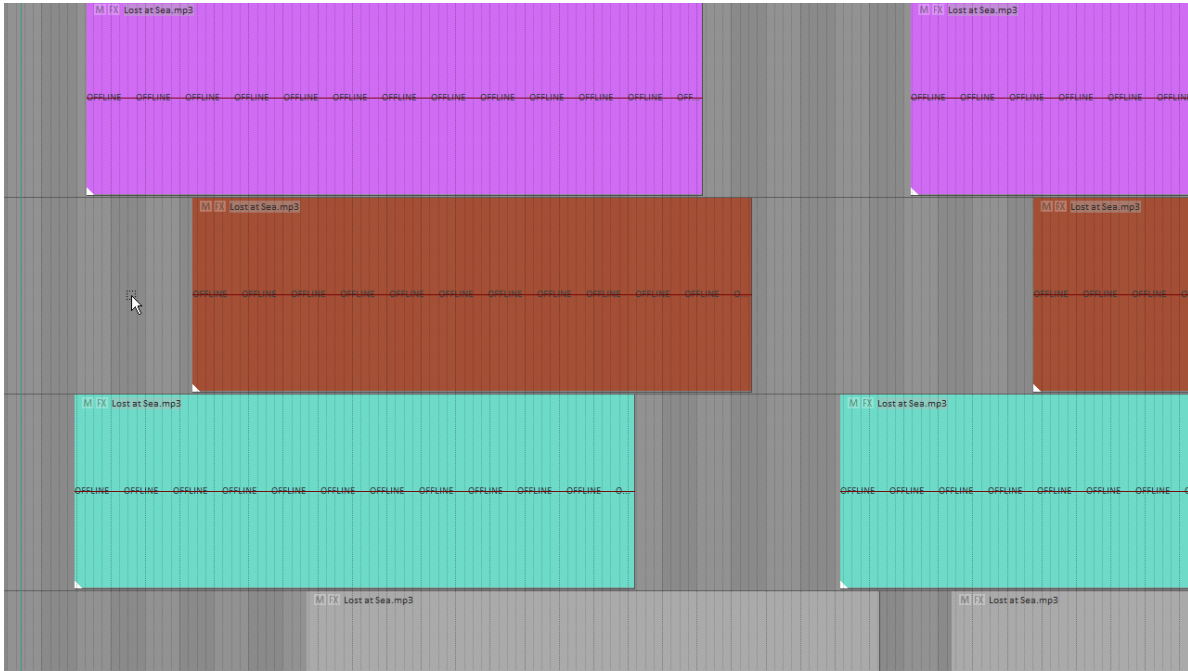


Figure 41.9: mouseEditing.gif

Moving items

Items can be move by simply dragging them by clicking on the bottom half. It is also dependent on the time selection. There is no need to cut the items prior to move part of them. If the time selection is longer than the selected items, the time selection is trimmed to adjust to the item selection.

Advanced mouse usage

- **Selecting time between two items**

Double clicking in an empty area of a track selects the time between two items or between the beginning of the session and the first item on the track.

- **Quickly move items**

ReaVolution has some quick way to move items to the play cursor:

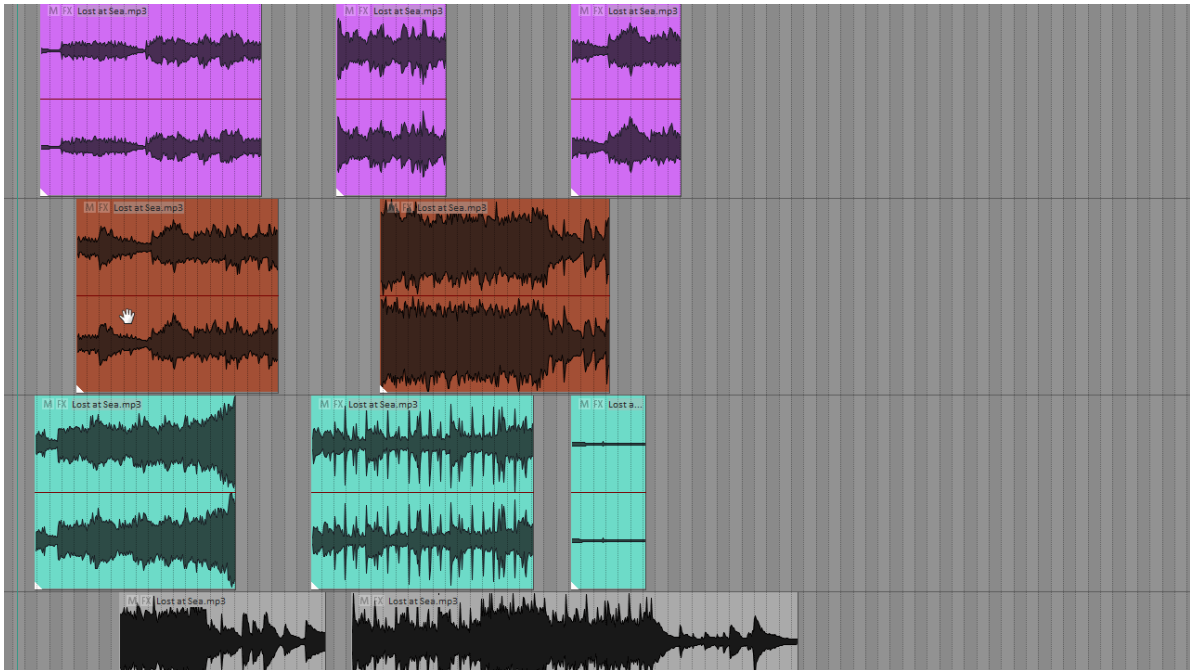


Figure 41.10: itemMove.gif

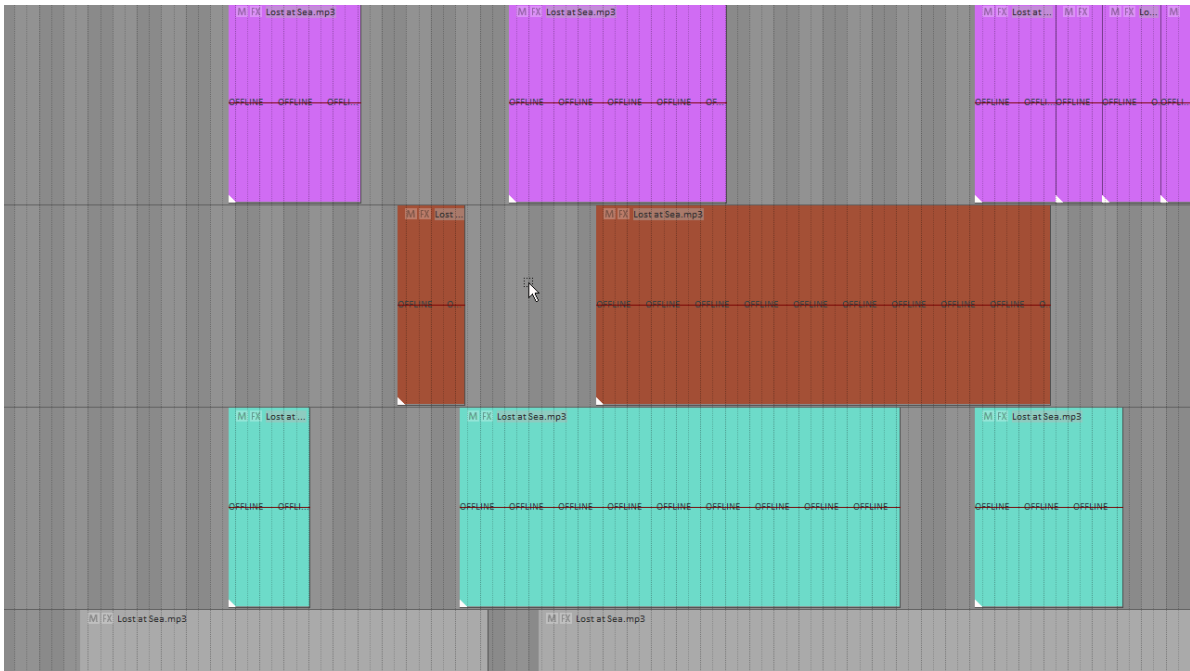


Figure 41.11: selectEmptyArea.gif

i Note

W : move the start of the selected items to the cursor position
X : move the end of the selected items to the cursor position
< : move snap offset of the selected items to the cursor position

You can set the snap offset of an item by using the mouse modifier **SHIFT+CTRL** and by clicking on the top half of the item.

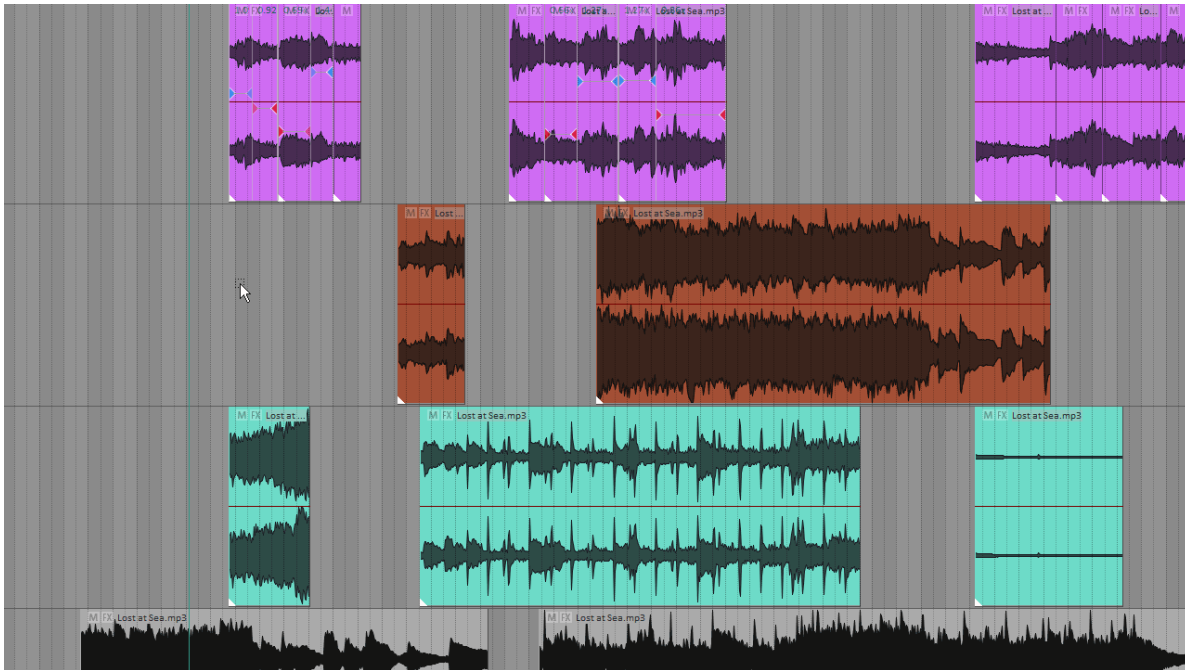


Figure 41.12: moveItem.gif

- **Automatic ripple edit**

Ripple edit is a very powerful feature to quickly edit and move lots of items. The problem is that forgetting to turn it off can do a lot of damage to a session.

ReaRevolution comes with a set of shortcuts and mouse modifiers that take advantage of ripple editing without having to engage or disengage it.

i Note

SHIFT + Left click : move item with ripple edit on all activated tracks
SHIFT + CTRL/CMD + Left click : move item with ripple edit on selected tracks
CTRL/CMD + Backspace : delete content of time selection
CTRL/CMD + SHIFT + Backspace : delete

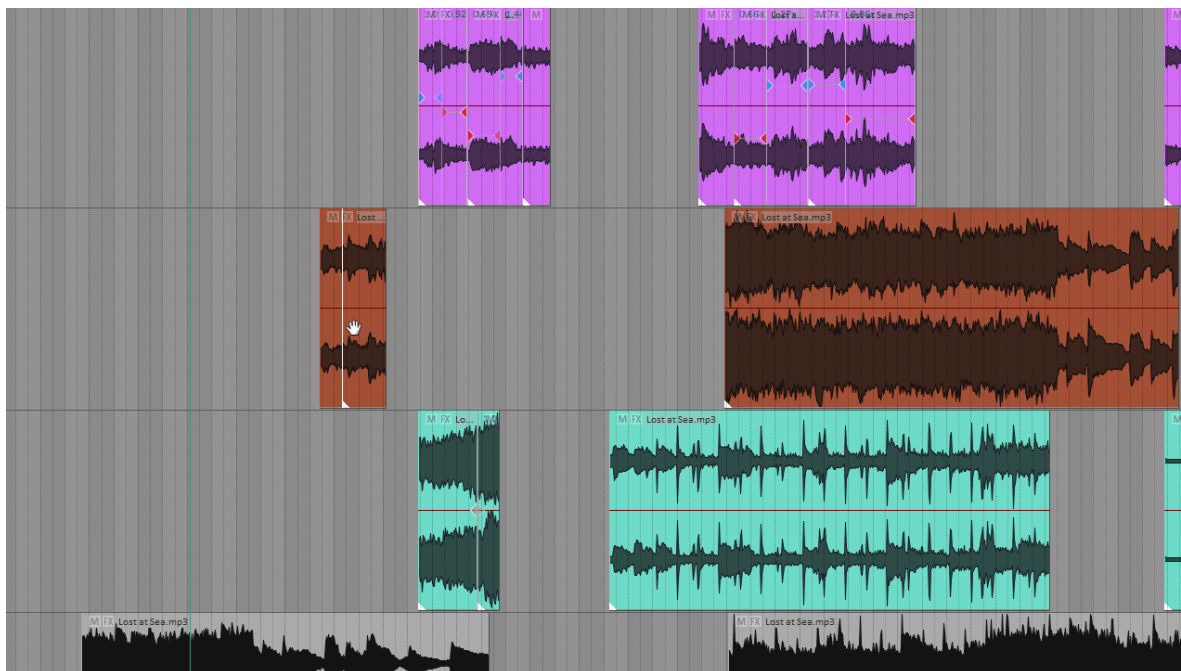


Figure 41.13: rippleEdit.gif

Using the keyboard

ReaRevolution comes with a complete set of shortcuts to easily edit audio items. To make it really simple, a zone of the keyboard is dedicated to editing.

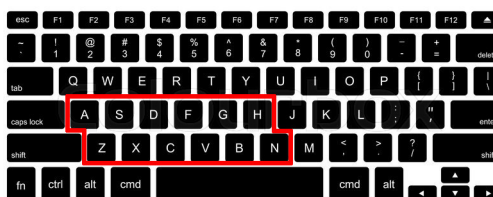


Figure 41.14: keyboard.jpg

To keep things logical, there is two sets of shortcuts provided by ReaRevolution : an AZERTY one and a QWERTY one. By default ReaRevolution is set in AZERTY. To change this layout, go to the action list (shift+?), click on “Key Map” and choose “Import shortcut key map”. In the newly opened explorer/finder window, choose the desire ReaRevolution key map file. **This key maps only affect the shortcut shown in this section.**

i Note

Mouse editing :

B : split item **N** : insert stretch marker **H** : Heal split in item **G** : group/ungroup items **D** : set item fade in **F** : set item fade out **A** or **Q** : trim item start **S** : trim item end **Z** or **W** : set time selection start **X** : set time selection end **C** : “special copy” **V** : “special paste”

With all of this, you should already have most of your editing workflow. There is a few ones more that depend on the cursor or time selection :

i Note

CTRL/CMD + B : split all items that cross edit cursor or time selection **ALT + B** : trim selected item(s) to time selection **CTRL + D** : duplicate items (follow time selection) **BackSpace** or **Delete** : Remove items (follow time selection)

41.6.6 Three and four points editing

When we need to edit big chunk of audio, across multiple takes, it’s hard to beat Pyramix. It uses a “source-destination” system. Sources are the different takes and destination is our final editing. To do the actual editing, we set a time selection inside the source and put a cursor in the destination (3 points edit). The selected time selection will be copied at the destination cursor, moving later destination items further. Or we can also set a time selection for the sources and the destination (4 points edit). This way the selected source will be paste inside the selected time of the destination.

ReaVolution supports a similar way of editing with the shortcut **C**, **V** and **ALT+V**.

i Note

C : copy source items **V** : paste source items at cursor position (move later items) **ALT + V** : paste source items inside the time selection

Zoom and movement shortcuts

i Note

R : Zoom out arrange view **T** : Zoom in arrange view **Mousewheel** : Scroll thru tracks **SHIFT + mousewheel** : Scroll in the timeline **CTRL/CMD + mousewheel** : Zoom in/out arrange view **ALT + Z** : Go back to the last view/zoom position

Some various useful shortcuts

These last shortcuts are not directly related to audio editing, but it can help with this task anyway.

i Note

CTRL + ALT + Mousewheel : Waveform Zoom In/Out **Shift + S** : Toggle solo for selected tracks **Shift + M** : Toggle mute for selected tracks or selected items **Shift + R** : Arm/disarm selected tracks for recording **CTRL/CMD + A** : Select all items/all tracks **CTRL/CMD + SHIFT + A** : Select all items in time selection **CTRL/CMD + R** : Create a new region based on time selection **M** or **Enter (num.)** : Insert a marker at current position **0** : Toggle mute for selected items **Alt + R** or **3** : Record

42 Avid Pro Tools (2022 Update)

Avid Pro Tools is a leading DAW in the professional audio community and is far from new to the object-based mixing workflow. *SPAT Revolution* can use the AAX version of its plug-in suite SEND, RETURN and ROOM to establish connection between Avid Pro Tools and the *SPAT Revolution* environment. These plugins were created at the base to provide the automation link via OSC for local or remote integration. All the SPAT plug-in parameters are made available to Pro Tools to record the immersive creation's metadata (automation).

42.1 Workflow

As mentioned in the integration documents and troubleshooting guide, **using specific tracks (such as aux) as your SPAT object** is the best practice for dealing with these sources/objects you send for external rendering. This way you can leave the session audio tracks and their channel insertions as they are, and simply send these audio tracks to the *SPAT Revolution* object tracks. This allows you to send a single audio track or multiple ones (stem) to the *SPAT Revolution* object tracks. With local audio integration (LAP) or alternate audio routing methods in single or dual computer setups (virtual audio bridge, network I/O, etc.), this approach is flexible.

42.2 2022 Technical Articles

Recent technical articles covers various aspects of Pro Tools integration.

- [Pro Tools integration to *SPAT Revolution* \(2022\)](#) is a good overview of the different possible integrations.
- [Using Pro Tools routing folders with *SPAT Revolution*](#) introduces the concepts of using routing folders as *SPAT Revolution* object tracks. (Requires Pro Tools 2020.03 and above)
- [Pro Tools 2022.9 AUX I/O routing with *SPAT Revolution*](#) covers the Pro Tools newly AUX I/O & Bridge device feature (macOS) and how this can be used to route to/from *SPAT Revolution* the SPAT Object track and the various renders.

- [Reporting latency for delay compensation in *SPAT Revolution*](#) covers the subject of reporting track latency when using Local Audio Path in order for SPAT Revolution to handle delay compensation.

42.3 Templates

Templates downloads

New updated templates are available for use with Avid Pro Tools. They are .ptxt session template files and can be used as examples to see how to integrate *SPAT Revolution*, using the SPAT plug-in suite SEND, RETURN and ROOM. While some templates use the Local Audio Path (LAP) feature, others rely on alternate audio devices such as audio bridge on single machine computer or other high channel count audio devices in dual computer setup.

The [Avid Pro Tools Session Templates](#) is a complete .zip package of all the 6 available templates and can be copied to your **Pro Tools Session Templates** folder.

- macOS and Windows: : Document/Pro Tools/Session Templates/

 Warning

Create a FLUX:: Folder for your templates

After copying the templates to the proper folder, they will appear in the template in the Pro Tools dashboard:

All accompanying *SPAT Revolution* .json session file for the above templates are available as a single download as well: [SPAT Revolution Session Templates](#).

You can download the following session templates and presets:

Pro Tools Templates and *SPAT Revolution* Sessions

- [AVID Pro Tools – Basic Music Local Audio Path](#) is a basic music template using Local Audio Path SEND and RETURN to render Binaural, 5.1, Atmos 7.1.2 output formats with Binaural monitoring. It contains 8 Mono and 4 Stereo object tracks.
- [AVID Pro Tools – Multiformat Local Audio Path](#) is a advanced multiformat template using Local Audio Path SEND and RETURN to render Atmos 7.1.2, NHK 22.2 (with 3 x 8 channel buses in Pro Tools) and 3 binaural outputs. It contains 8 Mono and 4 Stereo, 1 5.1 and 1 7.1 object tracks.
- [AVID Pro Tools Studio – Audio Bridge \(32ch\)](#) is a template for using audio interface for routing (single or dual computer) and is tested for support to Pro Tools 2022.09 AUX I/O & bridge feature. It renders Atmos 7.1.2, Binaural and HOA 3rd order. It contains 24 Mono and 4 Stereo object tracks for compatibility with Pro Tools Studio.

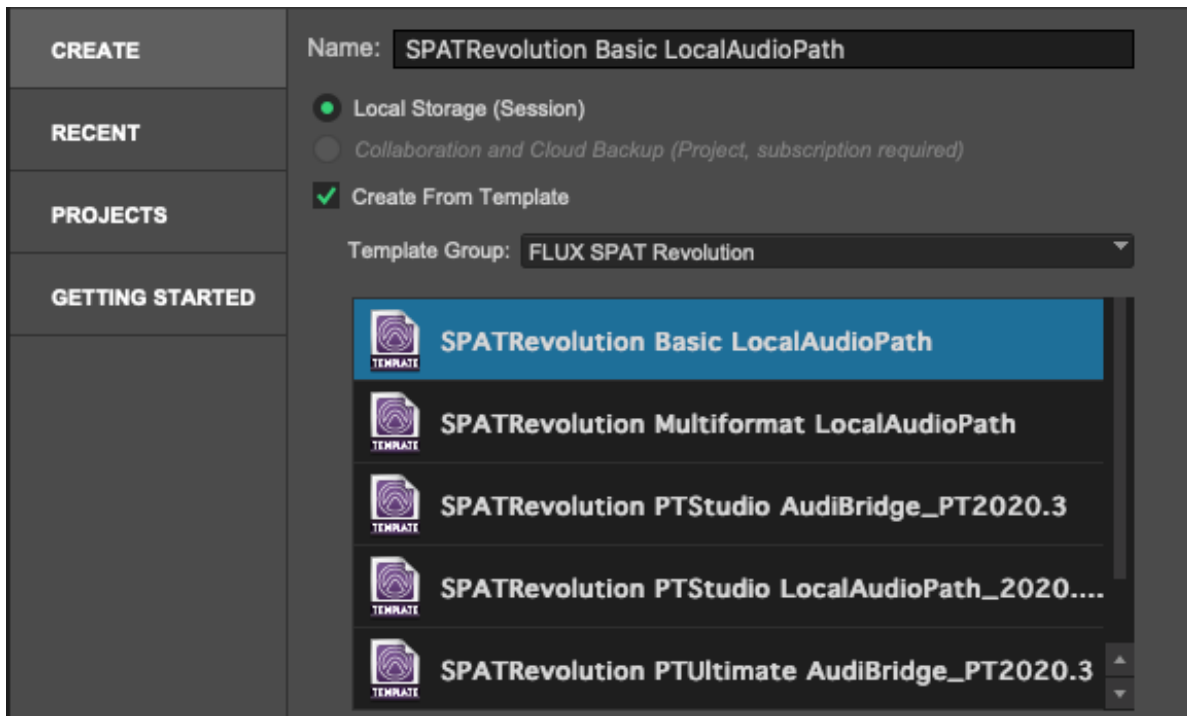


Figure 42.1: Create a new session

- **AVID Pro Tools Studio – Local Audio Path (32ch)** is a template for using Local Audio Path SEND and RETURN for routing (single computer). It renders Atmos 7.1.2, Binaural and HOA 3rd order. It contains 24 Mono and 4 Stereo object tracks.
- **AVID Pro Tools Ultimate - Audio Bridge (64ch)** is a template for using audio interface for routing (single or dual computer) and is tested for support to Pro Tools 2022.09 AUX I/O & bridge feature. It renders Atmos 7.1.2, Binaural and HOA 3rd order. It contains 32 Mono, 8 Stereo and 4 HOA 1st order object tracks.
- **AVID Pro Tools Ultimate - Local Audio Path (64ch)** is a template for using Local Audio Path SEND and RETURN for routing (single computer). It renders Atmos 7.1.2, Binaural and HOA 3rd order. It contains 32 Mono, 8 Stereo and 4 HOA 1st order object tracks.

 Warning

More object tracks can be added depending on audio interface and/or computer resources

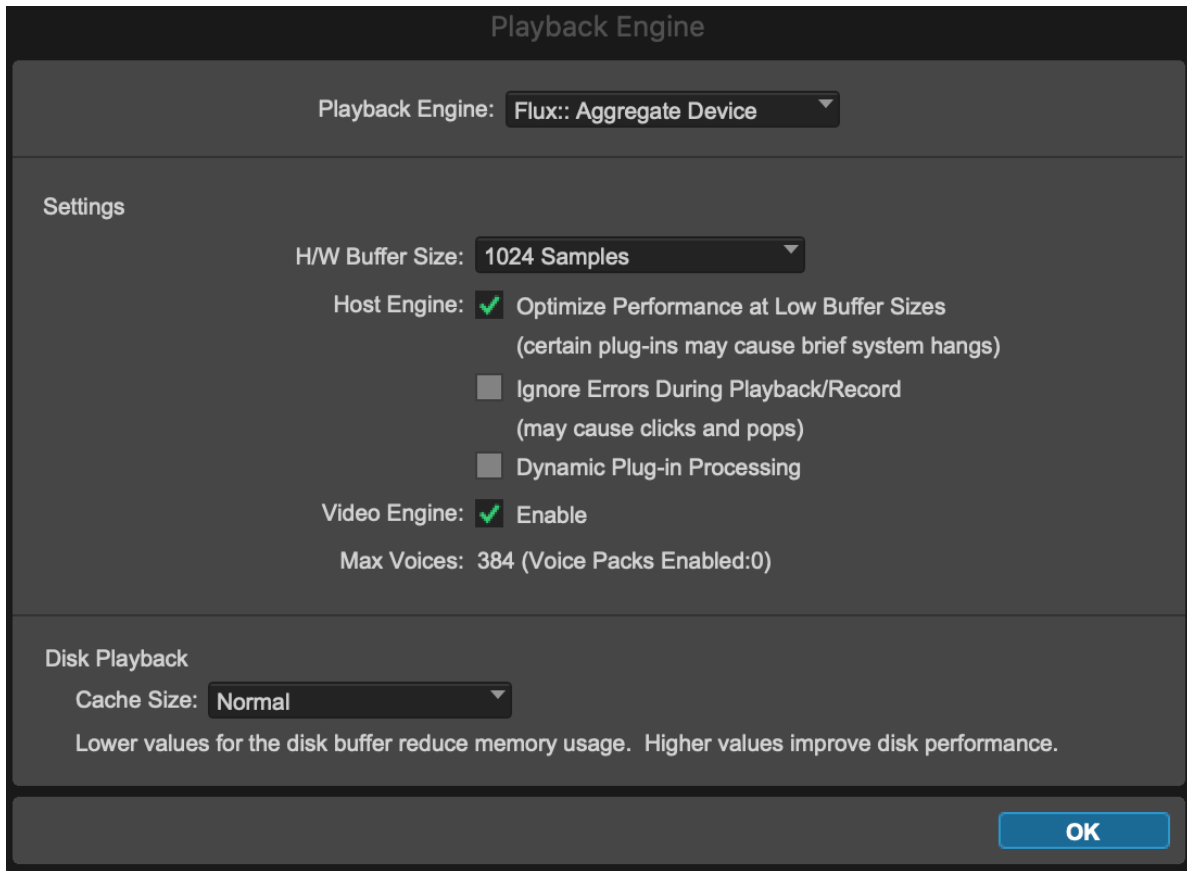
For troubleshooting, please review the **Appendix B - Troubleshooting C**

42.4 Setting Up Sync when using Local Audio Path (LAP)

With the latest release of *SPAT Revolution*, buffer size in Pro Tools needs to be set to 1024 samples, in both ProTools and *SPAT Revolution* if using the Local Audio Path option. In ProTools, go to *Setup/Playback Engine* then set H/W Buffer Size to 1024 samples.

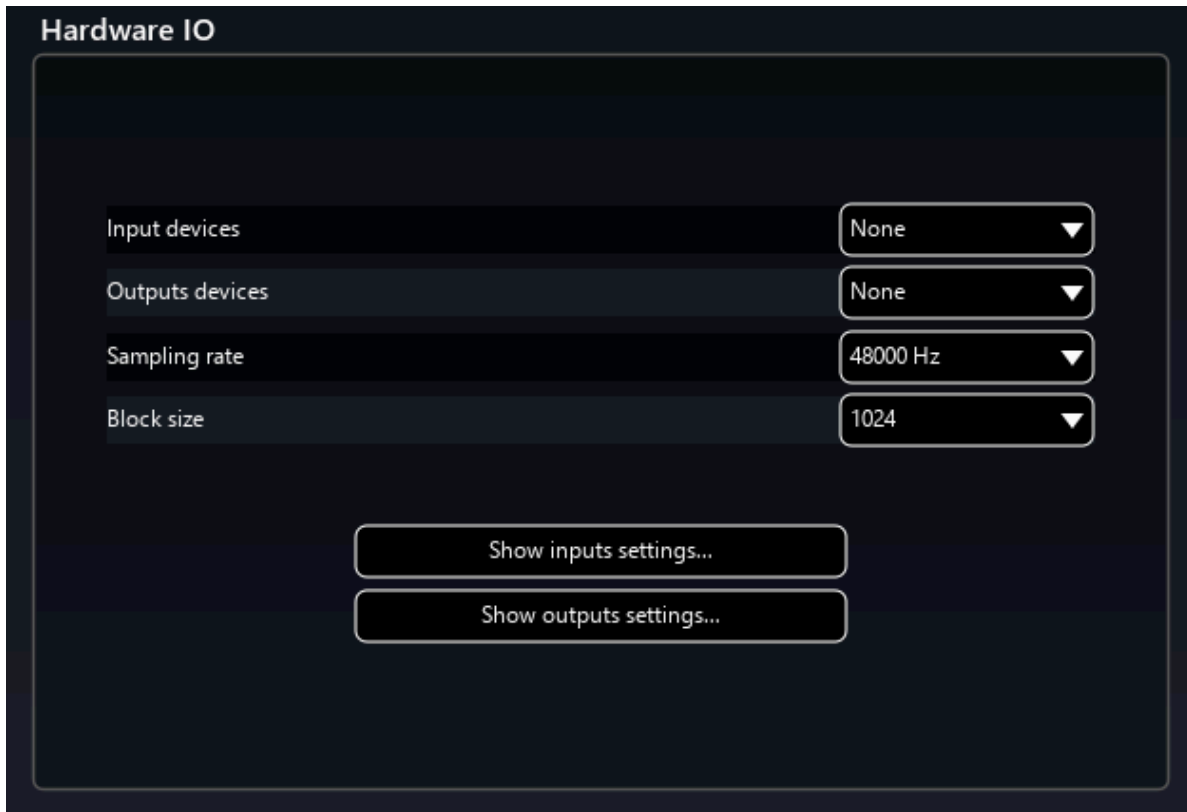
 Note

We highly recommend de-activating dynamic Plug-in processing.



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

- Device: None
- External Sampling Rate: Enabled
- Block Size: 1024



42.4.1 Latency compensation

42.4.2 Setting Up Object Tracks

Setup a Protools session to work with SPAT Revolution using the Local Audio Path mode is simple. Just add SEND and RETURN plug-ins and enable the 'Local Audio Path' (LAP) option for each. Although SPAT plug-ins are instantiated when using FLUX Pro Tools templates or presets, they don't have the LAP activated by default. (Supporting standard audio routing of objects to SPAT Revolution as external renderer)

Once you engage the *Enable* switch, SEND software inputs should appear at the top of the SPAT Revolution setup environment page, and Return outputs at the bottom.

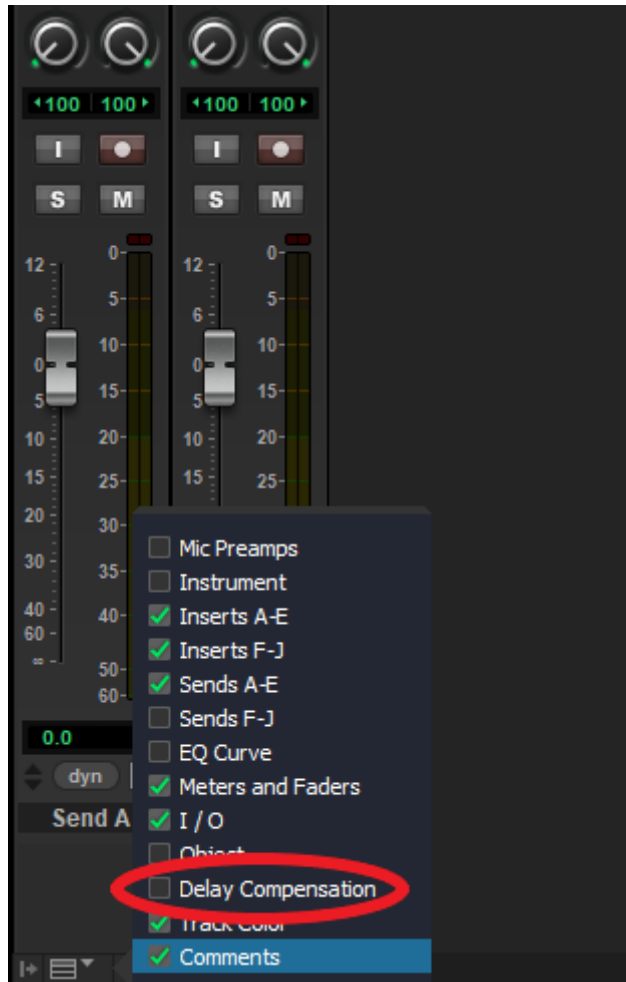
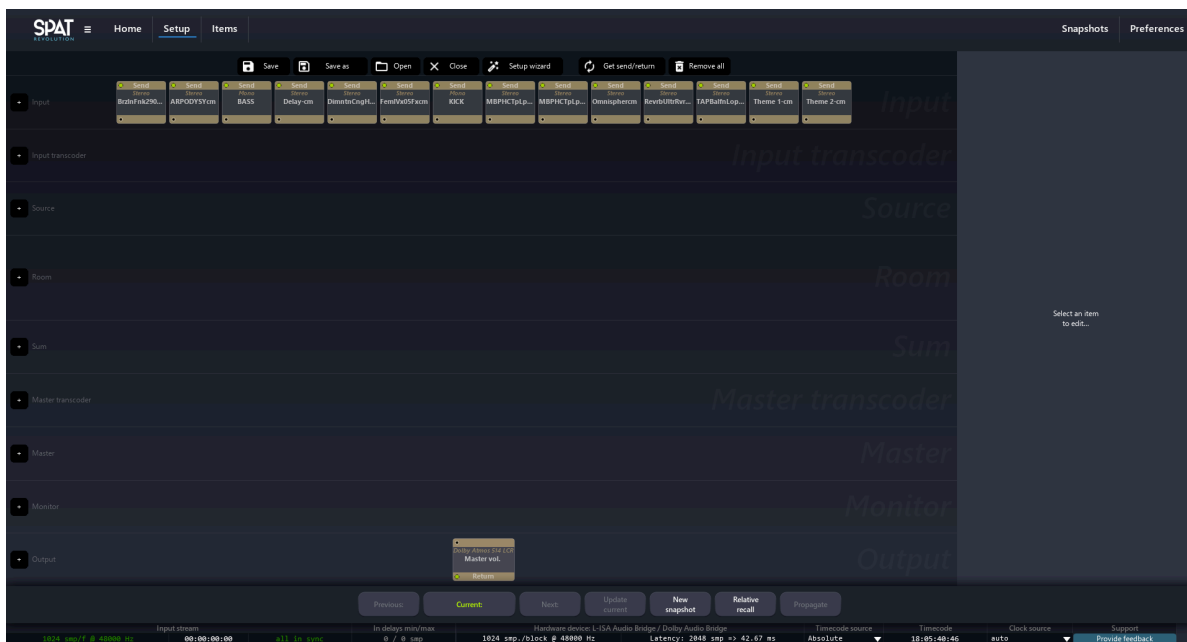
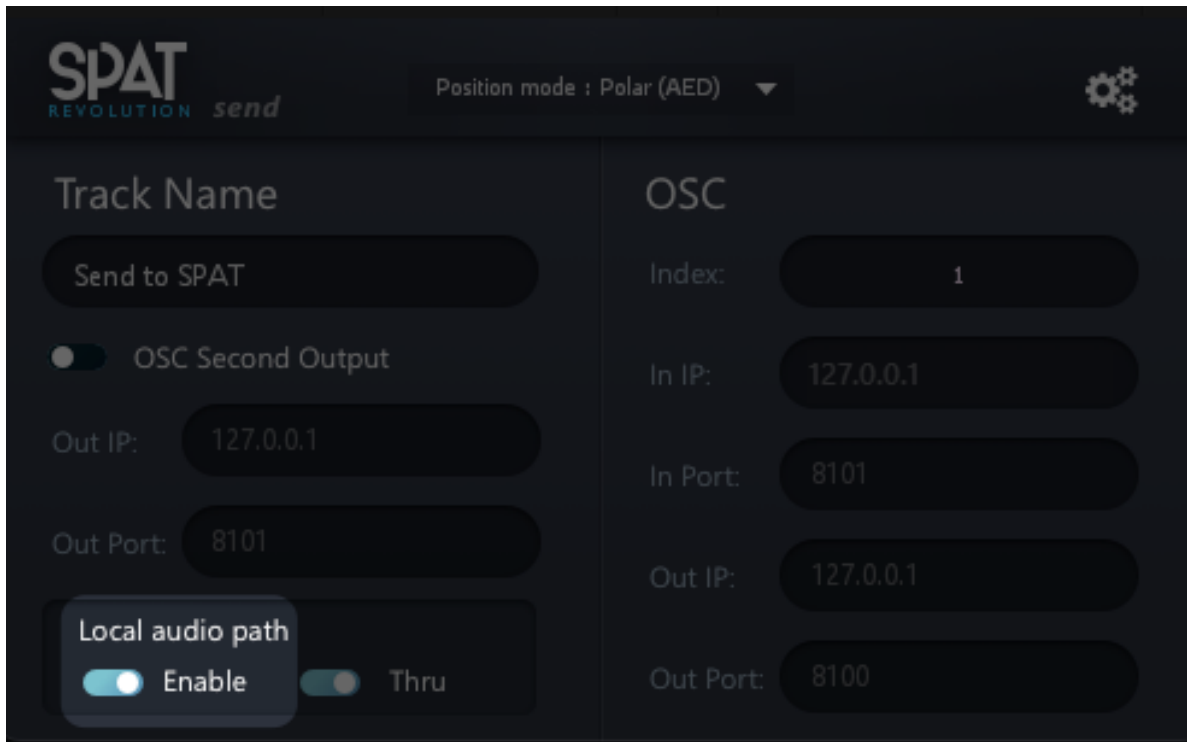


Figure 42.2: Show latency



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 module routing will be completed automatically with a default stereo virtual room.

Now you should be able to send and receive audio streams from/to Pro Tools and spatialize your objects.



42.4.3 The SPATSync (aka dummy bus) and why

Quite simply, all tracks or Aux object tracks that have a SPAT SEND need to be outputting to a common bus feeding each the input SPAT RETURN buses, in order for the SPAT Send to be computed before the SPAT Return. Using FLUX:: Immersive Pro Tools templates and sessions provided prevents the user having to do these steps.

A simple, nice and clean way to handle SPATSync bus for all scenarios and formats is to create a 16-channels ambisonic dummy bus (the largest bus that Pro Tools supports) and include all the sub arrangements from mono all the way to 3rd order ambisonic: HOA3, HOA2, HOA1, Atmos 7.1.2, 7.1, 6.1, 5.1, 5, 4, LCR, Stereo and Mono. Whatever the track format SPAT SEND plug-in is, it can be sent to this dummy bus that can handle all the formats. Then all these buses are available to feed any SPAT RETURN whatever the channel count.

This is the strategy used in templates.

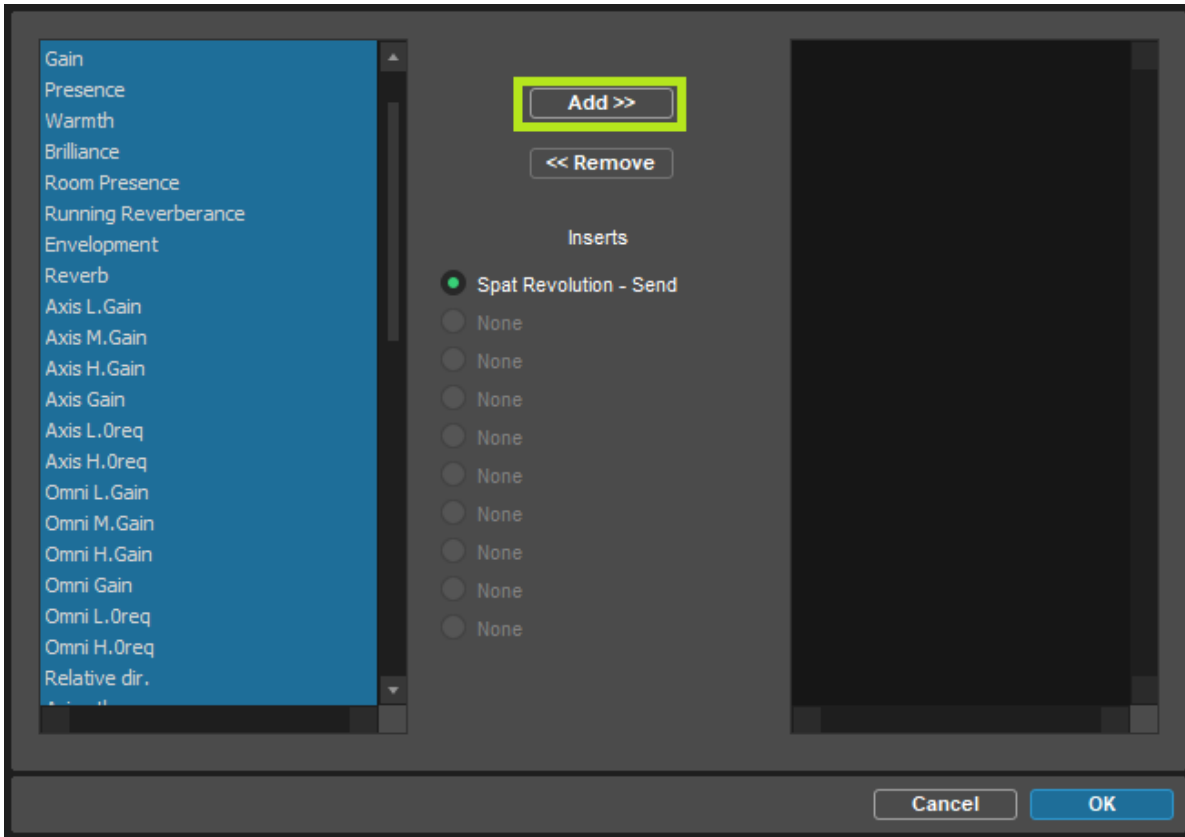
42.4.4 Pro Tools license versions

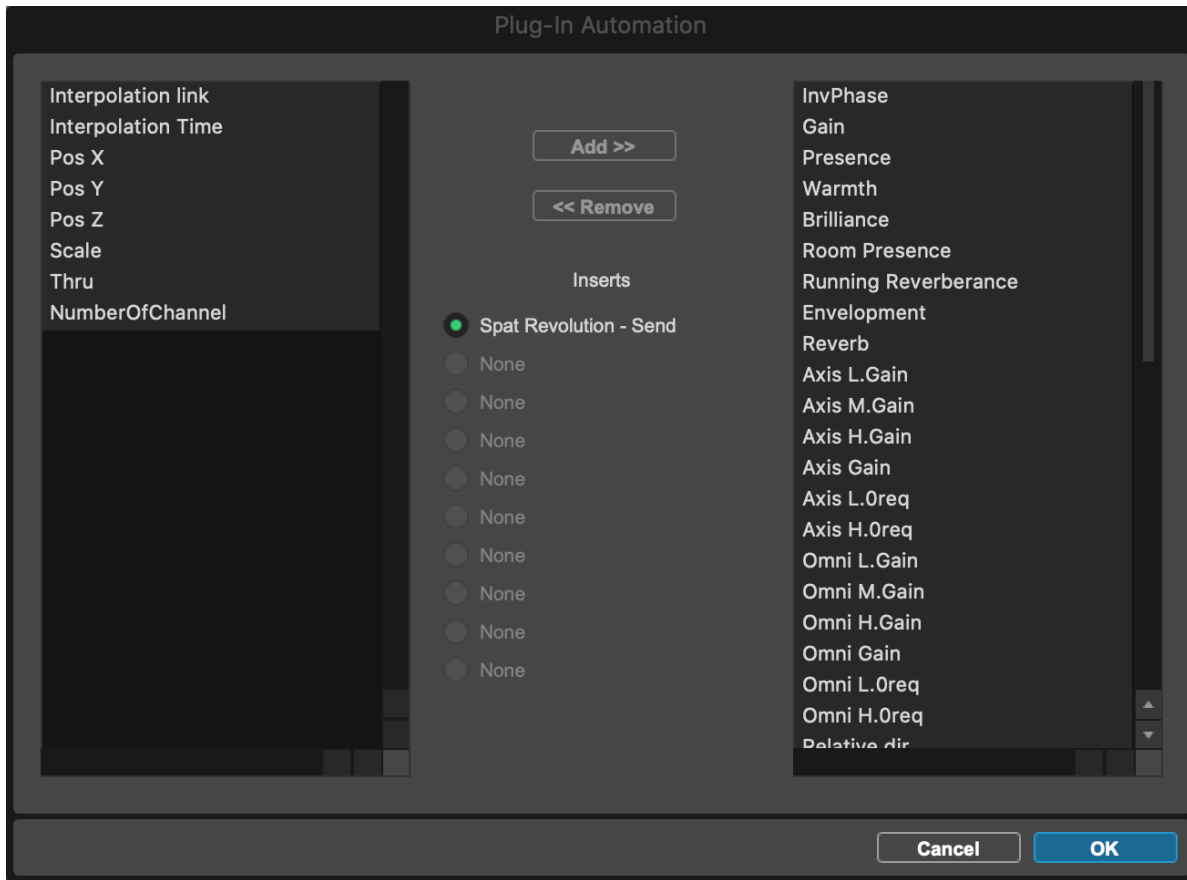
The Ultimate and Studio versions of Pro Tools are highly recommended because of the use of multichannel buses. Using a version of Pro Tools below forces us to return our audio streams by groups of maximum two channels. For example, you will need 12 SPAT RETURNS plug-ins (and 12 x 2-channels tracks) to return a 22.2 stream from *SPAT Revolution*.

This strategy is still used when dealing with channel-based format higher than the supported bus in Pro Tools. So for example for a NHK 22.2 bus, we can return this to 3 x 8-channels (7.1) bus or to 2 x 16-channels ambisonic buses where only 8 channels will be used on the second bus. So, Pro Tools Ultimate and Studio are highly recommended. Pro Tools Advance Multi-format template for *SPAT Revolution* reflects this.

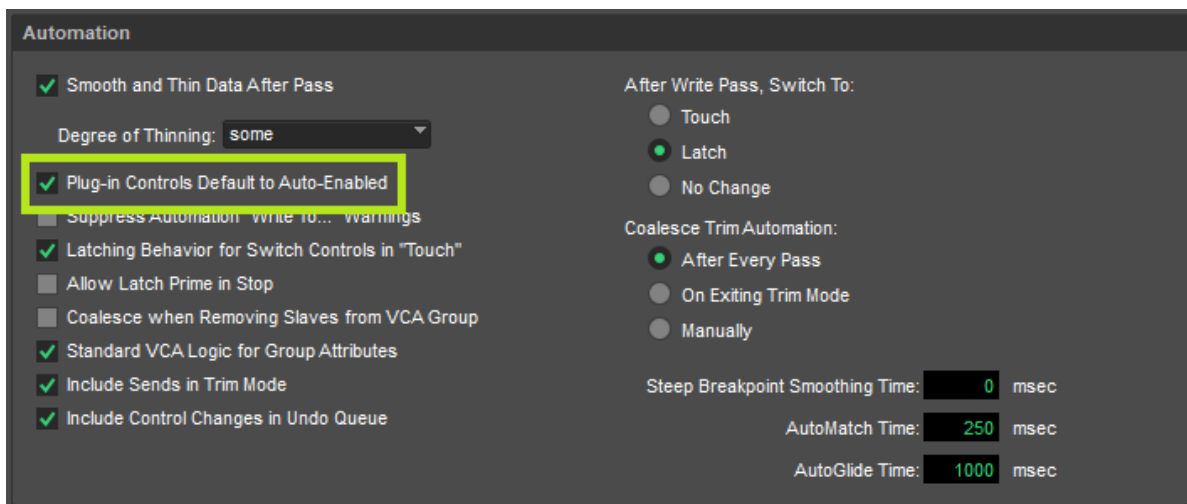
42.5 SPAT Source/Object Automation

To automate variable parameters in the SPAT rendering engine from Pro Tools automation lanes, activating the parameters for each plug-in instance first is necessary. This is already done in the above templates and track presets.





To do this one by one can be a laborious task. Save time by editing the Pro Tools preference for automation and enabling the ‘*Plug-in Controls Default to Auto-Enabled*’ option. When inserting a plug-in, all its parameters will automatically be available for automation.



42.5.1 Source/Object Index, moving from LAP to OSC

Another thing to remember is that as you instantiate plug-ins, they automatically generate an Index # that gets used when in OSC mode (without local audio path enable). So, if you take a template (or a fully deployed session) and delete a bunch of plug-in instances or create new instances, you end up with a mix of index numbers which may mean manually having to change them later. That can be a tedious job when dealing with a larger session. Granted this is not important when using LAP. Still, the day you decide to separate your playback of the rendering computer, disengaging the Local Audio Path (LAP), you end up with the same reality of Index # not reflecting your setup.

The rule of thumb is simple. The index # presents the Source # in *SPAT Revolution* and deleting (or re-ordering in the case of SPAT) will change the index number. Using the plug-in in OSC mode will impact the source that the SPAT plug-in is speaking to.

43 Ableton Live

43.1 Templates

Ableton Live is a well-known DAW for its creative possibility and has been preferred by content creators for many years. The fact that audio tracks in Ableton Live are always stereo, or at least they only support two-channel audio streams, does bring limitations to dealing with multichannel immersive content and the ability to return to Ableton Live the various streams.

One workflow is to use Ableton Live sources as objects to SPAT, use the SPAT plug-ins suite SEND and ROOM for automation parameters, and route audio via various hardware audio interfaces (including virtual soundcards and network audio) in a single or dual computer setup. The dual scenario is preferred as this level of processing can become heavy on larger sessions, at a higher sample rate, and with much automation.

To use the Local Audio Path (LAP) feature of the SPAT SEND and RETURN plug-in, a workaround to the stereo limitation needs to be applied if you want to integrate the audio portion in this manner.

New templates are available for use with Ableton Live. They are start sessions and can be used as examples to integrate *SPAT Revolution* using the SPAT plug-in suite SEND and RETURN. Now, let's dive into how to use *SPAT Revolution* with Ableton Live and how our templates work.

You can download the following session templates:

Project Sessions and Templates downloads.

- [Simple Stereo Binaural Project Template](#)

The console is organized in this way:

- The first eight tracks are where you will put audio events or even live input.
- Then is a folder containing eight other tracks. Each of our audio input tracks is routed to one of these tracks. These tracks are hosting SPAT SEND plug-in. They are your SPAT sources/objects and the bridge between Live and *SPAT Revolution*.
- A “REC” folder contains another bunch of tracks. They are for recording your audio streams returning from *SPAT Revolution*.



Figure 43.1: LIVE_TMPLT__simpleStrBin

- Lastly are a couple of “return tracks”, as named after Ableton Live nomenclature, which hosts your SPAT RETURN plug-ins.

Note

To keep the console organized, we put tracks inside folders. But, as there is a SPAT SEND on each of the sending tracks, and as the recording tracks are routed straight to our audio interfaces, there will never be any audio on these folders.

43.2 DAW optimization

How to bypass Ableton Live optimization?

As you may know, Ableton’s DAW is very focused on live performance and comes with many optimizations to reduce CPU usage. Its strategy seems to deactivate every processing unit, like tracks or plug-ins, that does not receive any signal at their input and does not send any to their output. If a track is actively monitoring input, no optimization is applied.

If it may be very efficient from the point of view of CPU consumption, it is a real problem with our SPAT plug-ins. This means that if a track hosting a SPAT plug-in does not play audio for a certain amount of time, the plug-in will be deactivated behind the hood and *SPAT Revolution* will lose the sync.

To each problem, a solution

To overcome this behavior, we create a straightforward Max4Live device that keeps a track “alive”. It’s called “SyncBox”: it generates an inaudible pink noise to keep the processing of the following devices on. To ensure a steady sync, you must add this device upfront for every SPAT send device in your session.

If you do not own a Max4Live device, you must ensure each track hosting a SPAT SEND plug-in will be fed with audio at any time. Another trick may be to route a silent input of your audio interface to tracks hosting a SPAT SEND plug-in and activate the monitoring input to force these tracks to stay awake.

How to install our Max4Live device?

To install our Max4Live device, simply go to your Ableton Live user library. If you don’t know where it is located, go to your preferences under the tab “Library”. You will find the information in the screenshot below.

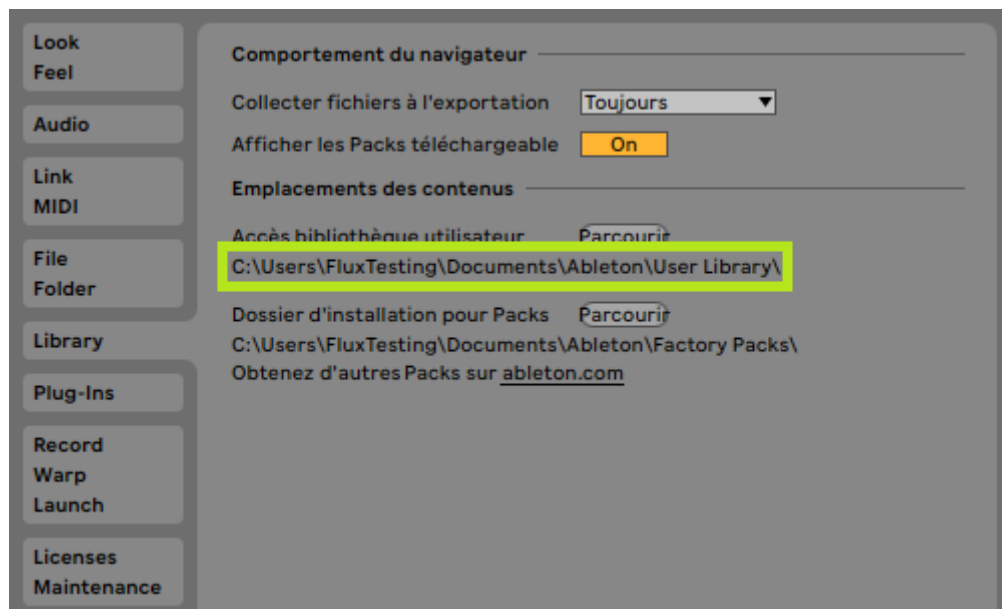


Figure 43.2: LIVE_TMPLT__simpleStrBin_userLib

Then, from the User Library, go to this location:

User Library\Presets\Audio Effects\Max Audio Effect

Simply drop the *SyncBox.amxd* device in this folder. It can be downloaded here.

[SyncBox Optimization workaround](#)

43.3 Ableton Live routing

The golden rule when working with *SPAT Revolution* plug-ins is to ensure each track containing a SPAT SEND plug-in is routed to **every and each** track containing a SPAT RETURN plug-in. This way, we are absolutely sure every SPAT SEND plug-in is processed before each SPAT RETURN plug-in.

For our purpose, Ableton Live has a very interesting type of track called a “Return track”. It is very similar to an auxiliary bus in an analog console. When a new return track is created in Ableton Live, all the audio and MIDI tracks are automatically routed to it. This is a perfect solution to work with SPAT Revolution as it saves us from a lot of routing we have to do in other DAWs.

43.3.1 Audio source tracks

Your audio tracks will have their audio routed to “send only”. This way, they do not send any audio to the master track.

Audio sources are being played/read on these tracks. If you are wondering where their signal is going, just continue reading: it will become clear below.

43.3.2 Object tracks “Send to SPAT”

In the “Audio From” section, you will notice that you are receiving audio from your source tracks. Again, these tracks are not routed to the master.

You will notice that these tracks have their send volume set to -inf dB. We do not need to send audio to validate the routing priority.

Selecting one of this track will reveal a SPAT send plug-in and our SyncBox workaround.

43.3.3 Return tracks

Our return tracks in Ableton Live host our SPAT Return plug-in and are unrouted from the master.



Figure 43.3: LIVE_TMPLT__simpleStrBin_inputIO



Figure 43.4: LIVE_TMPLT__simpleStrBin_sendIO

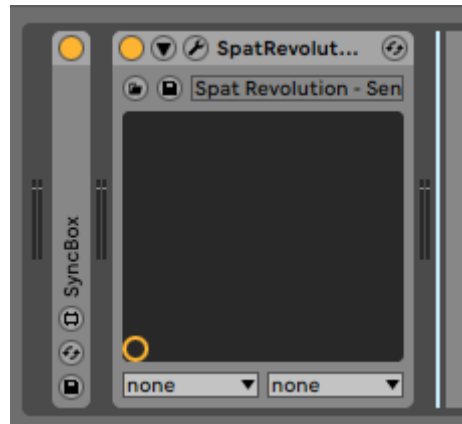


Figure 43.5: LIVE_TMPLT__simpleStrBin_chProcess

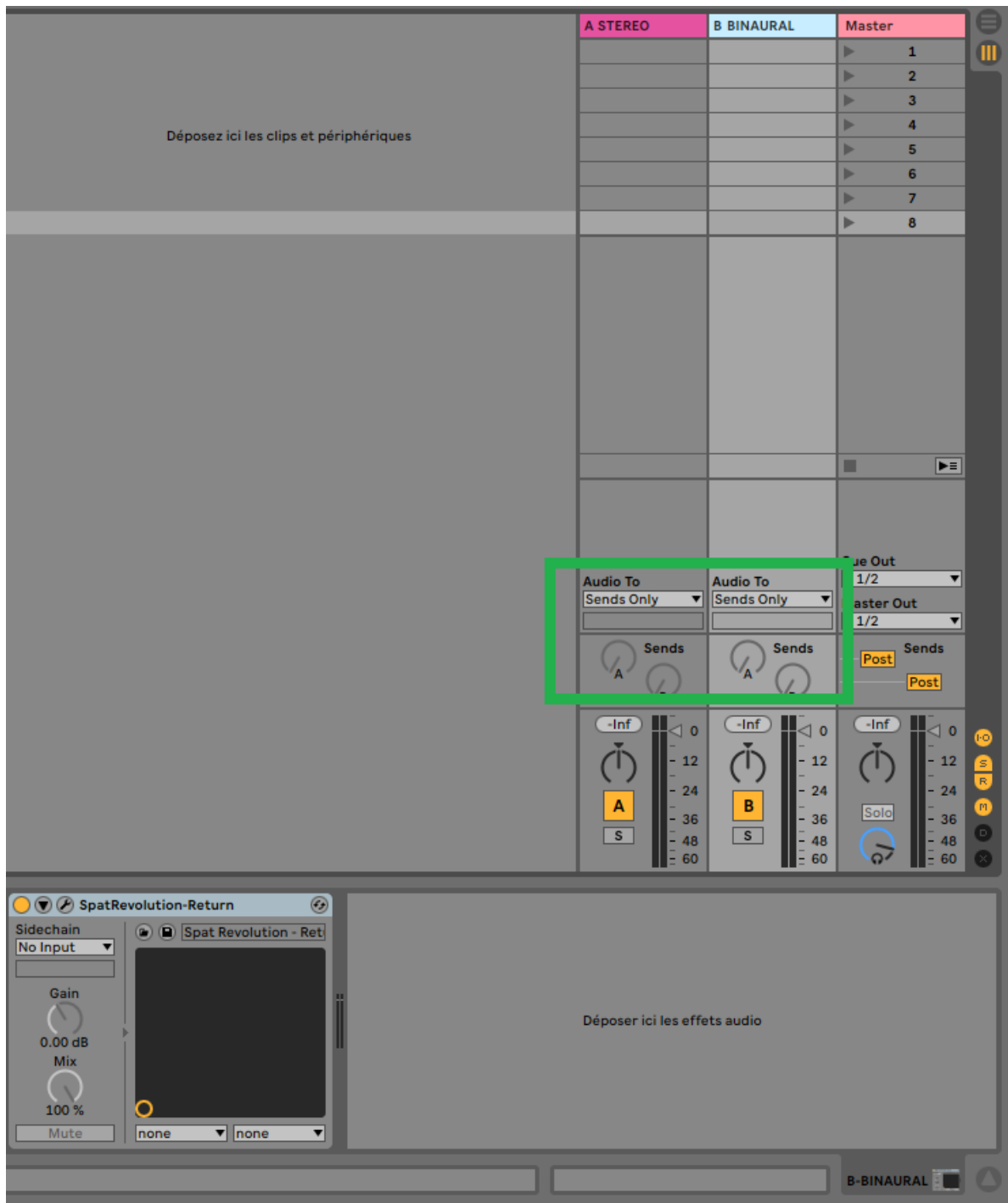


Figure 43.6: liveTemplate1_returnIO

43.4 Recording the immersive creation

Now that we have audio going and coming from SPAT, we can record our work to publish or archive it. As it is impossible to record audio on a return track in Ableton Live, the output of your return track is sent to the actual audio tracks.

To do so on a manual setup, we need to create as many audio tracks as we have return tracks hosting a SPAT RETURN plug-in. Then, for each new audio track, we will go to the I/O and, under the label “Audio From” choose each return track. Now, for the actual return track, we will unroute them from the master. In the I/O of each audio track, we will go under the label “Audio to” and choose “Send only”.

Lastly, to hear our audio, we must put the monitoring mode of our recording track on “In”. Of course, we will also need to arm our audio track to record something.

In the template, the tracks allocated to recording are configured to get their audio from the return track of Ableton Live. Their monitoring is switched on, and they are armed for recording. Also, their output is routed directly to the audio interfaces.

i Note

You will need to manually adjust this routing for your needs.

43.5 Higher number of channels

- [Dolby Atmos 7.1.4 and Binaural Project Template](#)

As we discussed above, Ableton Live can only handle two channels of audio streams. To overcome this limitation, we must use several return plug-ins to get our signal from a multichannel room output stream. This template shows how to return the audio from a 7.1.4 Dolby Atmos room. As we need 12 channels, six different SPAT stereo RETURN plug-ins are used.

The internal routing in Ableton Live remains the same as in the earlier stereo & binaural template above.

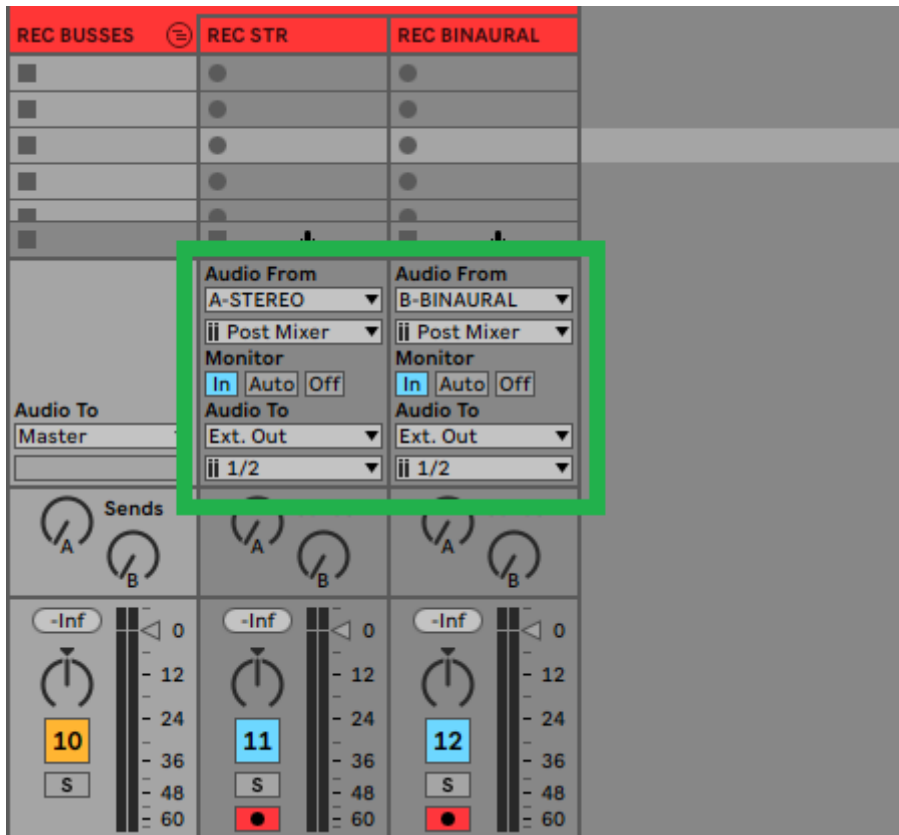


Figure 43.7: liveTemplate1_recIO

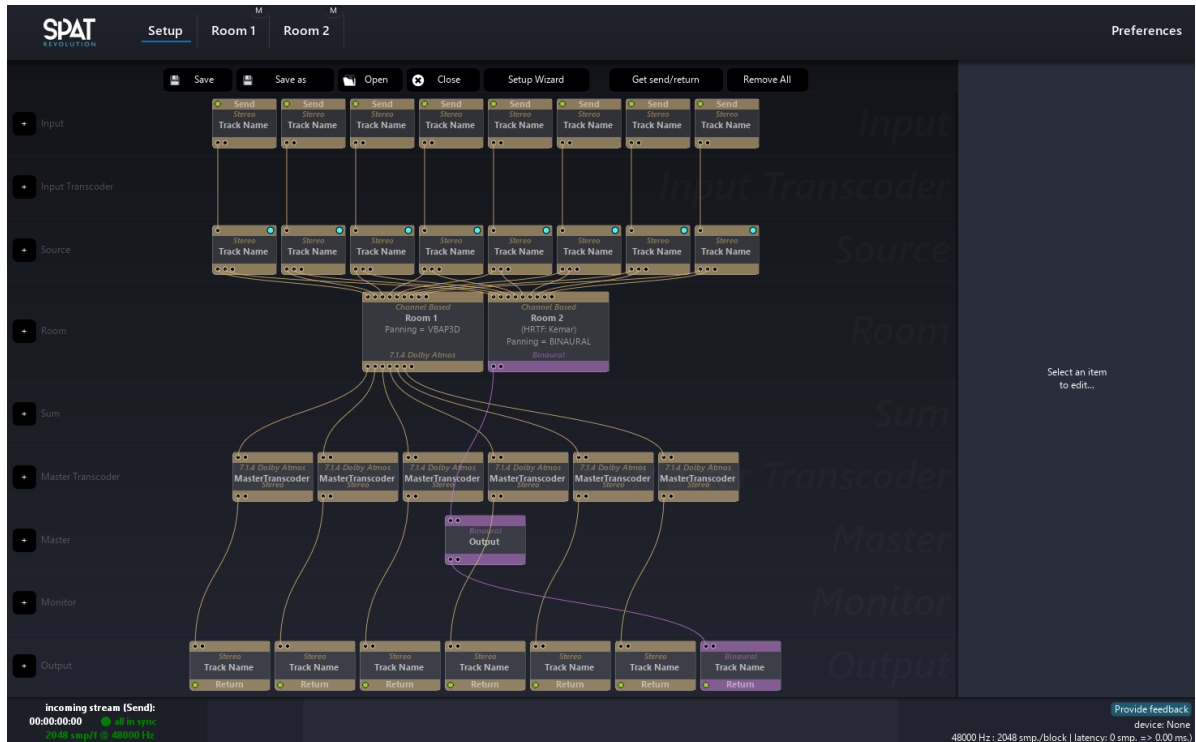


Figure 43.8: liveTemplate1_SpatSession

44 Ableton Live Tools for SPAT Revolution



Figure 44.1: Guide Cover

i Note

This guide is also available as [YouTube Videos](#).

44.1 Introduction

This is a complete guide of a workflow with a set of custom-built Ableton Live Devices, created to increase significantly the compatibility between the DAW and SPAT Revolution.

We will focus on everything from downloading and installing the tools, setting them up, optimizing your computer, a complete description of the provided tools, and then the actual workflow and how to create an entire project in the most efficient way possible!

44.1.1 Why use Ableton Live with SPAT Revolution?

Maybe Ableton Live is your main DAW, so it is the most obvious choice. This is the case for some of us at FLUX:: and is also why we created this workflow in the first place, but we would also like to add, that Live could be an excellent choice for anyone getting into creating, sound designing, etc... in an Immersive Audio Context.

The capabilities of Ableton Live in terms of automation are monstrous, making this DAW an incredible tool for making great Immersive Audio content that will include lots of spatial movements and effects.

44.1.2 Why use SPAT Revolution at all?

Immersive Audio is gaining more and more traction recently, with Apple Spatial Audio, Dolby Atmos, and Binaural. You may have heard of those if you have worked in the music industry at any level these past few years.

So SPAT Revolution will help you to create and to expand ideas into a whole new world or should I say, into new spaces!

And now, after reading this guide, it will be as easy as making good old fader automation, as this is not just a do this and it will do that guide, but more of a do this and here why it does that, type of guide!

This guide works as a base and not an “end all be all”, any ideas on how to improve this workflow, the devices provided, the templates, and any other type of feedback are very welcomed!

Warning

Disclaimer: This workflow has been specifically considered for maximum compatibility between Ableton Live and SPAT Revolution under macOS. Some parts of this guide (*mainly the devices provided*) can still be applied to Windows or/and other DAWs.

44.2 Demo

<https://www.youtube.com/embed/l575MU3l5SI>

44.3 Installation

i Note

If you already have a FLUX:: account and an iLok account and already have a SPAT Revolution License or Trial, you can jump to the next section of the guide.

You can also follow this part with the dedicated video:

<https://www.youtube.com/embed/7govpoyIFeA>

- So, we are going first to create our FLUX:: account [here](#).

Don't forget that you need to activate your account; an email will be sent to your email address.

(if you don't see it, please check your spam inbox or search for the email directly using its name: "email address verification")

- Once your FLUX:: account has been created, we are now going to create an iLok account [here](#).

For those who don't know, iLok is simply a license platform and will allow us to activate our SPAT Revolution license.

Same as for the FLUX:: account, don't forget to activate your account with the email sent to you.

We will now download the FLUX:: Center and the iLok License Manager app.

Download: - FLUX:: Center: <https://www.flux.audio/download> - iLok License Manager app: <https://ilok.com>

Now that both accounts have been created and activated, and both the FLUX Center and iLok license manager apps have been downloaded and installed, we can now link our iLok account to the FLUX:: account.

To do so, simply go into the [iLok Licenses](#) section of your FLUX:: account, and follow the procedure.

The last step is to go into the trial section and request a trial for *SPAT Revolution Ultimate* or **Essential**. The license will be automatically placed on your iLok account.

Now open the iLok License Manager app and activate your license by dragging and dropping it to your computer. You can then close the iLok app and open the FLUX:: Center.

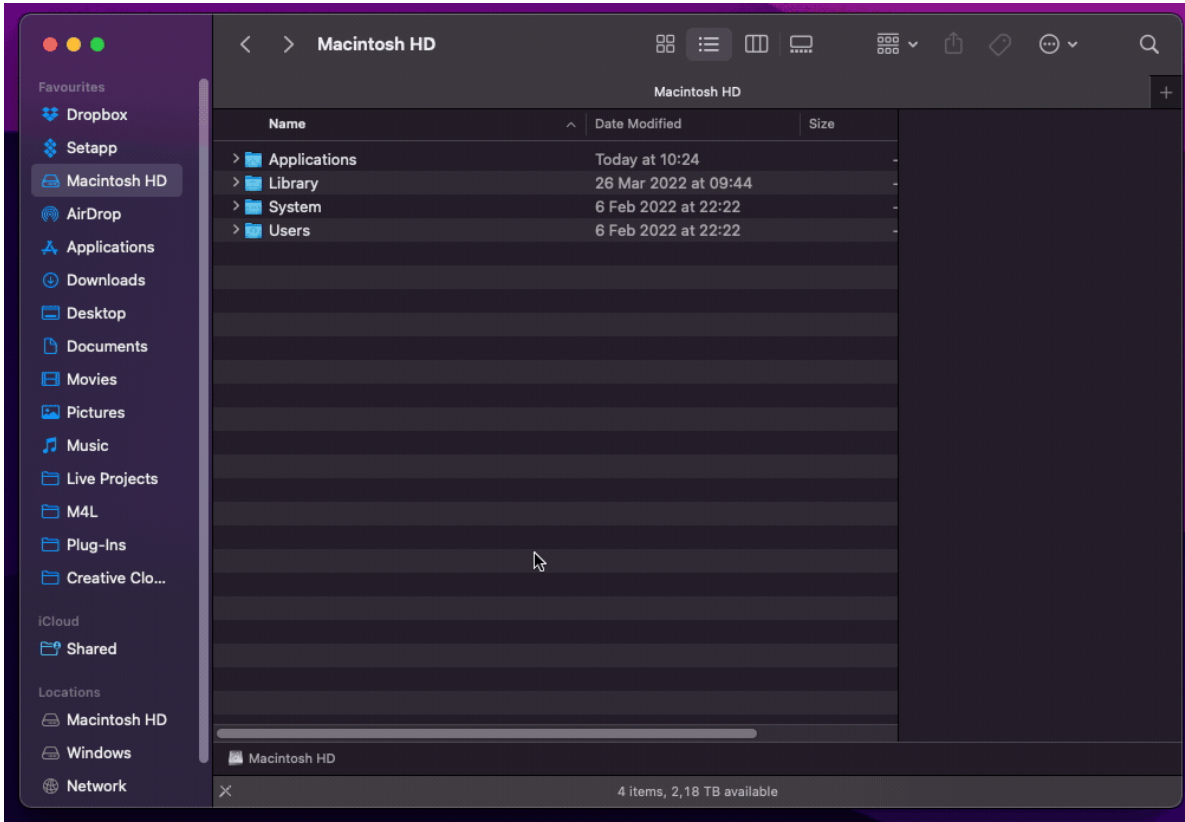
Connect to your FLUX account and download the *SPAT Revolution*; when you download the SPAT Revolution, the plugins and the Ableton Live Tools for SPAT Revolution will also be automatically installed.

44.3.1 Tools

macOS

Once the SPAT Revolution and the Ableton Live Tools for SPAT Revolution have been installed from the FLUX:: Center, you can look for them in the following folder /Library/Application Support/Flux/Ableton Live Devices/SPAT

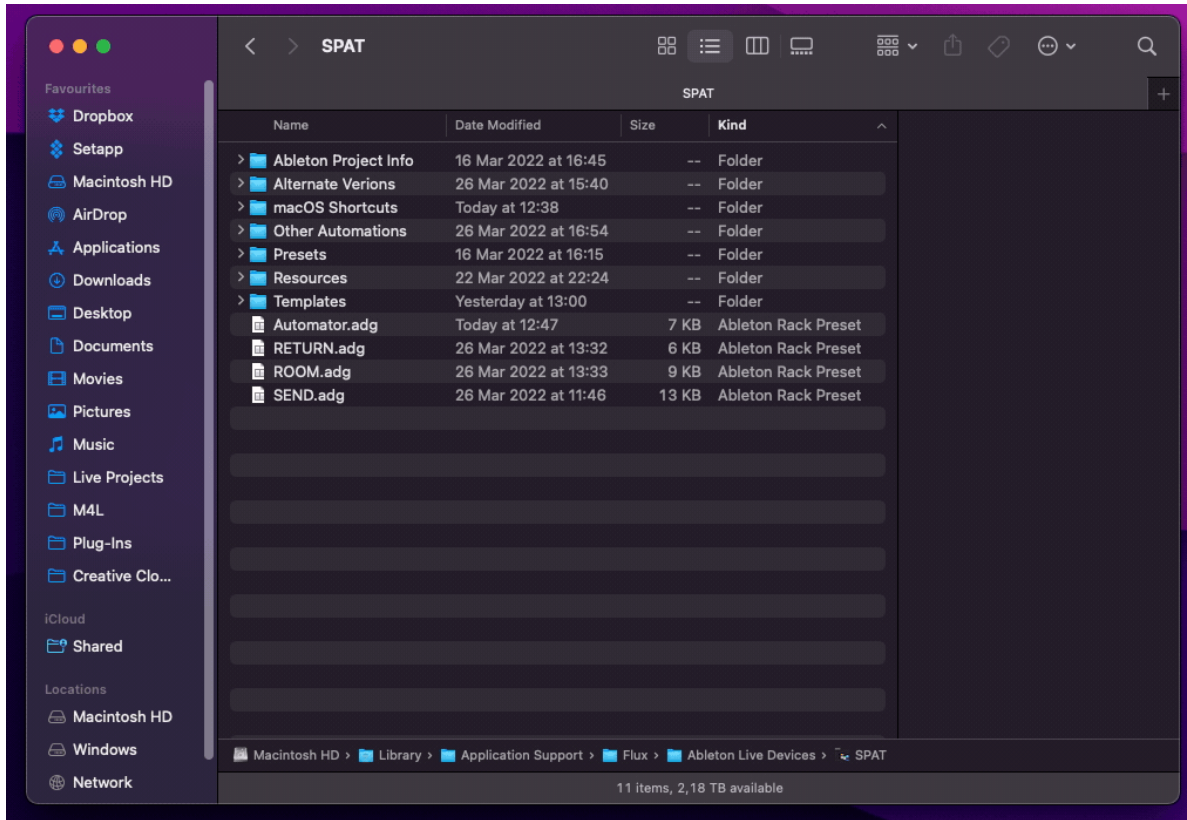
It is possible to go to the folder location by opening the **Finder** and, using the shortcut: **CMD + Shift + G** and copying the previous address.



Warning

Please do not change the location of this folder.

To go faster later, you can add this folder to your Finder's sidebar:



Windows

If you are on Windows and want to use the devices, please download the archive from this link: [download](#)

You can place it anywhere on your computer and then add the folder to your Ableton Live using the Add folder in the sidebar of Live.

44.3.2 Virtual Audio Driver

For this workflow to work, we must get some Virtual Audio Drivers to route audio from Live to SPAT Revolution.

For our Virtual Audio Driver, **BlackHole** from **Existential audio** is my preferred choice as it is simple to install, has many options in terms of channel count, is free, and is open-source!

You can also have all the different versions (*2ch/16ch/64ch*) installed simultaneously and have a different use for them simultaneously. We will see later in the guide that there are many

ways this can be very helpful and how to take advantage of that, So my advice is to install them all.

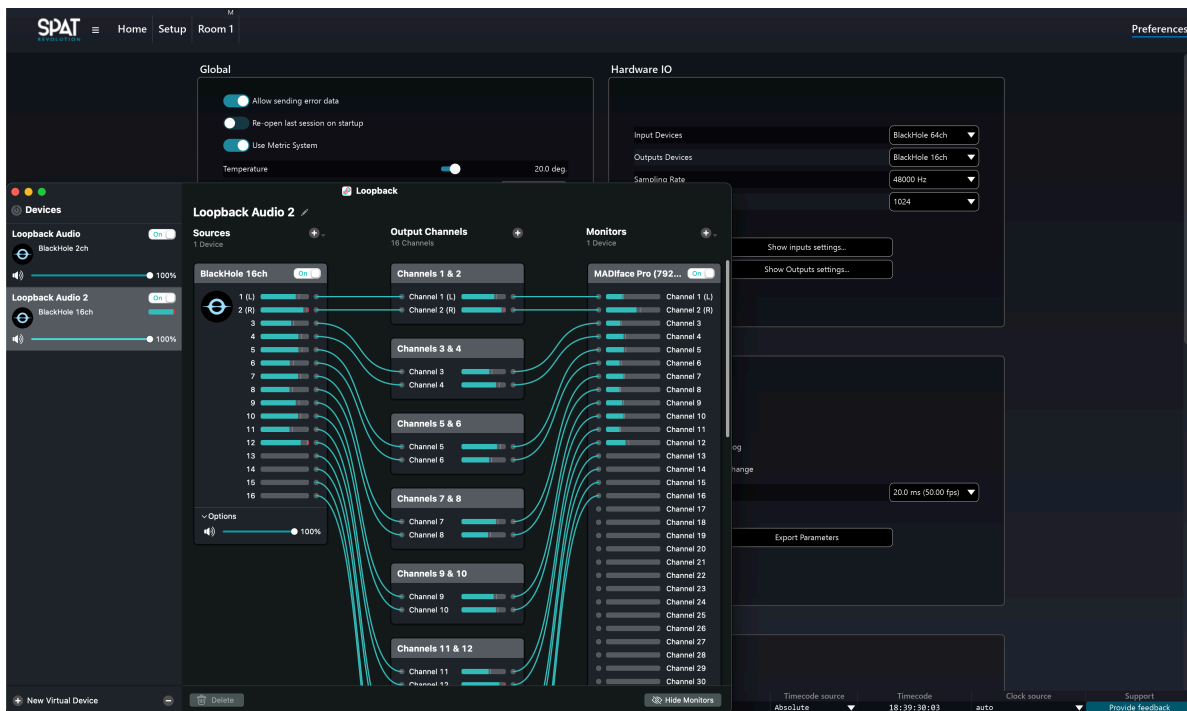
[BlackHole virtual audio driver can be found here](#)

i Note

If you plan to have a dual computer setup, you don't have to get BlackHole. More on that workflow later.*

44.3.3 Useful addition

Software such as **Loopback** from **Rogue Amoeba** can be handy to route audio between multiple devices/apps and can be an excellent alternative to Aggregate and Multi-Output devices that you can already make inside the **Audio MIDI setup** of macOS.



! Warning

Don't forget to restart your Mac after installing all that.

44.4 Optimization

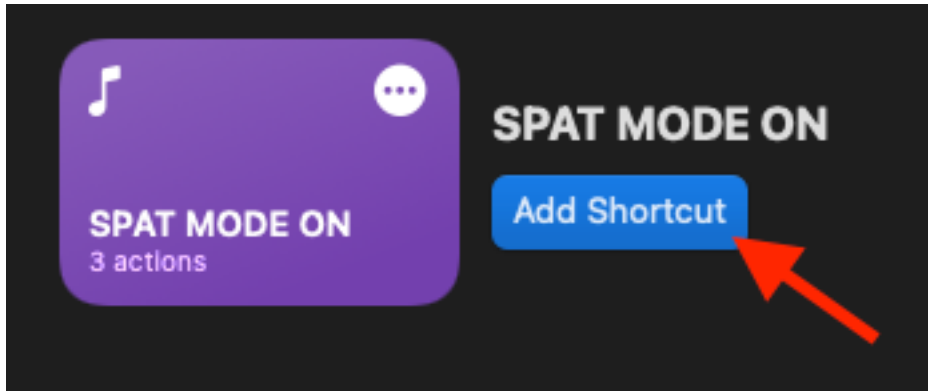
Now that everything is installed, let's optimize a few parameters in your macOS before getting into the setup.

When working with *SPAT Revolution*, and this can extend to any other audio work, you do not want any interference or background apps refreshing that would consume even the tiniest bit of CPU, so disabling Bluetooth, Wi-Fi as well as AirDrop is a great way to start and very simple to do.

You can simplify it using the provided *Shortcuts package* we created to do just that. They are located in the same folder as the other device so: `/Library/Application Support/Flux/Ableton Live Devices/SPAT/macOS Shortcuts`

What are Shortcuts?

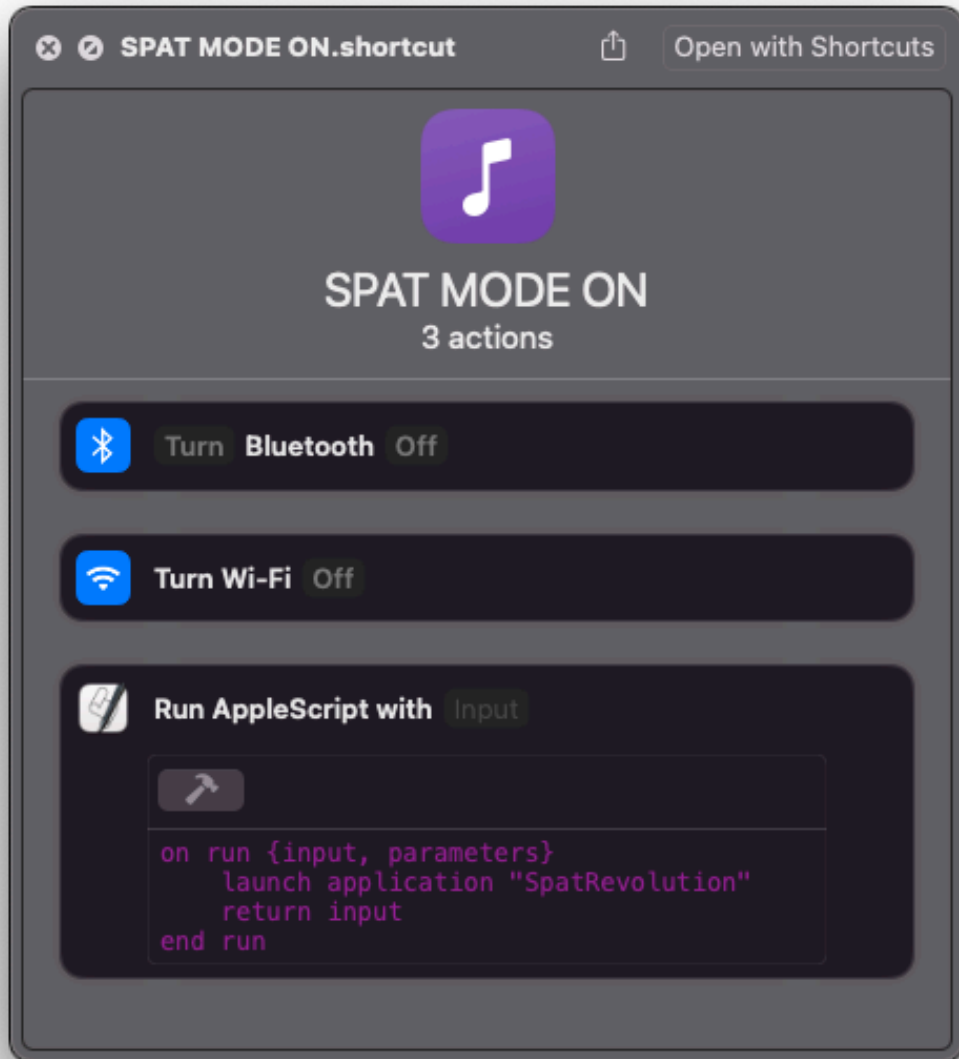
Shortcuts is simply an app (*native in macOS*) that lets you combine multiple steps across multiple apps to create powerful task automation. To add the provided Shortcuts, double-click each of them and select the **Add Shortcut** button.



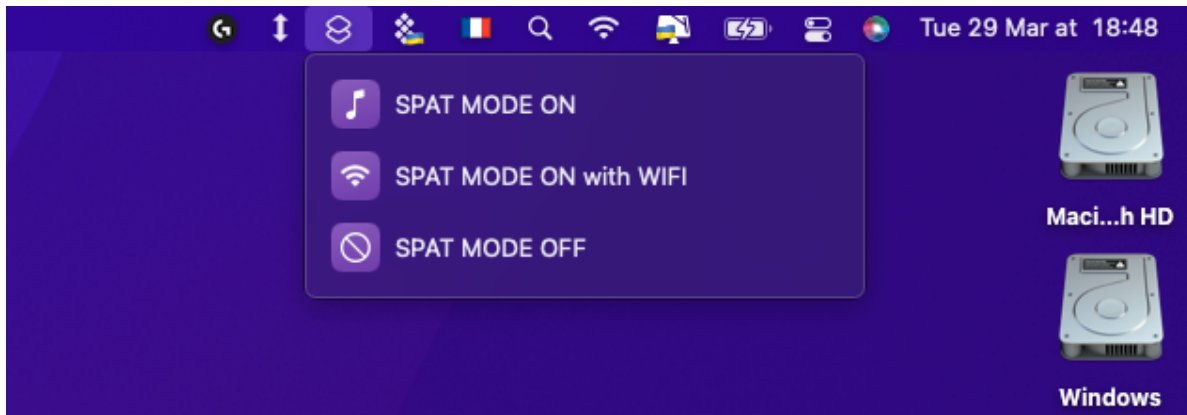
You will have three shortcuts available:

- **SPAT MODE ON:** will disable WI-FI, Bluetooth, AirDrop, and start-up SPAT Revolution if it is not already running.
- **SPAT MODE ON with WI-FI:** all the same as the previous one but will let the Wi-Fi on if you plan to use OSC control devices via Wi-Fi.
- **SPAT MODE OFF:** will start back the WI-FI, Bluetooth, and AirDrop.

These shortcuts are very simple to make using the Shortcuts app inside macOS if you want to modify them or make them yourself:



Once added, they should appear in your top bar for easy access:



Of course, you do not have to do this at all; everything else should work properly anyway, this is to save you a bit of CPU.

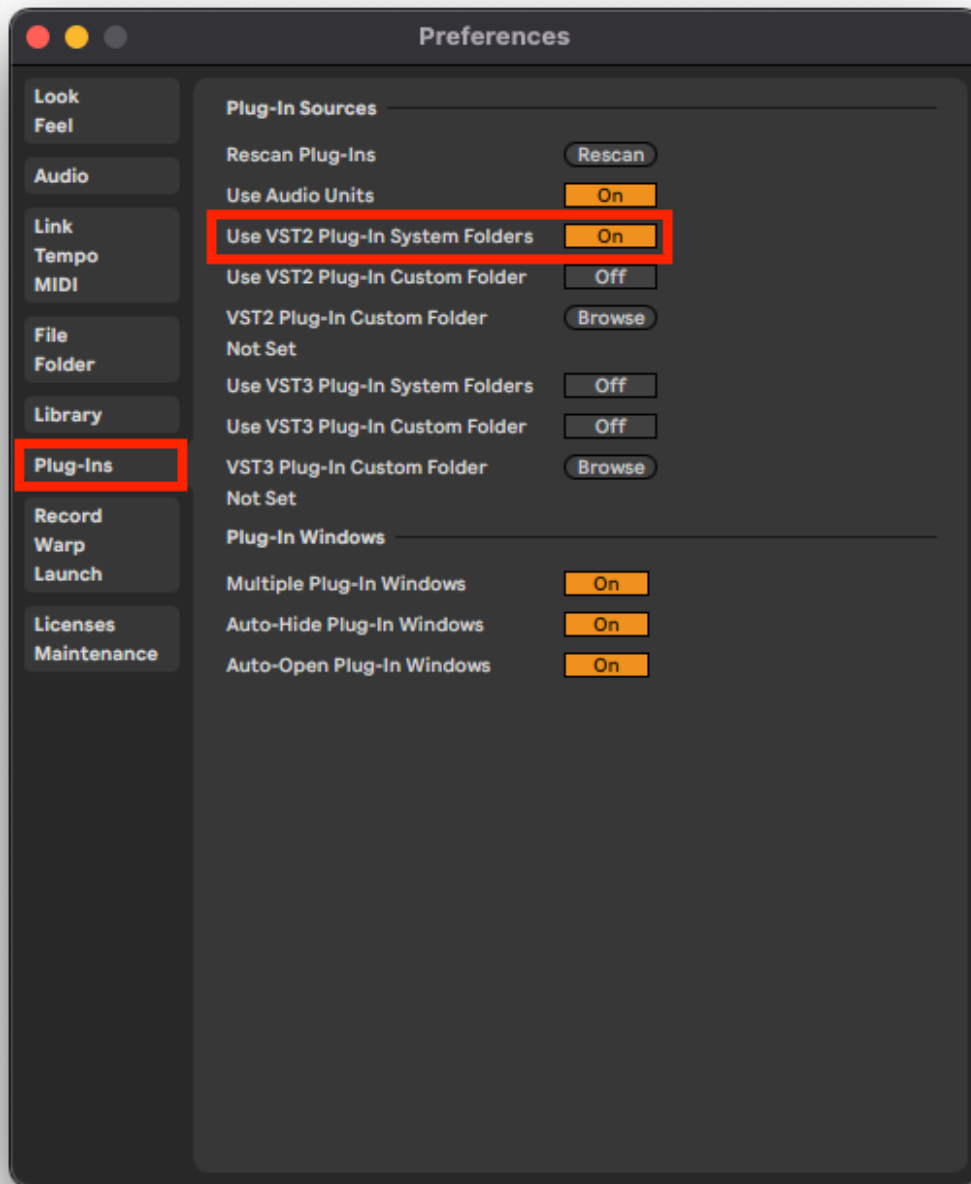
44.5 Setup

44.5.1 Ableton Live

Before getting into the workflow, we must set up just a few things inside Live and SPAT Revolution.

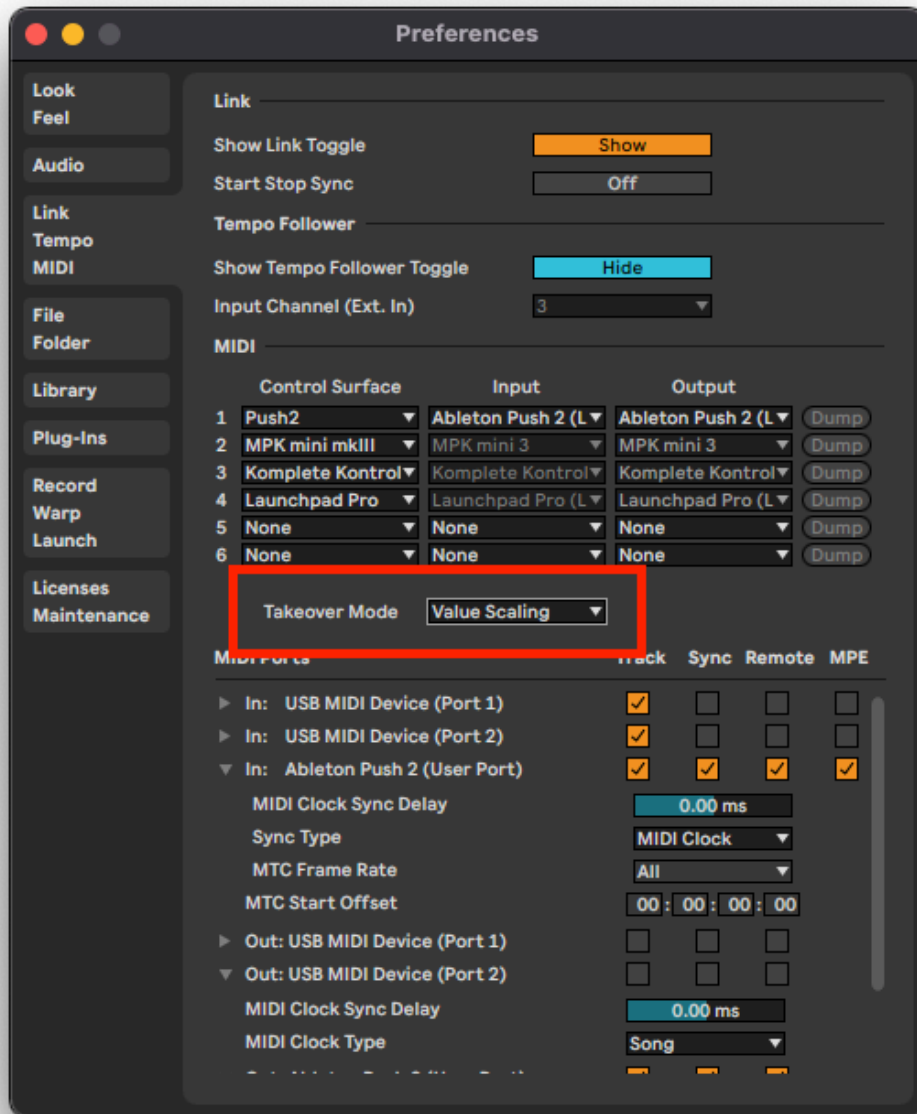
So open Ableton Live and configure the following:

- Make sure Live is scanning for VST2 plugins

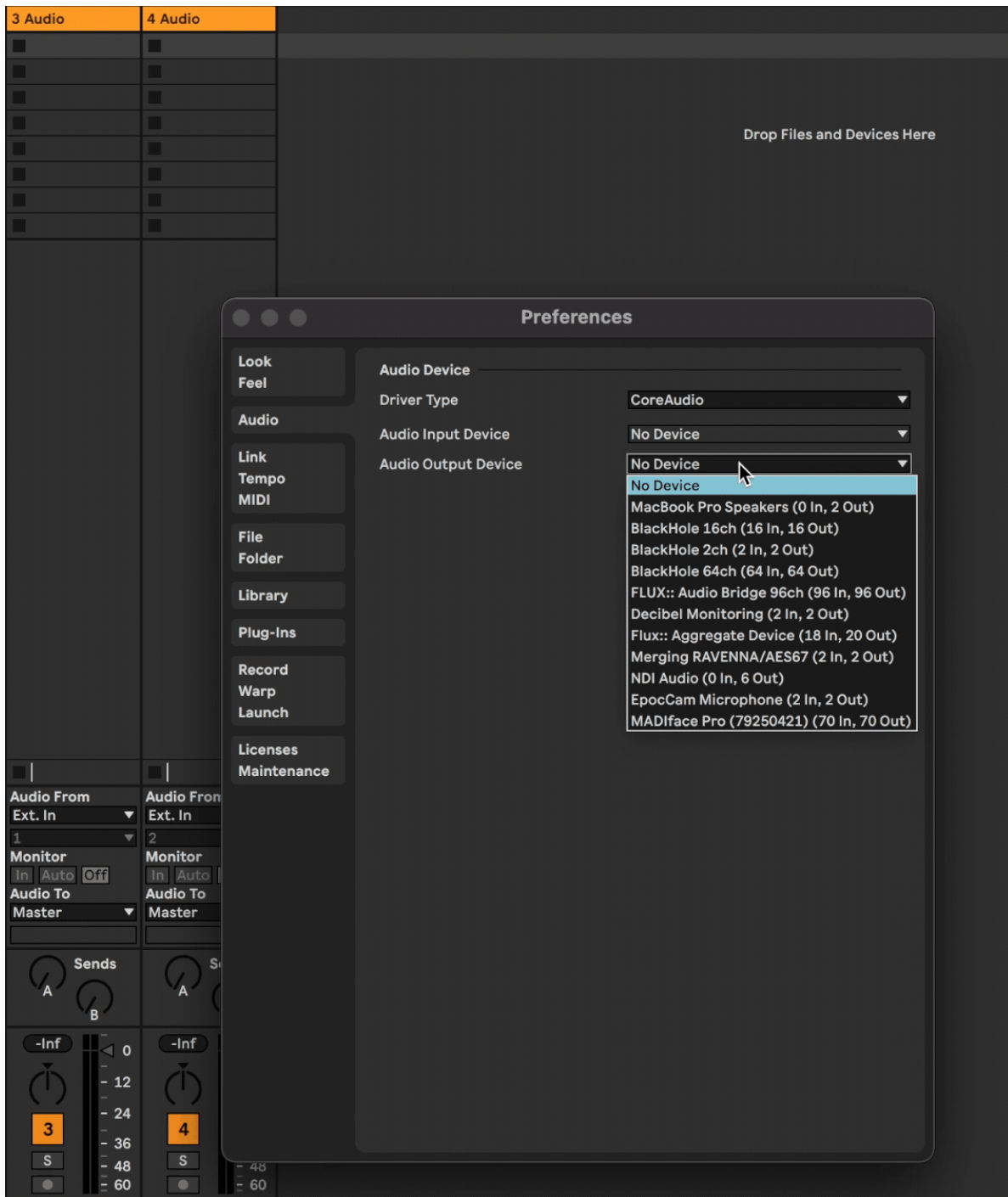


Warning

If you work with a MIDI controller, we also highly recommend that you go into the Live Preferences, into the MIDI tab and change the Takeover Mode to Value Scaling so that parameters you play with don't get too jumpy.



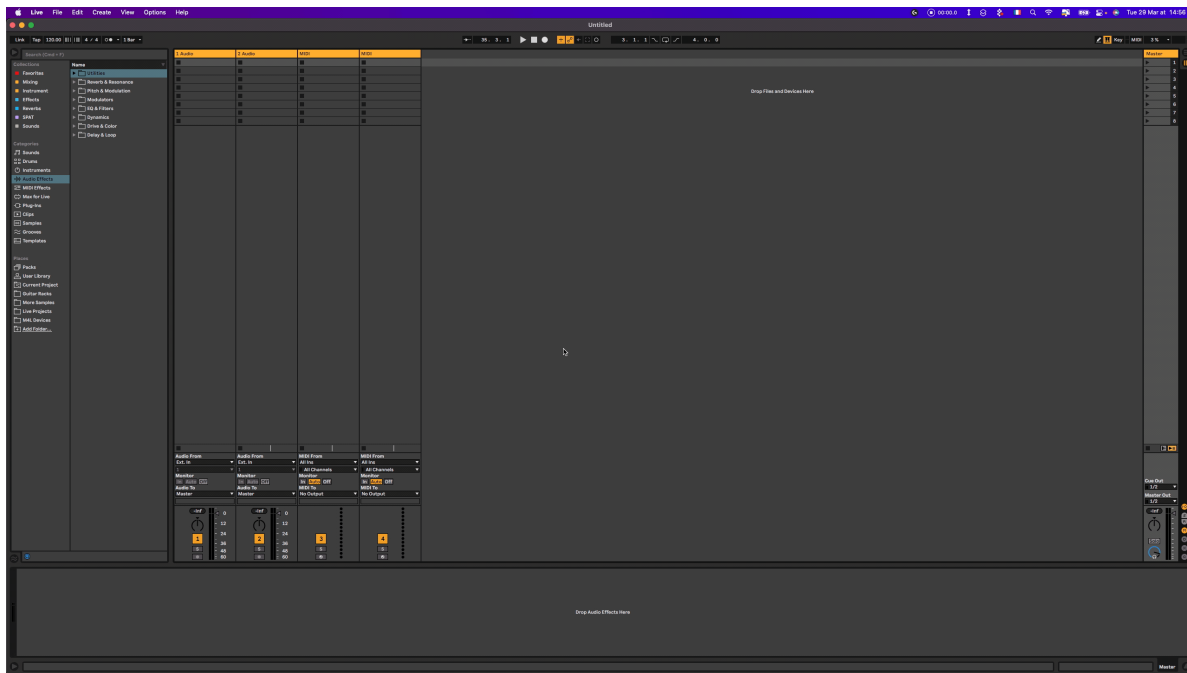
- Set BlackHole as your Output Device and enable all the channels of the driver:



You can do this part when you need more channels, the only advantage of doing it now is that

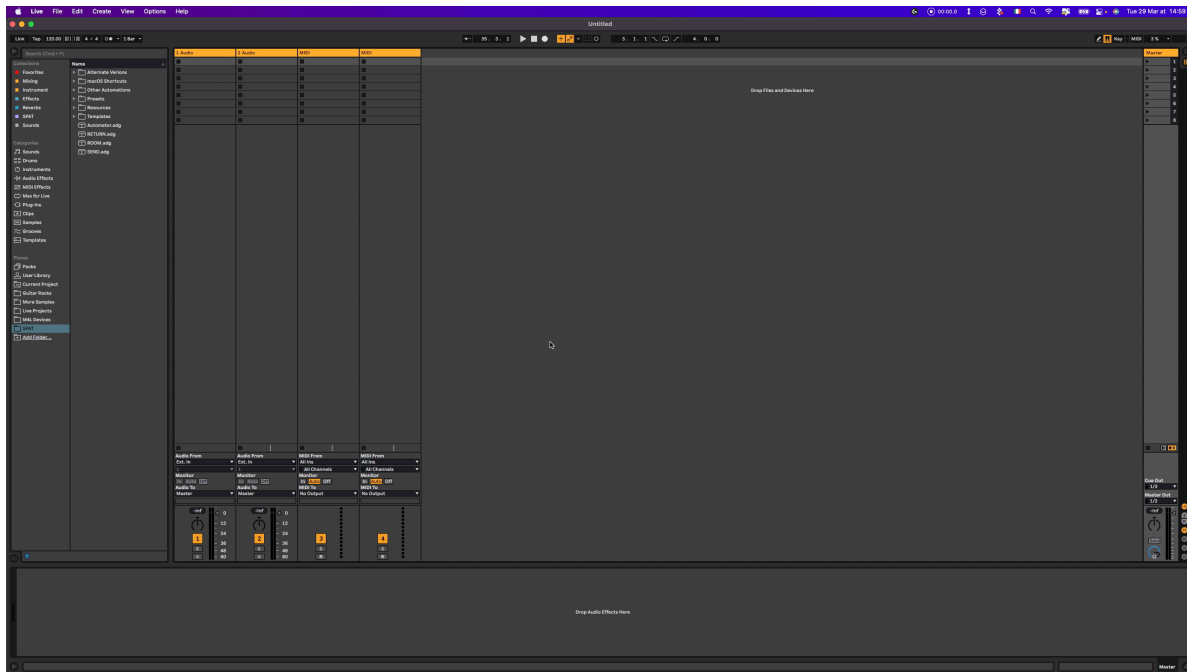
you don't have to do it later!

- Add the SPAT folder containing the Tools using the `add folder` in the sidebar of Live:



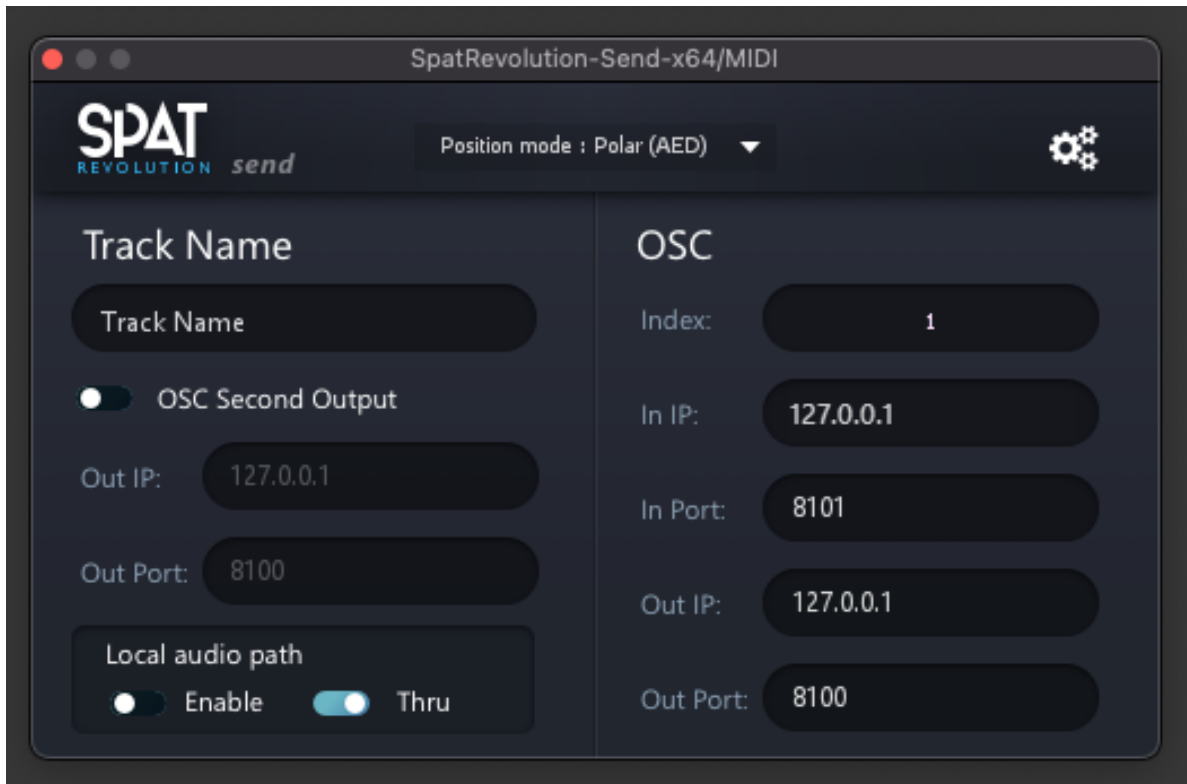
- Place on a track the devices `SEND`, `ROOM`, and `RETURN` located in the SPAT Folder:

The small windows popping up are the actual spat revolution plugins, on which the devices are based. They are here to host the parameters of the Sources (`SEND`), Room/Reverb (`ROOM`), and Master (`RETURN`) and send the automation data to SPAT Revolution using the OSC protocol.



Verify that they are all set on the following:

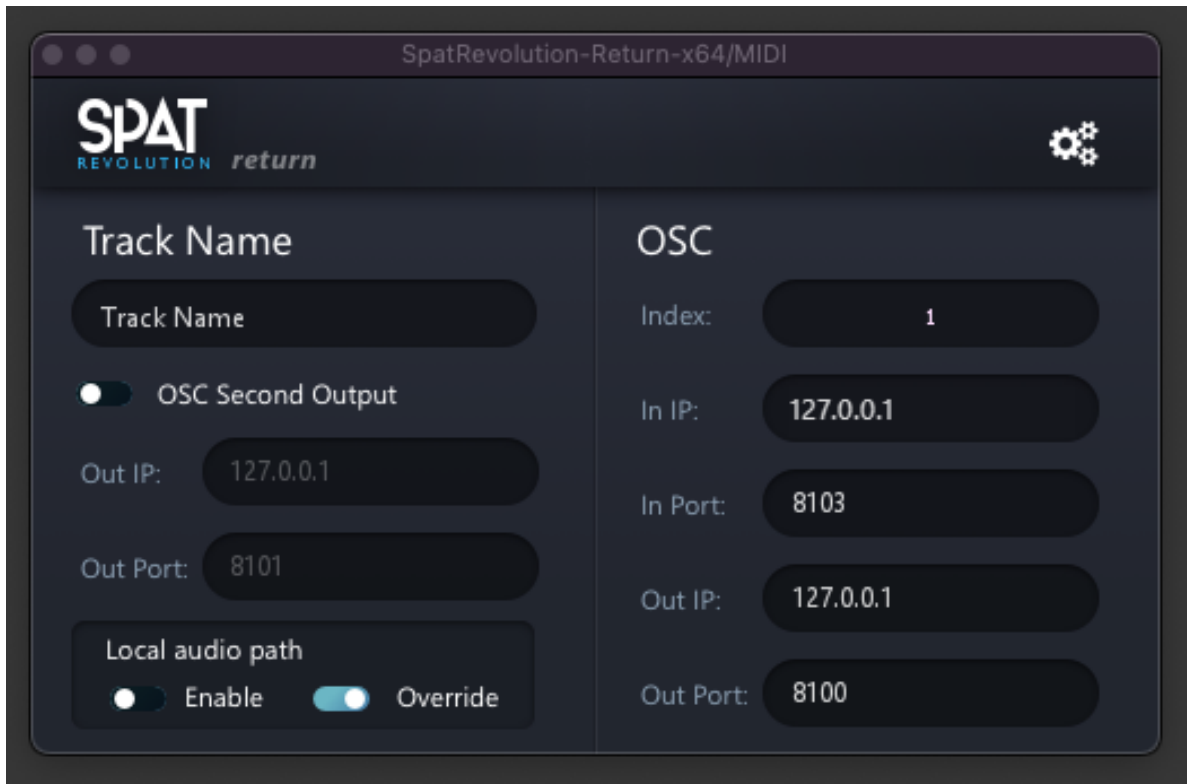
- In IP: 127.0.0.1 (for all)
- In Port (SEND): 8101
- In Port (ROOM): 8102
- In Port (RETURN): 8103
- Out IP: 127.0.0.1 (for all)
- Out Port: 8100 (for all)
- **SEND:**



- ROOM:



- **RETURN:**

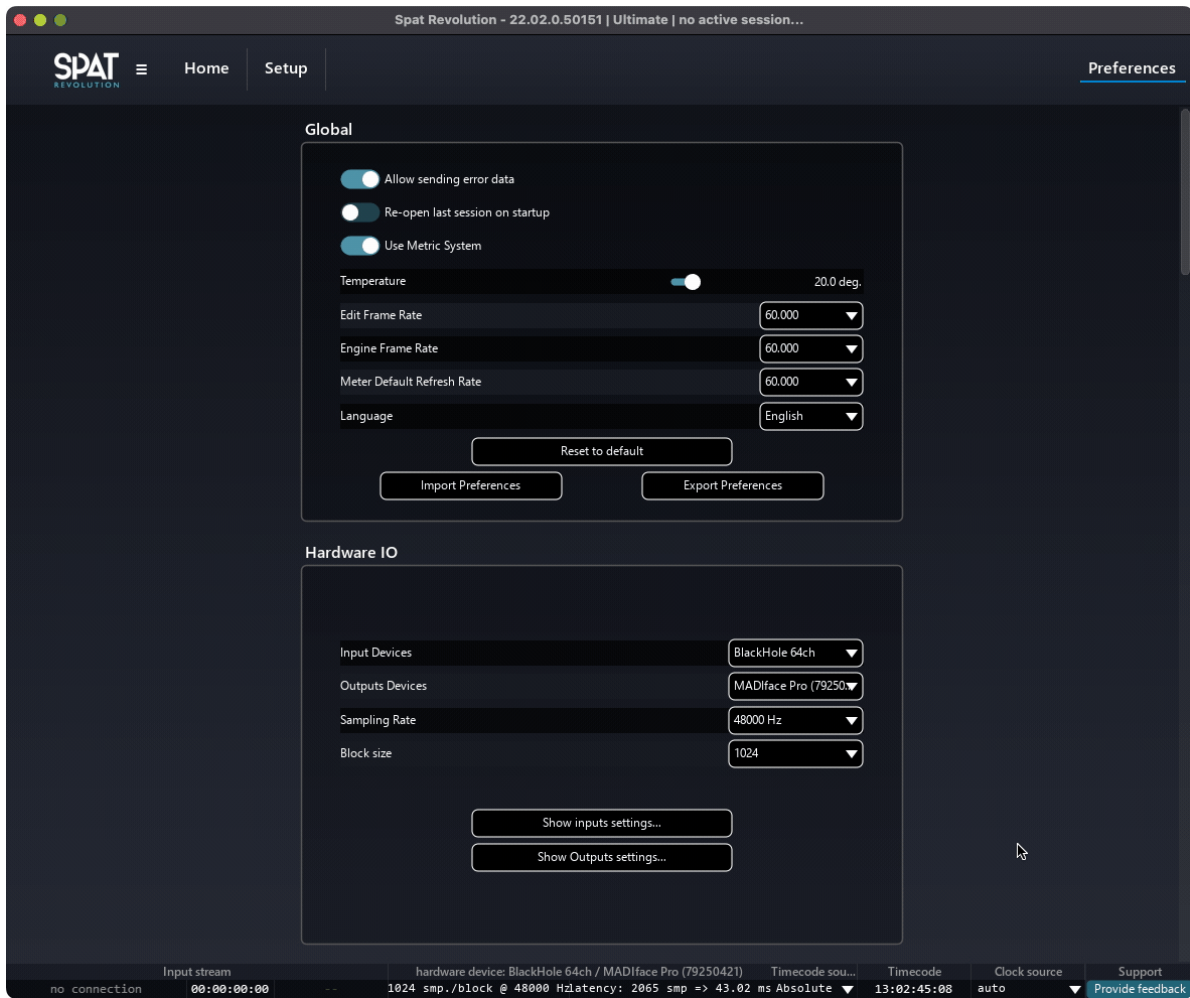


44.5.2 SPAT Revolution

- Open SPAT Revolution and go to the Preference panel to set BlackHole as the input device and your preferred Output device:



- Activate the OSC and set up the connection (find more about OSC configuration on the OSC documentation section [34.1](#)):



44.5.3 What is OSC?

Open Sound Control (OSC) is a protocol for networking sound synthesizers, computers, and other multimedia devices for purposes such as musical performance or show control. OSC's advantages include interoperability, accuracy, flexibility, and enhanced organization and documentation. The first specification was released in March 2002.

Here, we will use the OSC to send the automation information made in Live to SPAT Revolution.

Want to learn more about OSC? wikipedia.org

44.6 Devices Description

None of those devices add any latency or modification to the audio being passed through them, so do not worry about adding them anywhere in your chain!

44.6.1 SEND

This device lets you control some of the most valuable parameters inside SPAT for the currently selected source.

You must set up the OSC connection between Ableton Live and SPAT Revolution. The default settings of the plugins should already be set to connect with SPAT Revolution.

Please check the setup part of the guide to learn more about the OSC Connection.



1. **DEFAULT:** Reset the parameters of the source to the default settings.
2. **Gain:** Configure the gain of the source.
3. **Azimuth:** Configure the angle between the source location and the center reference point.
4. **Elevation:** Configure the elevation angle of the source.
5. **Distance:** Configure the distance from the source to the center reference point.
6. **Room Presence:** Configure the prominence of the reverberation of the source.
7. **Presence:** Configure the prominence of the direct sound.
8. **Warmth:** Configure the presence of the low-frequency content part of the source.
9. **Brilliance:** Configure the presence of the high-frequency content part of the source.
10. **Enable:** Mute or unmute the currently selected sources inside SPAT Revolution.
11. **Reverb:** Enable or disable the reverb inside SPAT Revolution for the currently selected source.
12. **Doppler:** Activate the Doppler effect for the currently selected source.

13. **SpatRevolution-Send-x64:** The original plugin on which the device is based, you can still access the individual parameters by un-folding the device or accessing the GUI by clicking the minor key at the bottom.
14. **How to use:** This is a small notepad device with additional notes to prevent you from having to check back the guide if you forget something.
15. **Macro Controls:** Click this button to hide the encoder knobs of the device (*ref from n°2 to n°9*)
16. **Macro Variation:** Click this button to hide the macro variations (*ref to n°1*)
17. **Show/Hide Devices:** Click this button to hide everything after the encoder knobs. (*ref from n°10 to n°14*)
18. **Info View:** This opens/closes Live's built-in text info box, every parameter in the device has text info, so this box might be useful if you need to know which does what.

Additional Information

- **Automations:**

Note that Ableton Live takes over the automation parameters inside the DAW, so you can move the source inside SPAT Revolution without the parameter moving inside Ableton Live. To fix this simply, right-click on the parameter in question and select **Remove Mapping to Send**.

- **Moving the device in Live:**

Moving this device to another track instead of deleting and creating a new one will un-map the Enable, Reverb, and Doppler buttons. Deleting the device first is safer, and then placing a new one on the new track. It also could help you prevent any OSC Index-related issues.

Duplicating the track will result in the same issue. So you should manually place the device on each new track you create, although you can still duplicate your track (if needed) and simply delete the device and place a new one on the track.

- **Undo/Redo:**

Usually, in SPAT Revolution, we currently cannot do undo/redo action, but Live with this device works! As long as the action you are backtracking has been operated inside Ableton Live directly.

- **SEND - LAP:** (available in the Alternate Versions folder)

The LAP version is exactly the same but is programmed to activate the Local Audio Path right of the start if you prefer working with this instead of virtual audio drivers.

- **SEND - Advanced:** (available in the Alternate Versions folder)

A more advanced version of the original device, with simply most of the available source settings inside SPAT Revolution.

- **SEND - LAP - Advanced:** (available in the Alternate Versions folder)

A combo of the two versions described above, so Local Audio Path activated and more source control parameters.

44.6.2 ROOM

This device allows you to control the reverb quickly; many more parameters will be available if you go directly into the Reverb settings inside your Room.

You must set up the OSC connection between Ableton Live and SPAT Revolution. The default settings of the plugins should already be set to connect with the SPAT Revolution.

Please check the setup part of the guide to learn more about the OSC Connection.



1. **DEFAULT:** Reset the Room's parameters to the default settings.
2. **Gain:** Configure the gain of the room.
3. **Size:** Change the reverb size value.
4. **Liveness:** Configure the relative decay time of high-frequency content, relative to the reverberance.
5. **Cross HiFreq:** Configure the frequency above which the Liveness setting determines decay time.
6. **Reverb Gain:** Configure the reverb gain.
7. **Reverberance:** Change the reverberance value.
8. **Heavyness:** Configure the relative decay time of low-frequency content, relative to the reverberance.
9. **Cross LowFreq:** Configure the presence of the high-frequency content part of the source.
10. **Reverb:** Completely disable the reverb for the currently selected Room.

11. **Enable:** Mute or unmute the currently selected room.
12. **Infinite:** Toggle the infinite more of the reverb, be careful with your gain when using this one.
13. **SpatRevolution-Room-x64:** The original plugin on which the device is based, you can still access the individual parameters by un-folding the device or accessing the GUI by clicking the small key at the bottom.
14. **How to use:** This is a small notepad device with additional notes to prevent you from having to check back the guide if you forget something.
15. **Macro Controls:** Click this button to hide the encoder knobs of the device (*ref from n°2 to n°9*)
16. **Macro Variation:** Click this button to hide the macro variations (*ref to n°1*)
17. **Show/Hide Devices:** Click this button to hide everything after the encoder knobs. (*ref from n°10 to n°14*)
18. **Info View:** This opens/closes Live's built-in text info box; every parameter in the device has text info, so this box might be useful if you need to know which does what.

Additional Information

- **Automations:**

Note that Live takes over the automation parameters inside the DAW, so you can move the source inside SPAT Revolution and without the parameter moving inside Ableton Live. To fix this, simply right-click the parameter in question and select **Remove Mapping to Send**.

- **Moving the device in Live:**

Moving this device to another track instead of deleting and creating a new one will un-map the Enable, Reverb, and Infinite buttons. Deleting the device first is safer, and then placing a new one on the new track. It also could help you prevent any OSC Index-related issues.

Duplicating the track will result in the same issue. So, if you have more than one Room, you should manually create a new track and place the ROOM device.

- **Undo/Redo:**

Usually, in SPAT Revolution, we currently cannot do undo/redo action, but in Live with this device it does work! As long as the action you are backtracking has been operated inside Ableton Live directly.

- **ROOM - Advanced:** (available in the Alternate Versions folder)

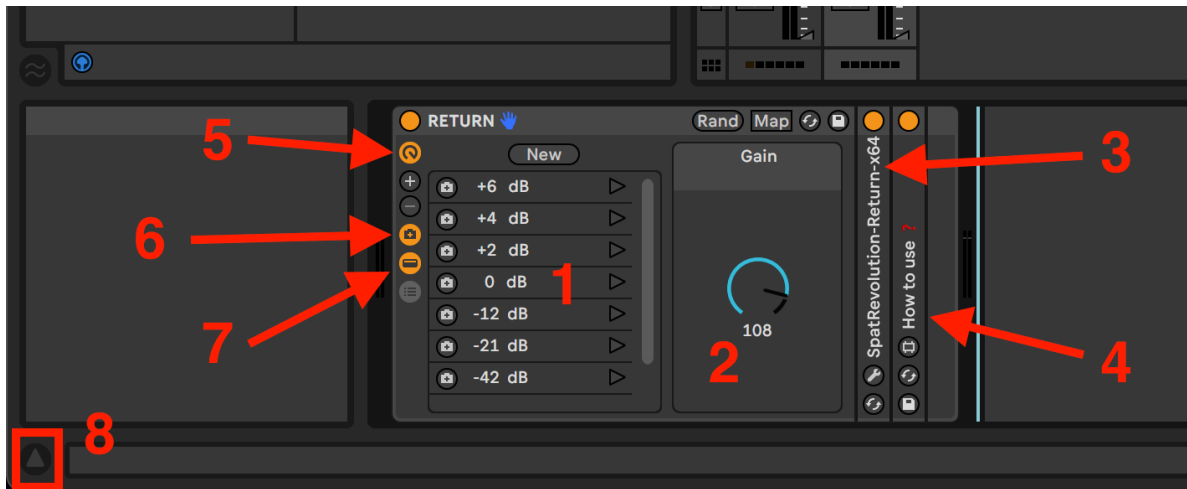
A more advanced version of the original device, with simply most of the available reverb settings inside SPAT Revolution.

44.6.3 RETURN

This device allows you to control the gain of the selected Master. You can find this block between the Room and the final output.

You must set up the OSC connection between Ableton Live and SPAT Revolution. The default settings of the plugins should already be set to connect with the SPAT Revolution.

Please check the setup part of the guide to learn more about the OSC Connection.



1. **Macro Variations:** Quick shortcuts to jump to specific values instead of using the encoder knob.
2. **Gain:** Set the gain of the currently selected Master.
3. **SpatRevolution-Return-x64:** The original plugin on which the device is based, you can still access the individual parameters by un-folding the device or accessing the GUI by clicking the small key at the bottom.
4. **How to use:** This is a small notepad device with additional notes to prevent you from having to check back the guide if you forget something.
5. **Macro Controls:** Click this button to hide the encoder knobs of the device (*ref to n°2*)
6. **Macro Variation:** Click this button to hide the macro variations (*ref to n°1*)
7. **Show/Hide Devices:** Click this button to hide everything after the encoder knobs. (*ref from n°3 and n°4*)
8. **Info View:** This opens/closes Live's built-in text info box; every parameter in the device has text info, so this box might be useful if you need to know which does what.

Additional Information

- **Automation:**

Note that Live takes over the automation parameters inside the DAW, so you can move the source inside SPAT and without the parameter moving inside Ableton Live. To fix this, simply right-click the parameter in question and select **Remove Mapping to Send**

- **Moving the device in Live:**

Unlike the Send and Room devices, which have specific M4L devices that can't be moved from one track to another without being unfortunately un-mapped, you can easily move this device around without any issues.

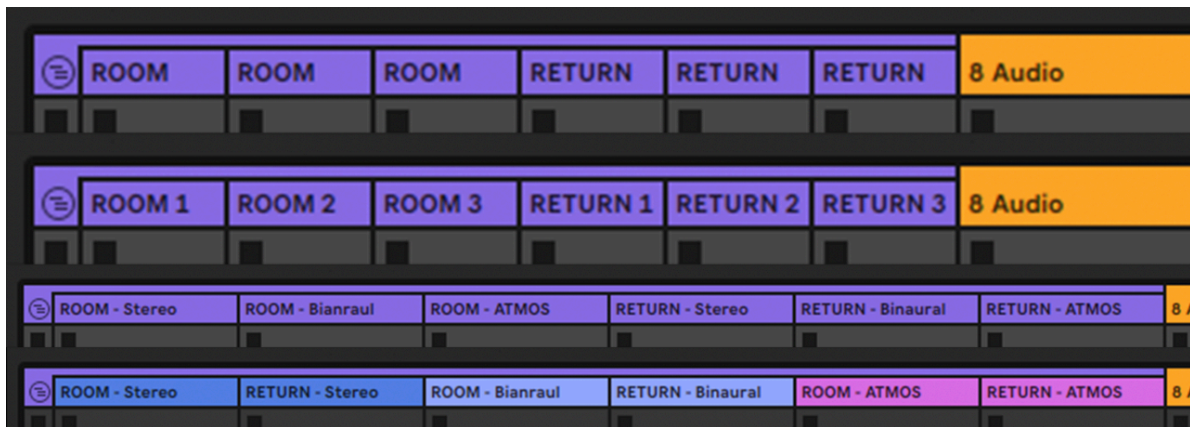
And again, unlike the two previously mentioned devices with the return, if you have more than one Master in your SPAT Revolution session, you can easily duplicate the track hosting the RETURN device, and it will automatically set up everything correctly.

- **Undo/Redo:**

Usually, in SPAT Revolution, we currently cannot do undo/redo action, but in Live with this device, it does work! As long as the action you are backtracking has been operated inside Ableton Live directly.

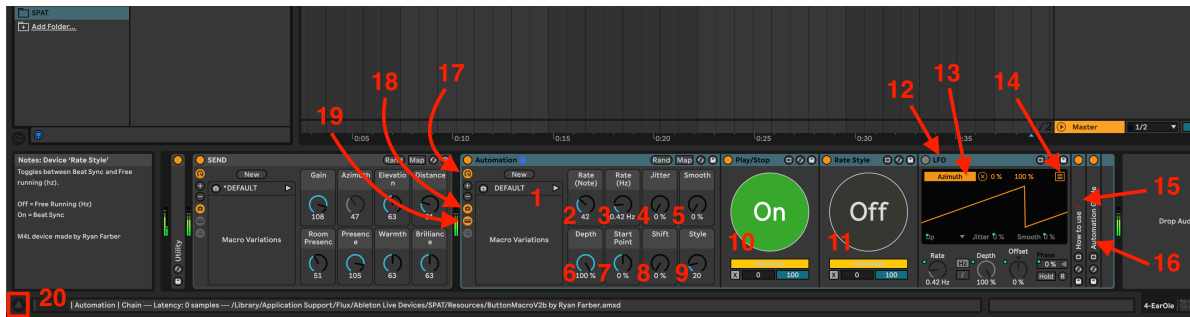
i Note

“Don't forget to rename your tracks for an easier read of your session!”



44.6.4 Automator

The Automator device allows you to make the selected source move around in SPAT Revolution in a more organic and continuous way compared to traditional automation.



1. **Macro Variations:** Quick shortcuts to reset to the device's default parameters, you can create new macros to change the automation of a source quickly.
2. **Rate (Note):** The base rate of the automation (expressed in note values).
3. **Rate (Hz):** The base rate of the automation (expressed in hz).
4. **Jitter:** Introduces some jitter in the automation signal.
5. **Smooth:** Smooth value changes (and jitter, as well).
6. **Depth:** Depth of the LFO automation.
7. **Start Point:** Shift the start point of the automation.
8. **Shift:** This allows you to shift the placement of the automation relative to other LFOs in Live (requires restarting the session to get all the LFO synced together).
9. **Style:** Change the automation style (Sine, Saw, Square, etc.).
10. **Play/Stop:** Play or stop the automation.
11. **Rate Style:** Toggles between Beat Sync and Free running (Hz).
12. **LFO:** The LFO generates automation on the primary SEND device; click the Map button, and then select the Azimuth parameter.
13. **Map:** Click this button to activate the Map Mode; you can then click the parameter in the SEND device that you wish to automate.
14. **Multimap:** This is the Multimap browser; it can allow you to map the same automation to multiple parameters. You can then tweak each one separately using the Min/Max percentage.
15. **How to use:** This is a small notepad device with additional notes to prevent you from having to check back the guide if you forget something.
16. **Automator Guide:** Same as the “How to use”, this one contains specific information about the different types of automation you can generate with the different LFO shapes.
17. **Macro Controls:** Click this button to hide the encoder knobs of the device (*ref from $n^{\circ}2$ to $n^{\circ}9$*)
18. **Macro Variation:** Click this button to hide the macro variations (*ref to $n^{\circ}1$*)
19. **Show/Hide Devices:** Click this button to hide everything after the encoder knobs. (*ref from $n^{\circ}10$ to $n^{\circ}16$*)
20. **Info View:** This opens/closes Live's built-in text info box; every parameter in the device has text info, so this box might be useful if you need to know which does what.

Additional Information

- **Moving the device in Live:**

Moving this device to another track instead of deleting and creating a new one will un-map the Play/Stop and Rate Style buttons, it is safer to delete the device first and then place a new one on the new track.

44.7 Workflow

i Note

Watching the corresponding video is highly recommended for this part.

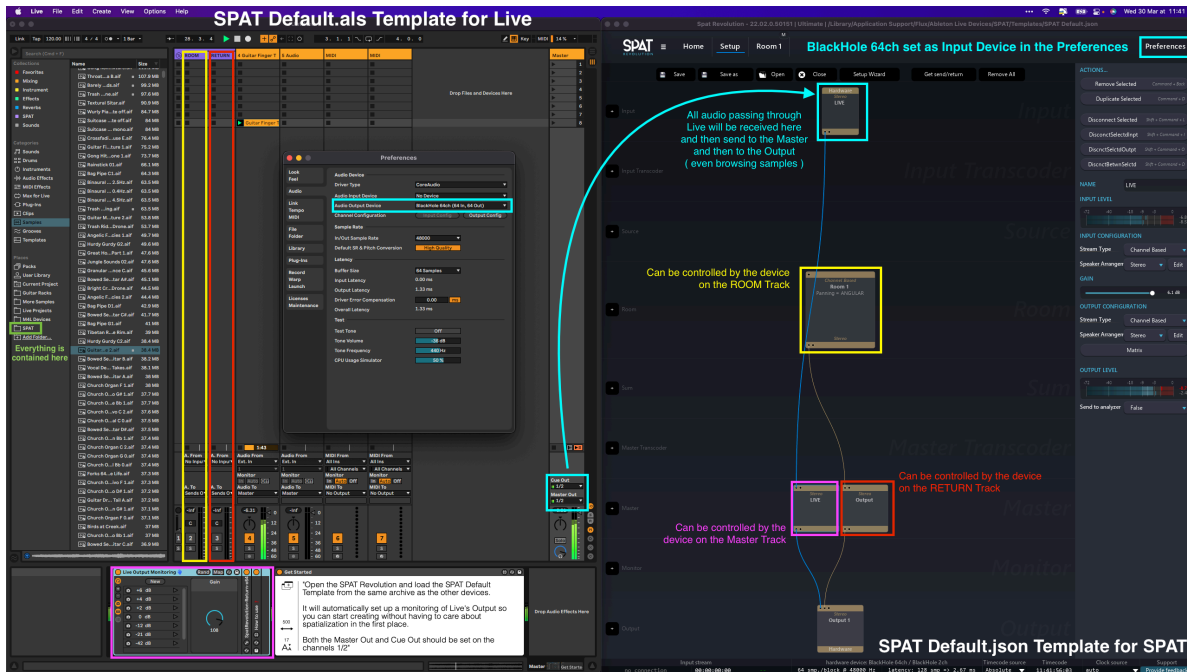
<https://www.youtube.com/embed/YYcnobAmRFk>

Now that we are entirely setup, the formula is straightforward:

- Optional Step: Use the SPAT MODE ON shortcut to turn off your external connections
- Open Live and select the “SPAT” folder in the Sidebar of Ableton Live
- Open the “SPAT Default.als” Template
- Open SPAT Revolution and load the Template “SPAT Default.json” from the same archive

If you have lost the location of the archive, it's here: /Library/Application Support/Flux/Ableton Live Devices/SPAT/Templates

With both templates open, you will have a direct connection between Ableton Live's output and your physical output so that you can start making music without thinking about spatialization in the first place.



make sure your “Cue Out” and “Master Out” are both set to 1/2, that your Output device in Live is “BlackHole 64ch” and that your Input device in SPAT Revolution is “BlackHole 64ch” as well, and the device controlling the output Level of Live directly is placed on the Master Track in Live

You can also note that the Ableton Live Template will set up a small group call “SPAT CTRL” which already contains a ROOM device and RETURN device, each on individual tracks for better visibility.

The ROOM device controls the room gain and reverb, and the RETURN controls the master volume.

- You can change the space you are working with by changing the reverb using the device on the ROOM track inside Ableton Live and by clicking Reverb on the Room page of SPAT Revolution.

This will change the virtual space sonority but not what your actual output is; you can change your speaker arrangement or stream type by going back to the setup page, selecting the Room block, and you will have the option to change the Stream Type, Speaker Arrangement (you can make or re-create your own), and the Panning Type used.

- Now, once you want to convert a track to an object in SPAT Revolution, start by attributing a channel to the track in Ableton Live.

Depending on which type of sound you are going to spatialize, you have the choice between Stereo Tracks and Mono Tracks. We would, for example, attribute a Kick to a single channel

as we want to keep it Mono, a pad to a stereo, and so on. Note that by default, an input block in SPAT is Mono

- Then go back to the SPAT Revolution Setup page and add an Input Block
- select the same channel(s) you just attributed in Live and connect it to your Room
- Now you can go to the Room page and see your object in the 3D view of SPAT Revolution
- From there, you can double-click the Source and Start playing with the parameters that appear before you.

You can also click the ball in the 3D view and move it around in the Space

- You can also add the “SEND” device from our “SPAT” folder in the sidebar of Live to Control the source directly from the DAW.

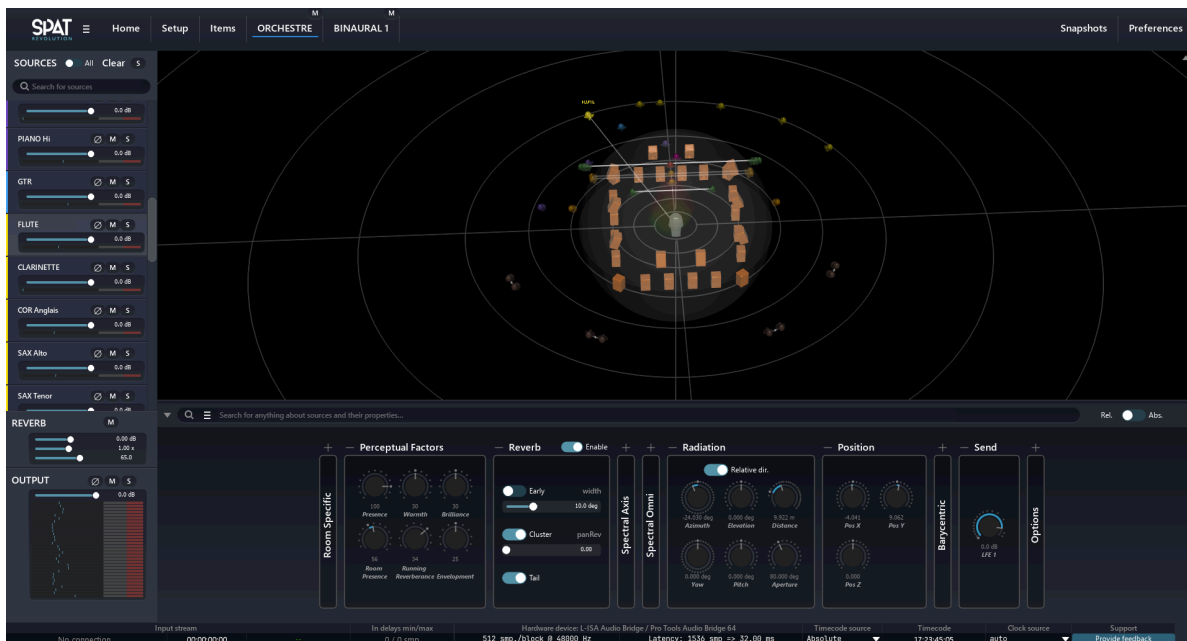
When adding a “SEND” device to your track, the UI of the original send plugin will open. You can take the time to rename the track in the plugin before closing the GUI so that the name translates automatically into SPAT Revolution.

- You can also start automating it using traditional automation or the “Automator” device in the same folder as the other devices.

<https://www.youtube.com/embed/xarMJBSSwU0>

direct link if not loading: <https://youtu.be/xarMJBSSwU0>

Happy Immersive Creating!



44.8 Printing an Immersive Mix

Note

This part is also available as a YouTube Video
<https://www.youtube.com/embed/1OA86iQuxls>

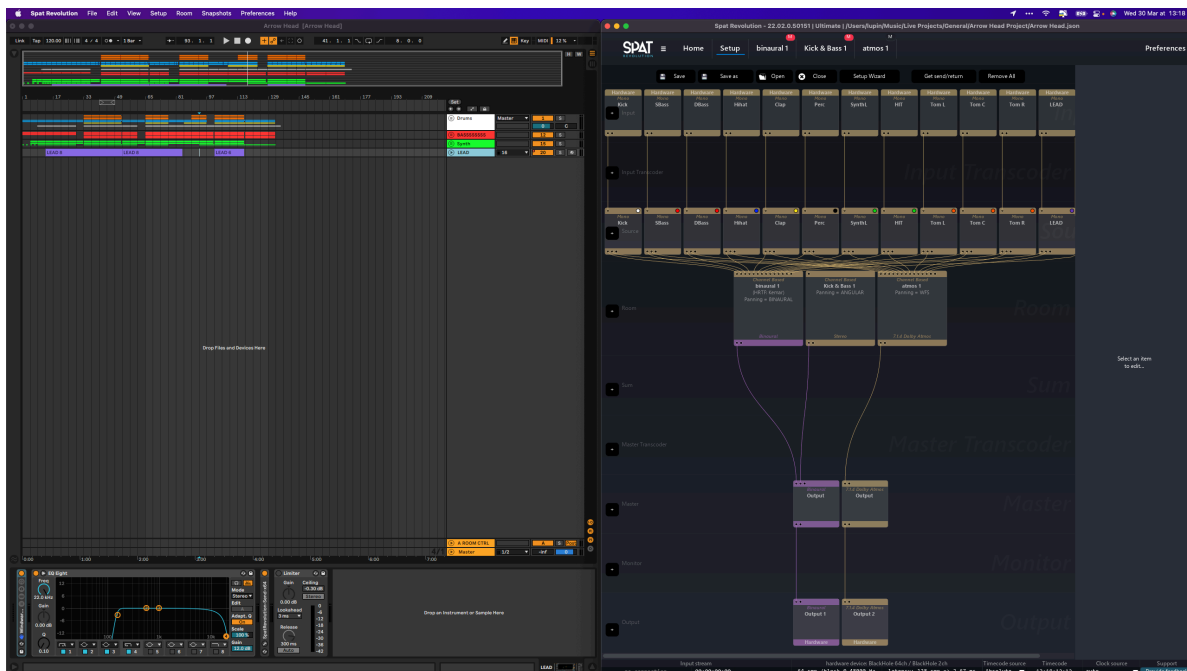
Now comes the time to think about how to render what we made inside Ableton Live and SPAT Revolution.

There are a few strategies about this, which have multiple solutions, and a few factors to consider like the number of Room, Return, Channels, etc...

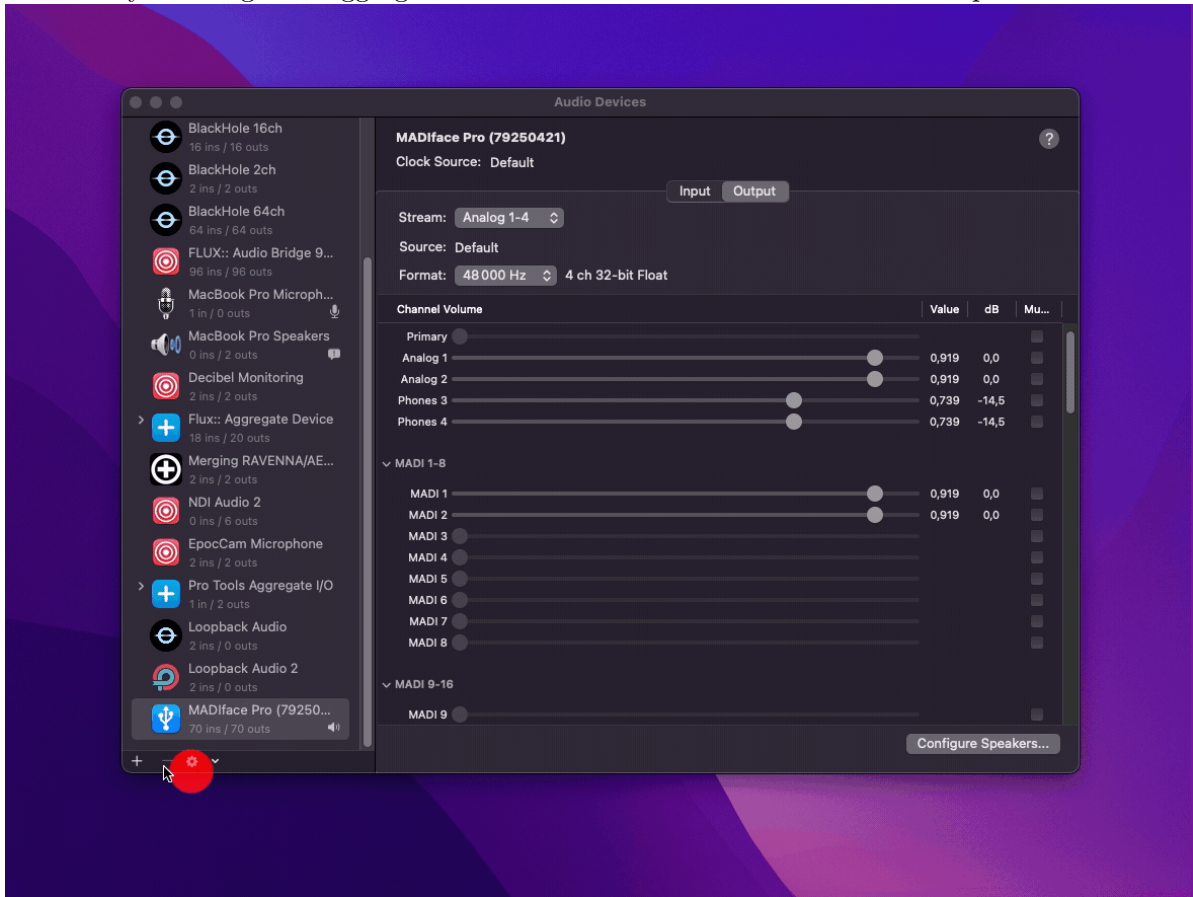
But the absolute main thing we cannot avoid is that we need to render real-time everything we made.

Let's have a use case to demonstrate how you can print our mix.

Here is a session that we will render for the Binaural (*3d on headphones*) Version and, at the same time, generate a Bed for an Atmos render later.



We will start by creating an “Aggregate Device” in the “AUDIO MIDI SETUP” panel of ma-



cOS:

Warning

Both Ableton Live and SPAT Revolution must be relaunched to detect the newly created Aggregate.

Then we will use this device as Output in SPAT Revolution and Input in Ableton Live, set the out channels of the binaural output block to the two corresponding channels of my 2ch device and the rest to the Atmos output.

Then the only thing left to do is to create the tracks in Live that will receive the audio coming out of *SPAT Revolution*, routing them accordingly, press record and let the thing play!

https://www.youtube.com/embed/FBz65vx_xJM

44.9 From Single Machine to Dual Computer Setup

i Note

This part is also available as a YouTube Video here: <https://youtu.be/TnZSjCVULIQ>
<https://www.youtube.com/embed/TnZSjCVULIQ>

The process of switching this workflow from a single computer to a dual one is pretty simple.

But why? If you have a heavier Ableton Live Session and want to reduce the load on your computer, that is often the why here.

We will need two things to make it possible, besides having two computers of course.

- a network audio solution such as MADI, DANTE, RAVENNA, etc...
- and a network switch to send the OSC automation from the computer hosting Live to the one with SPAT Revolution

i Note

We could do it via a Wi-Fi network, but there are better things in terms of speed and stability.

So now, let's change the Output device in Ableton Live for our network audio solution and do the same for the Input device in SPAT Revolution.

i Note

Check that all the channels are active in Live

Depending on which Network Audio Solution you use, you might need to re-route some of your channels, as output one will not always mean it will go into input one.

And now we need only to set up the new IPs as we are not sending the OSC automation locally.

Simply go into the SPAT Preferences and change (in the OSC Connections part) the plugin Outputs to the IP address of the Live Computer.

i Note

Make sure you are using the IP of the correct network if you work with multiple ones.

Select the corresponding IP for the plugin Input.

Now, we need to do the same inside for the spat plugins inside Live.

Go back to Ableton Live, open the plugin ROOM in your ROOM track, put the IP address of the computer hosting SPAT as the output, and select the IP corresponding IP address as your input. Do the same for the RETURN and the SEND.

Of course, don't forget to check if the OSC is enabled in the preferences, or SPAT will not receive anything.

Once this is done, our automation should now be received in SPAT.

And that is it!

44.10 Uninstall

44.10.1 With the FLUX Center

To Uninstall the Ableton Live Tools for SPAT Revolution with the FLUX:: Center, simply open the Center and click uninstall next to the Ableton Live Tools for SPAT Revolution.

44.10.2 Manually

To Uninstall the Ableton Live Tools for SPAT Revolution, simply go to this location:

`/Library/Application Support/Flux`

And delete the Ableton Live Devices Folder.

44.11 Credit

Ryan Farber for his M4L device: (*Simple One-Button Macro 2.0*) - <https://www.ryanfarber.com>
- <https://www.maxforlive.com/library/device/5353/simple-one-button-macro>

ELPHNT for his M4L device: (*NTPD2 Lite*) - <https://elphnt.io> - <https://maxforlive.com/library/device/4997/lite>

Devin Roth for creating BlackHole: - <https://existential.audio> - <https://www.devinrothmusic.com>

To the Ableton Team for making such an amazing DAW.

44.12 Troubleshooting / Q&A

44.12.1 Windows?

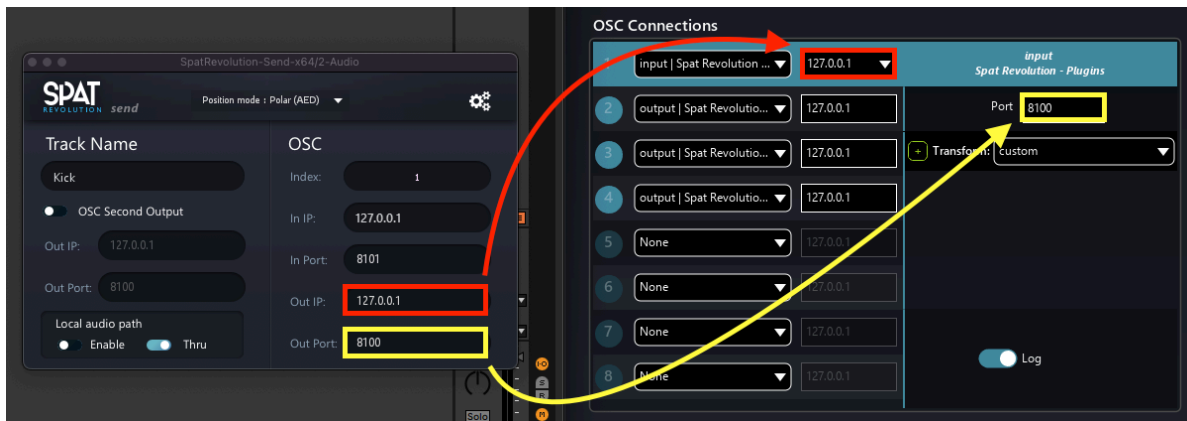
Q: Do you plan on making a Windows-specific guide?

A: Probably if we get some requests for it.

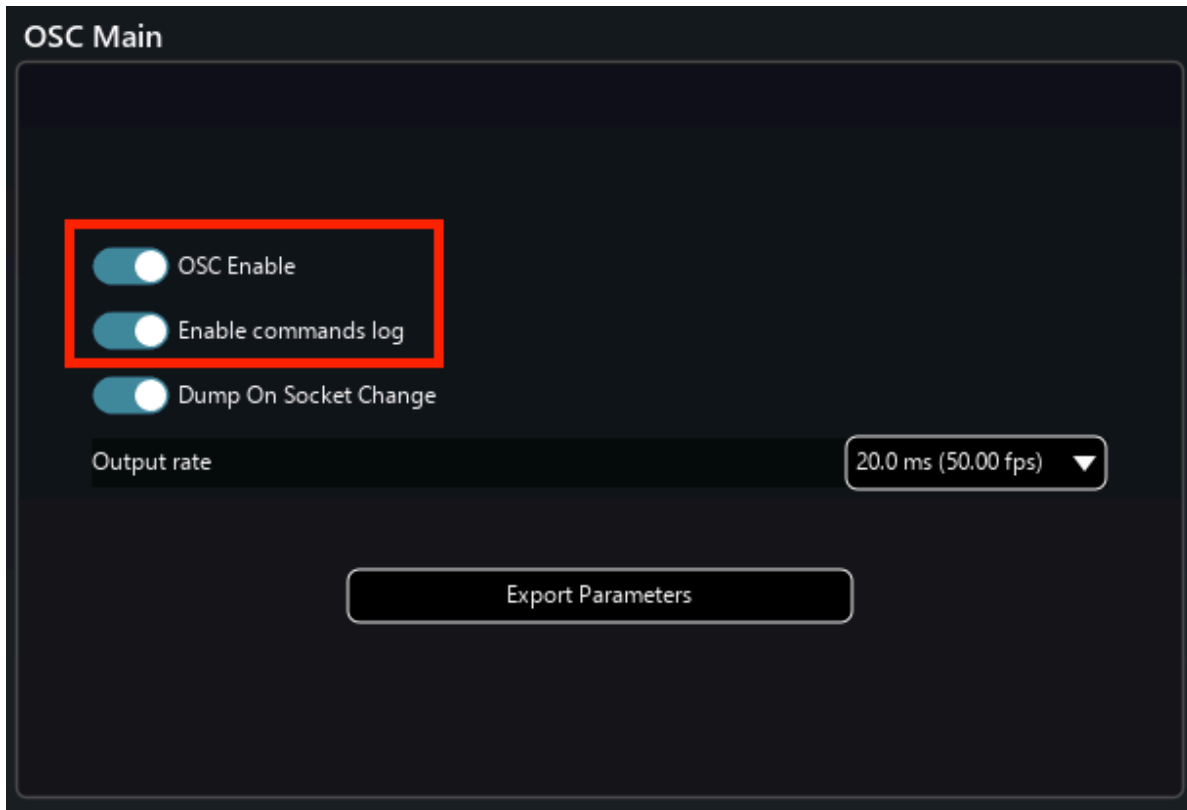
44.12.2 OSC issues? (Automation not going to SPAT)

If a source in SPAT does not seem to receive the Automation parameters, please check the following:

- Does the OSC Output port and IP Output in the send plugin (*ref n°13 in the SEND description*) match the IP and port inside the OSC settings (*Preference panel in SPAT*)?



- is OSC enabled (*Preference panel in SPAT*)



If you want to see if SPAT is correctly receiving the OSC messages, you can activate the “Enable commands log” option, which will now display every message received by SPAT in the Terminal.

You can open the SPAT Terminal with the following shortcuts:

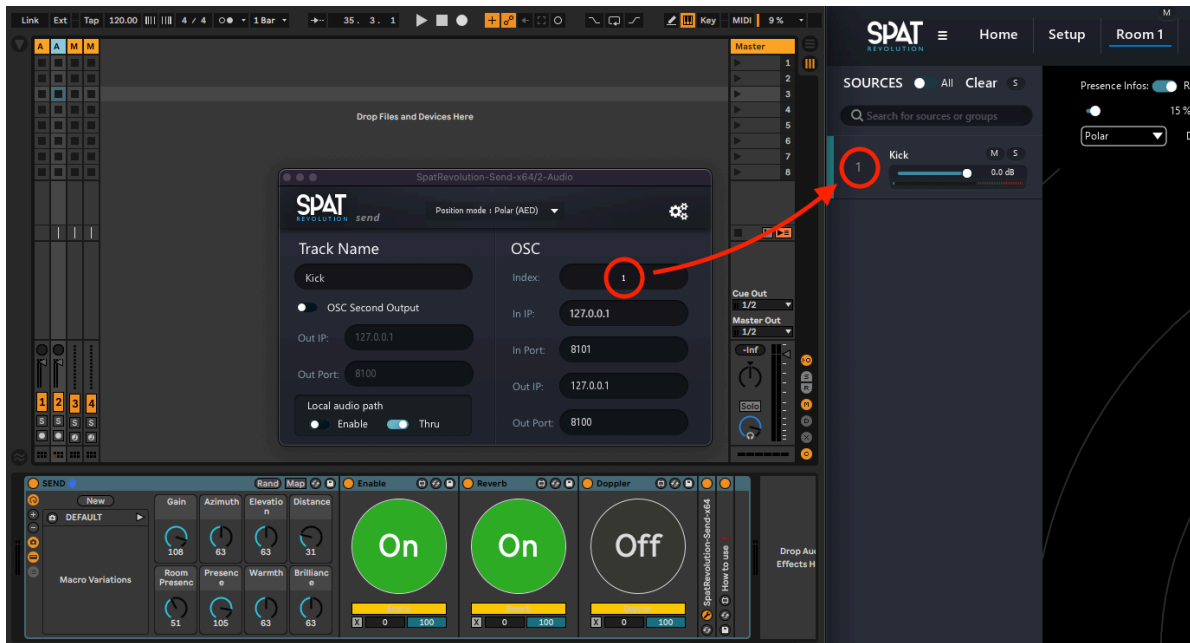
Show/Hide Terminal = *F7*

Show/Hide Terminal (mini) = *shift+F7*

Popup Terminal = *cmd+F7*


```
Terminal
OSC IN << /room/l/gain, [f][-24.20000076] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-24.39999962] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-24.79999924] - 127.0.0.1 : 8100
OSC OUT >> /room/gain, [if][1, -24.80000000] - 192.168.0.111 : 8000
OSC IN << /room/l/gain, [f][-25.10000038] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-25.29999924] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-25.70000076] - 127.0.0.1 : 8100
OSC OUT >> /room/gain, [if][1, -25.70000000] - 192.168.0.111 : 8000
OSC IN << /room/l/gain, [f][-26.00000000] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-26.20000076] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-26.39999962] - 127.0.0.1 : 8100
OSC OUT >> /room/gain, [if][1, -26.40000000] - 192.168.0.111 : 8000
OSC IN << /room/l/gain, [f][-26.79999924] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-27.00000000] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-27.50000000] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-27.70000076] - 127.0.0.1 : 8100
OSC OUT >> /room/gain, [if][1, -27.50000000] - 192.168.0.111 : 8000
OSC IN << /room/l/gain, [f][-27.89999962] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-28.10000038] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-28.50000000] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-28.70000076] - 127.0.0.1 : 8100
OSC OUT >> /room/gain, [if][1, -28.50000000] - 192.168.0.111 : 8000
OSC IN << /room/l/gain, [f][-29.10000038] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-29.39999962] - 127.0.0.1 : 8100
OSC OUT >> /room/gain, [if][1, -29.40000000] - 192.168.0.111 : 8000
OSC IN << /room/l/gain, [f][-29.60000038] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-29.79999924] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-30.20000076] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-30.39999962] - 127.0.0.1 : 8100
OSC OUT >> /room/gain, [if][1, -30.20000000] - 192.168.0.111 : 8000
OSC IN << /room/l/gain, [f][-30.79999924] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-31.00000000] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-31.20000076] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-31.39999962] - 127.0.0.1 : 8100
OSC OUT >> /room/gain, [if][1, -31.20000000] - 192.168.0.111 : 8000
OSC IN << /room/l/gain, [f][-31.79999924] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-32.00000000] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-32.20000076] - 127.0.0.1 : 8100
OSC IN << /room/l/gain, [f][-32.40000153] - 127.0.0.1 : 8100
OSC OUT >> /room/gain, [if][1, -32.20000000] - 192.168.0.111 : 8000
OSC IN << /room/l/gain, [f][-32.79999924] - 127.0.0.1 : 8100
>>>
```

- also make sure the OSC Index number in the SPAT plugin (*inside the send device*) matches the number of the track inside SPAT Revolution



You can apply the same treatment to the ROOM and RETURN as well if the automation on those aren't working too.

So, to sum up again: - IP in/out check in both the plugins and in the SPAT Revolution Preferences - OSC Index check in both the plugin and the source number in SPAT Revolution - And, of course, check if OSC is simply not activated

If you work with moving setup, you might get an error showing the OSC Port in red. To fix it, simply change the IP address to another network and then return to the original one.

If that does not do it, you can also change the OSC Port to something else, and then set back the correct port number.



44.12.3 SPAT Revolution Essential license?

Q: Is this workflow compatible with the **Essential** license of *SPAT Revolution*?

A: You do not have to get the Ultimate version to start creating with this workflow.

44.12.4 Demo Sessions?

Q: Is there demo sessions so I can have a quick look?

A: Yes, they are located in the same folder as the rest of the tools found here:
/Library/Application Support/Flux/Ableton Live Devices/SPAT/demo sessions

45 Steinberg Nuendo

A tutorial on [How to set up SPAT Revolution with Nuendo](#) is available here for a quick dive into this integration. You can also have a look at [SYNC: Steinberg Yamaha Network Channel on YouTube for Nuendo 11 | SPAT Revolution integration](#).

45.1 Templates

The following session templates are available for use with Nuendo. They are start sessions and can be used as examples to integrate *SPAT Revolution* using the SPAT plugin suite SEND, RETURN and ROOM.

You can download the following session templates:

- [Steinberg Nuendo Tutorial Template](#) is the template used in the above video tutorial and includes sends and returns setup for rendering Binaural, Atmos 5.1.4, NHK 22.2 with binaural monitoring.
- [Basic music NPR](#) is a basic template using sends and returns to render Stereo, 5.1 and Atmos 5.1.4 output formats with binaural monitoring.
- [Advanced multi-format NPR](#) is an advanced template using sends and returns to render Atmos 7.1.2, NHK 22.2 and multiple binaural outputs.
- [Ambisonic HOA mixing NPR](#) is a template for using 3rd order HOA and binaural monitoring output formats.

Please review the [Appendix B - Troubleshooting C](#) for troubleshooting.

45.2 Nuendo and External OSC rendering

Starting with Nuendo V11, it is now possible to deploy object-oriented sessions using open sound control (OSC). This brings the possibility to send/receive metadata from/to the Nuendo object VST MultiPanner. This functionality allows the support of *SPAT Revolution* as an external rendering engine using OSC thanks to the ADM-OSC [39](#) initiative. More information and specifications on the ADM-OSC initiative can be found on the dedicated GitHub repository, [immersive-audio-live/ADM-OSC](#).

Once configured, you can playback or record object-oriented sessions (audio and metadata) for live production and immersive creation workflows while using *SPAT Revolution* mixing and rendering capabilities.

As Nuendo can import and export ADM files, this allows for an ADM master, exported from another environment, to be imported into a Nuendo session and mapped to *SPAT Revolution*. This integration brings the ability to render the object-based mix for various stream types (Ambisonic, Binaural, Channel-based), from standard to custom speaker arrangements, and using multiple spatialization options and techniques.

The functioning is based on the declared ADM object approach in Nuendo. When a track is assigned to a multichannel output bus, the VST MultiPanner becomes available and can work in bed or object mode. Here, we are interested in the object mode allowing us to stream or listen to the object's position.

Complete information on dealing with objects in Nuendo is available in their documentation, steinberg.help - Nuendo 11.

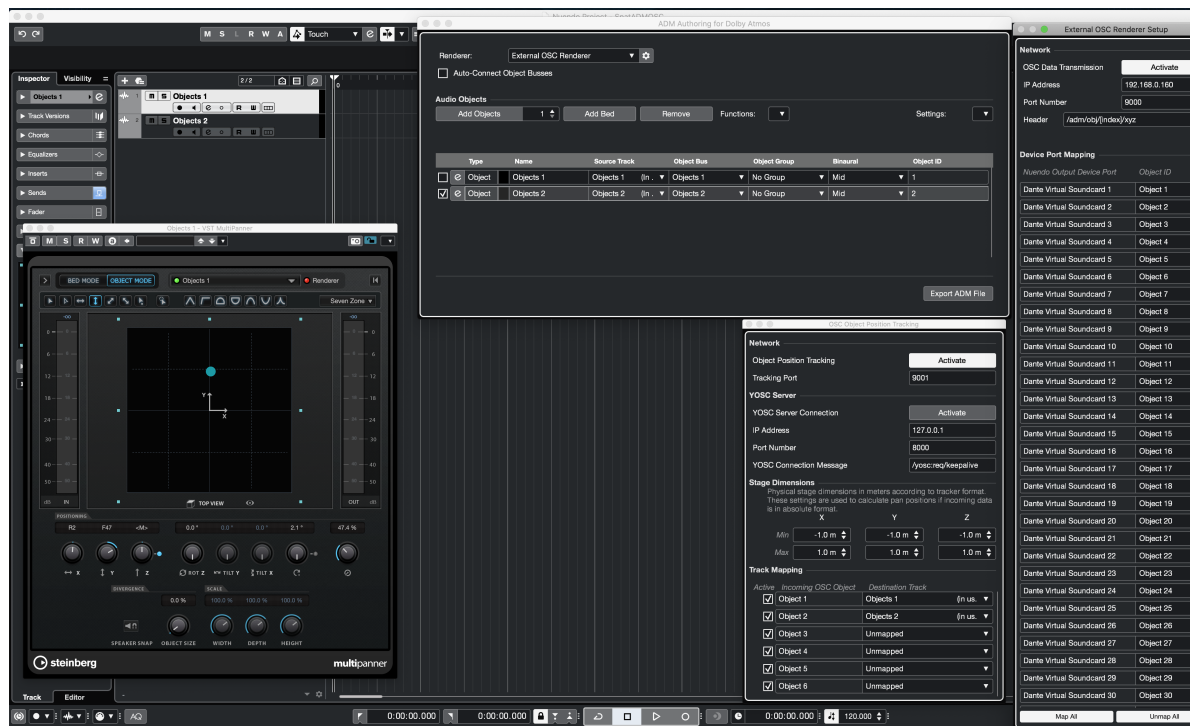


Figure 45.1: Object-based in Nuendo V11

45.3 Using ADM-OSC in Nuendo / Use cases

Although controlling *SPAT Revolution* source objects from Nuendo audio tracks is possible with the SPAT send plugin and its automation, the actual integration of the Nuendo VST MultiPanner with *SPAT Revolution* brings the ability to remain in the mixer environment. With this, users can stay within the typical mixer panner automation and use the same standard remote control tools (HUI, MCU, and EUCON compatible controllers can be used, bringing tactile functionality to *SPAT Revolution*). It can be used with current sessions by simply adding the connection to SPAT Revolution as an external rendering tool, leading to perceptual factors of objects and an acoustic simulation helping build soundscapes. Various use cases are possible:

- Import an ADM file from another environment, render with SPAT Revolution in channel-based (various panning and speaker arrangement formats) or scene-based (binaural, ambisonic up to 7th order).
- Deliver alternate formats from the same session you have rendered your Dolby Atmos deliverables.
- Re-render old sessions using the existing VST MultiPanner position but with SPAT Revolution as the rendering engine.
- Record all SPAT Revolution object position metadata (from live or studio) to the Nuendo VST MultiPanner(while still being able to use the SPAT Send for other parameters).

45.3.1 System schematics - Nuendo and SPAT Revolution

Basic setup where Nuendo is playing back to SPAT Revolution via software input, and the system output/monitoring is going out directly to your audio hardware device in SPAT Revolution.

Setup where you are using SPAT Send and Return Local Audio Path mode to route the signal to/from Nuendo and SPAT Revolution. The return(s) allow to bounce in Nuendo the rendering result and manage the monitoring bussing needs.

Setup where audio bridge devices are used to connect Nuendo to/from SPAT Revolution. The typical scenario involves a macOS system with the audio bridge part of an aggregate device with your actual audio interface used for monitoring.

Typical dual computer setup where AoIP AES67 (Ravenna), AVB, Dante, or other multichannel audio interfaces such as with MADI are used to send and receive signals between Nuendo and SPAT Revolution.

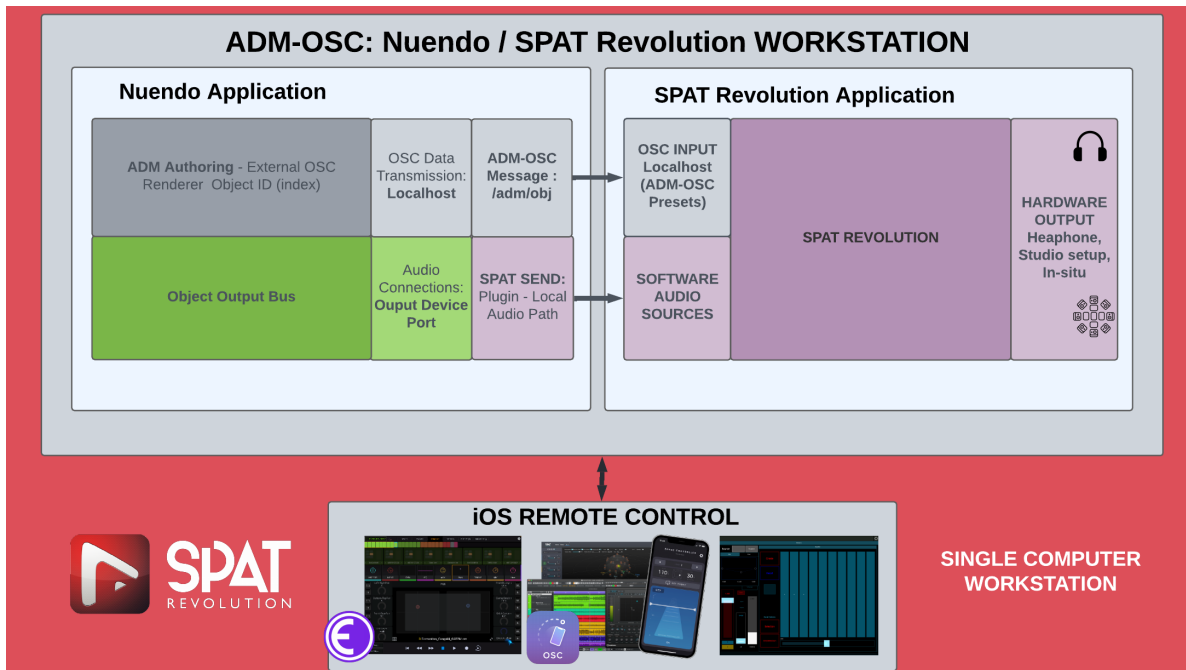


Figure 45.2: System schematics - Nuendo and SPAT Revolution - Software In / Hardware Out

45.3.2 Creating an up to 64 objects session in Nuendo

- First, create an empty Nuendo project
- In the menu **Studio / audio connection**:
 - Delete any outputs not required in the audio connection/studio menu.
 - Add Bus: 1 x 7.1.4 Atmos Master and 1 x 7.1.2 Bed (in some cases, you won't need the 7.1.2 as it would get created by the ADM import process if all goes well).
 - Make sure output 1 to 64 remains unpatched, leaving this part of the patch to objects.
 - Patch 7.1.2 bus to 65 to 74 and 75 to 84 for the Atmos master bus.

Even if you aren't going to be using the Dolby capabilities of Nuendo and are planning only to render with *SPAT Revolution*, you need the 7.1.4 Atmos Master bus as this is being used to declared tracks as objects and assign objects IDs

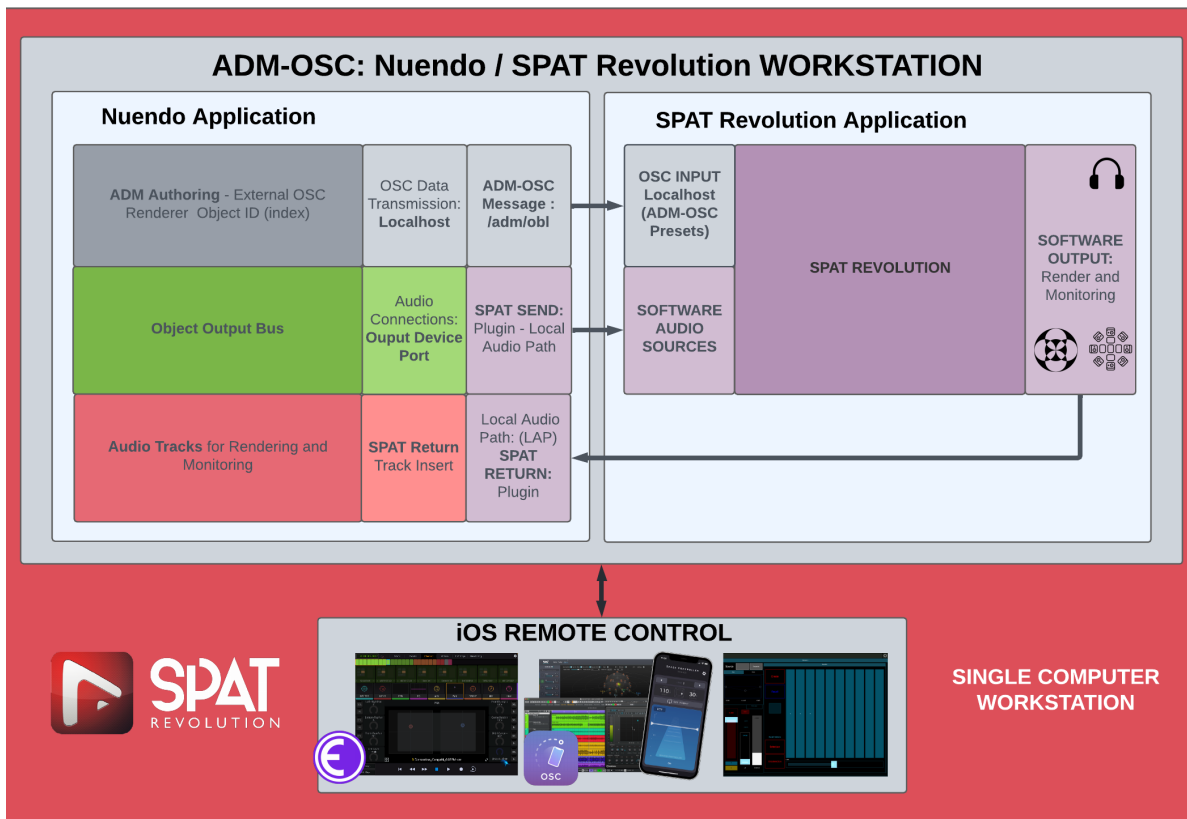


Figure 45.3: System schematics - Nuendo and SPAT Revolution - Software I/O with SPAT Plugins

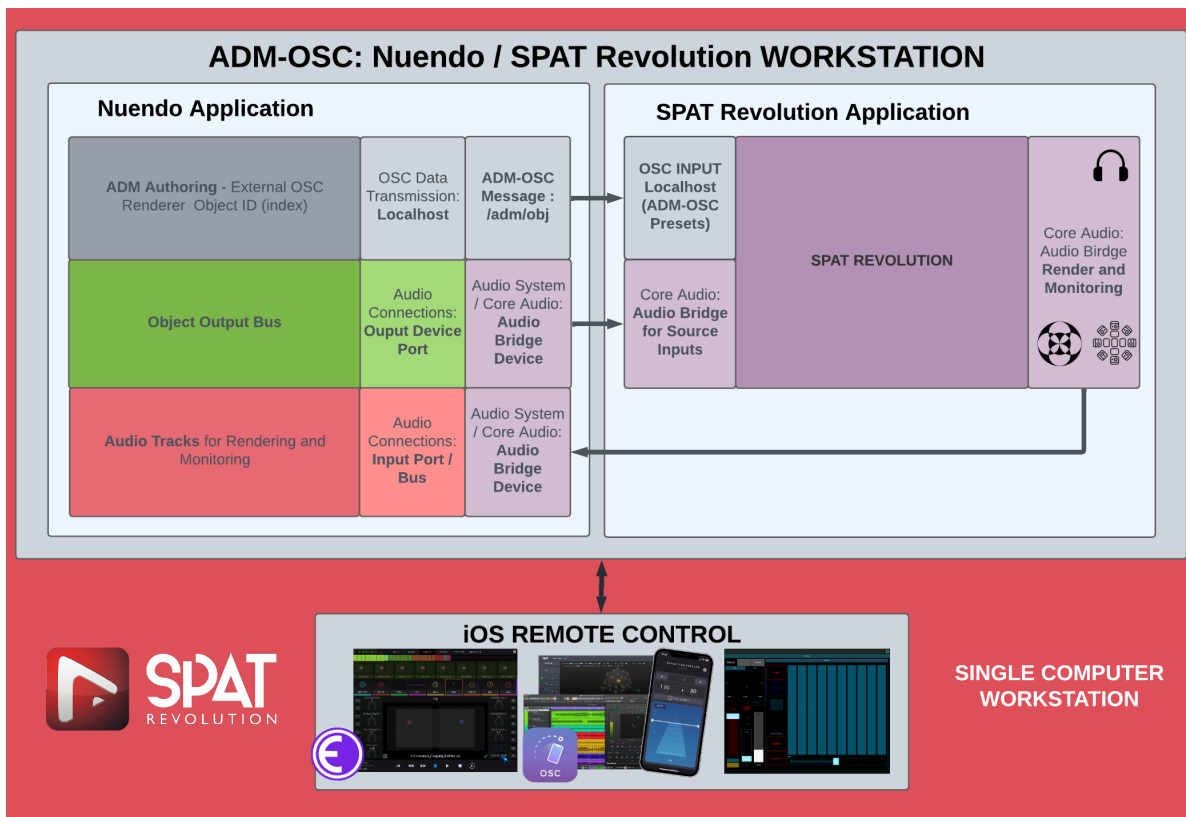


Figure 45.4: System schematics - Nuendo and SPAT Revolution - Core Audio / Audio Bridge

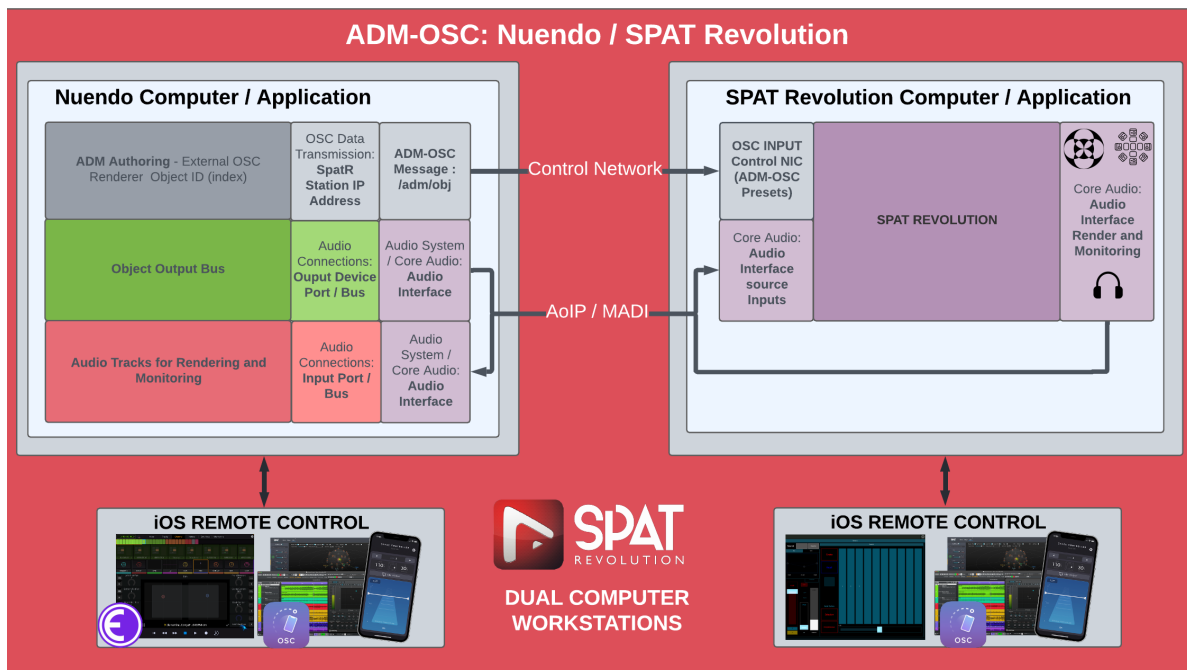


Figure 45.5: System schematics - Nuendo and SPAT Revolution - Dual Computers

45.4 Nuendo v11 - Setting SPAT Revolution as external renderer

The first step consists of configuring SPAT Revolution as the External Renderer. You can access the setup window by choosing External OSC Renderer setup in the Studio Menu.

In the Studio / External OSC Render setup menu:

- Make sure the OSC Data Transmission is activated.
- Make sure the port number is set to 9000, as it is the default SPAT Revolution ADM OSC preset input port.
- Set the IP address to 127.0.0.1 (localhost) for a single computer setup, or set the IP address of the second computer hosting SPAT Revolution (the rendering computer).
- Header: Insert the ADM-OSC header message, /adm/obj/[index]/xyz
- Device Port Mapping: Map All - 1 to 1

45.5 Importing ADM File

To start your project from an existing ADM master file of another environment, go to the File menu and choose Import ADM

Inputs		Outputs	Group/FX	External FX	External Instruments	Control Room
Add Bus		Presets	No Preset		Expand All	Collapse All
Bus Name	Speakers	Audio Device	Device Port			
▼ 7.1.2 Bed	7.1.2	FLUX:: Audio Bridge 96ch				
⇒ Left			FLUX:: Audio Bridge 96ch 65			
⇒ Right			FLUX:: Audio Bridge 96ch 66			
⇒ Center			FLUX:: Audio Bridge 96ch 67			
⇒ Lfe			FLUX:: Audio Bridge 96ch 68			
⇒ Left Surround			FLUX:: Audio Bridge 96ch 69			
⇒ Right Surround			FLUX:: Audio Bridge 96ch 70			
⇒ Side Left			FLUX:: Audio Bridge 96ch 71			
⇒ Side Right			FLUX:: Audio Bridge 96ch 72			
⇒ Top Side Left			FLUX:: Audio Bridge 96ch 73			
⇒ Top Side Right			FLUX:: Audio Bridge 96ch 74			
▼ Atmos Master	7.1.4	FLUX:: Audio Bridge 96ch				
⇒ Left			FLUX:: Audio Bridge 96ch 75			
⇒ Right			FLUX:: Audio Bridge 96ch 76			
⇒ Center			FLUX:: Audio Bridge 96ch 77			
⇒ Lfe			FLUX:: Audio Bridge 96ch 78			
⇒ Left Surround			FLUX:: Audio Bridge 96ch 79			
⇒ Right Surround			FLUX:: Audio Bridge 96ch 80			
⇒ Side Left			FLUX:: Audio Bridge 96ch 81			
⇒ Side Right			FLUX:: Audio Bridge 96ch 82			
⇒ Top Front Left			FLUX:: Audio Bridge 96ch 83			
⇒ Top Front Right			FLUX:: Audio Bridge 96ch 84			
⇒ Top Back Left			FLUX:: Audio Bridge 96ch 85			
⇒ Top Back Right			FLUX:: Audio Bridge 96ch 86			

Figure 45.6: Audio Connection in Nuendo V11

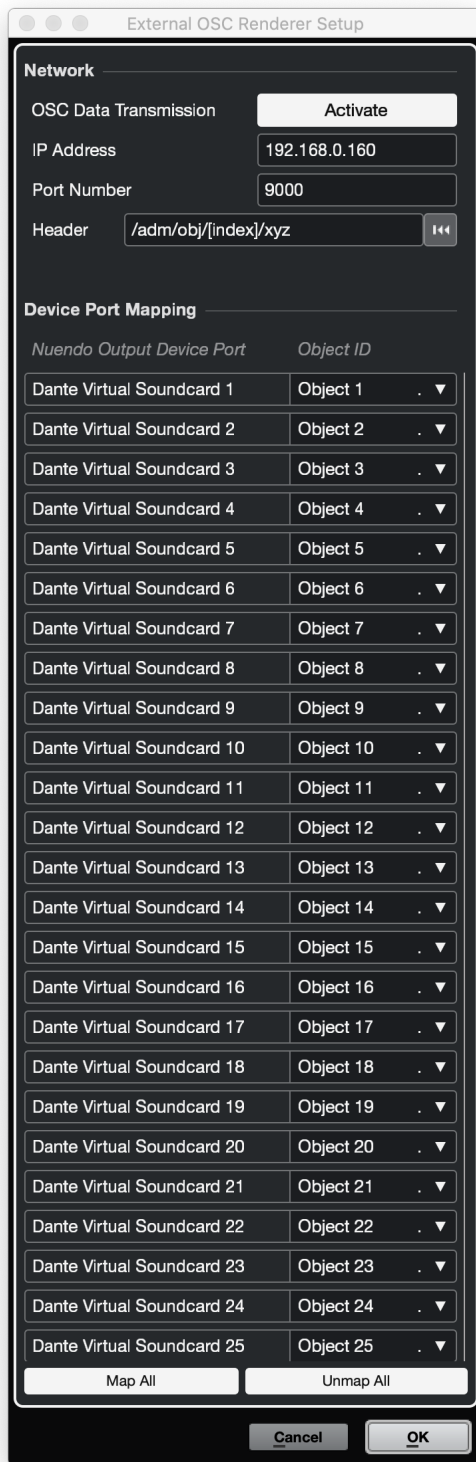


Figure 45.7: Nuendo OSC Renderer Setup

- Select all tracks in the ADM import menu
- When the import is done, ensure the Atmos_bed track is assigned to the 7.1.2 output bus.
- Select all Atmos_Obj and assign them to the 7.1.4 Atmos master bus created (Shift + Control + choose the 7.1.4 bus assignment to do them as a batch).
- Keeping all Atmos_Obj tracks selected (only, not the bed), go to the menu **Project / ADM Authoring for Dolby Atmos**. If objects already exist, start by deleting them.
- Choose **External OSC** renderer on the Renderer pulldown.
- Make sure **Auto-Connect Objects Busses** is marked.
- Click on the **Functions** arrow pulldown and choose **Create Objects from the Selected Tracks**.
- It should create all that you need. All objects are mapped to index 1 to N.
- New output buses are getting created to match objects 1 to N.

45.6 Nuendo audio connections configured for object

45.7 Nuendo object position to SPAT Revolution source objects.

After the following setup details, your object position data in your Nuendo session will be streaming to SPAT Revolution for an external rendering.

As this metadata is sent with normalized value according to the ADM-OSC specification, SPAT Revolution ADM-OSC input preset and transformation will allow scaling to the desired automation zone range.

Nuendo can also receive normalized position data from SPAT Revolution ADM-OSC XYZ output (preset), map them to the VST MultiPanner (position tracking), and write automation data with the corresponding audio object if desired.

45.8 Setting up Nuendo OSC Object Position Tracking

This next part covers incoming data to Nuendo. This would be to actually record the object information (from a live performance, for example) to the Nuendo VST MultiPanner, so ultimately use it as automation.

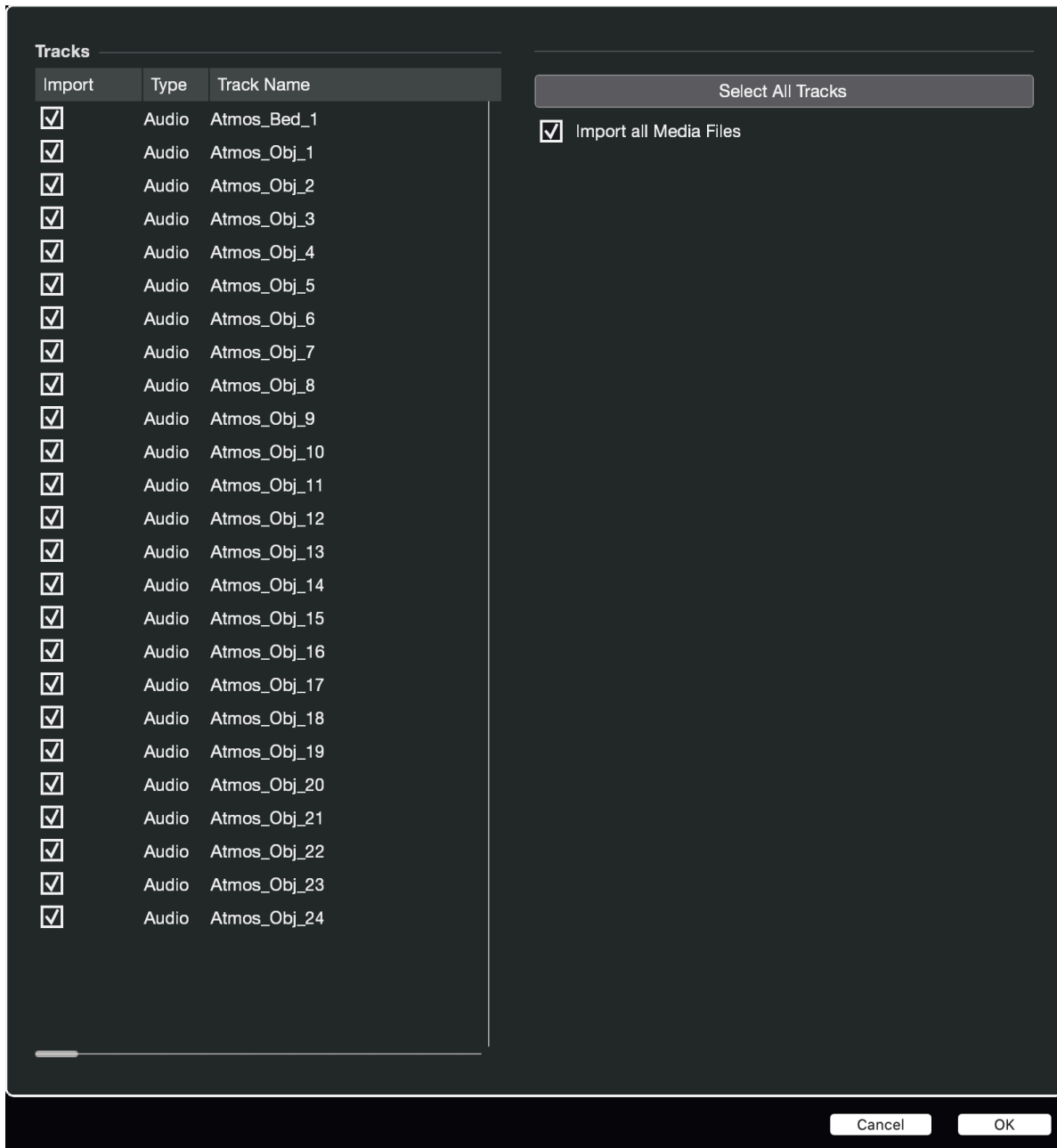


Figure 45.8: Nuendo Import ADM

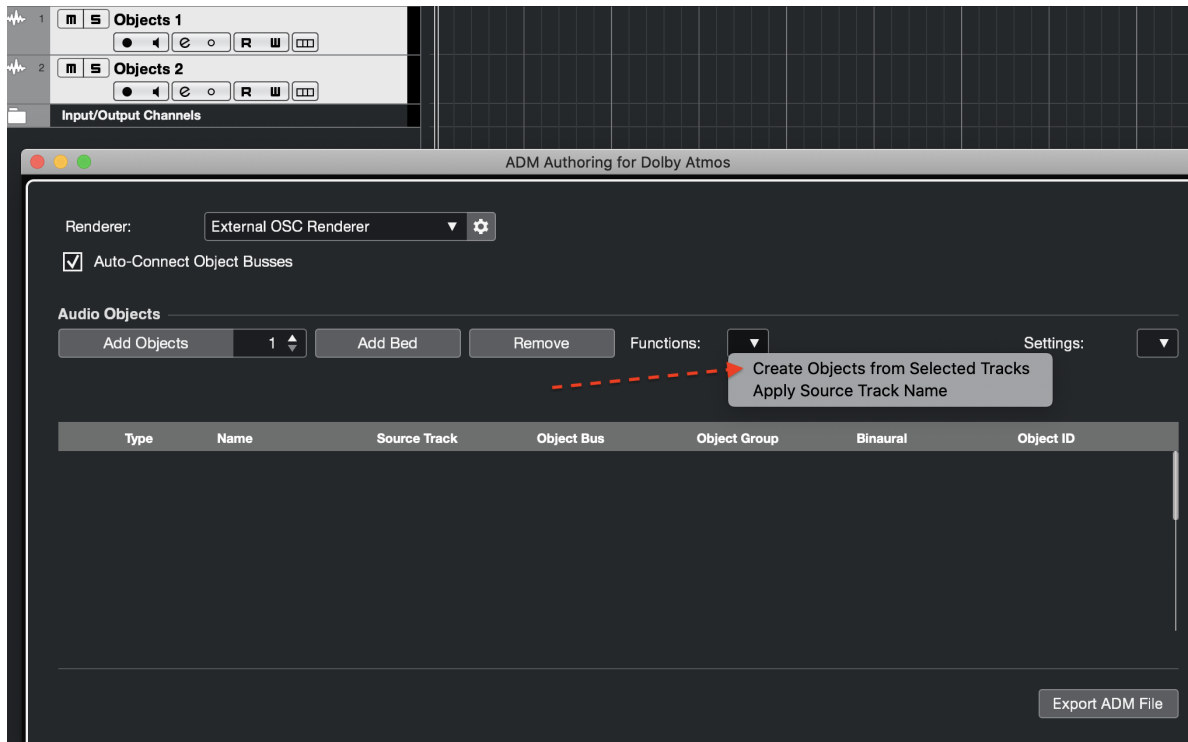


Figure 45.9: Nuendo ADM Authoring for Dolby Atmos 1

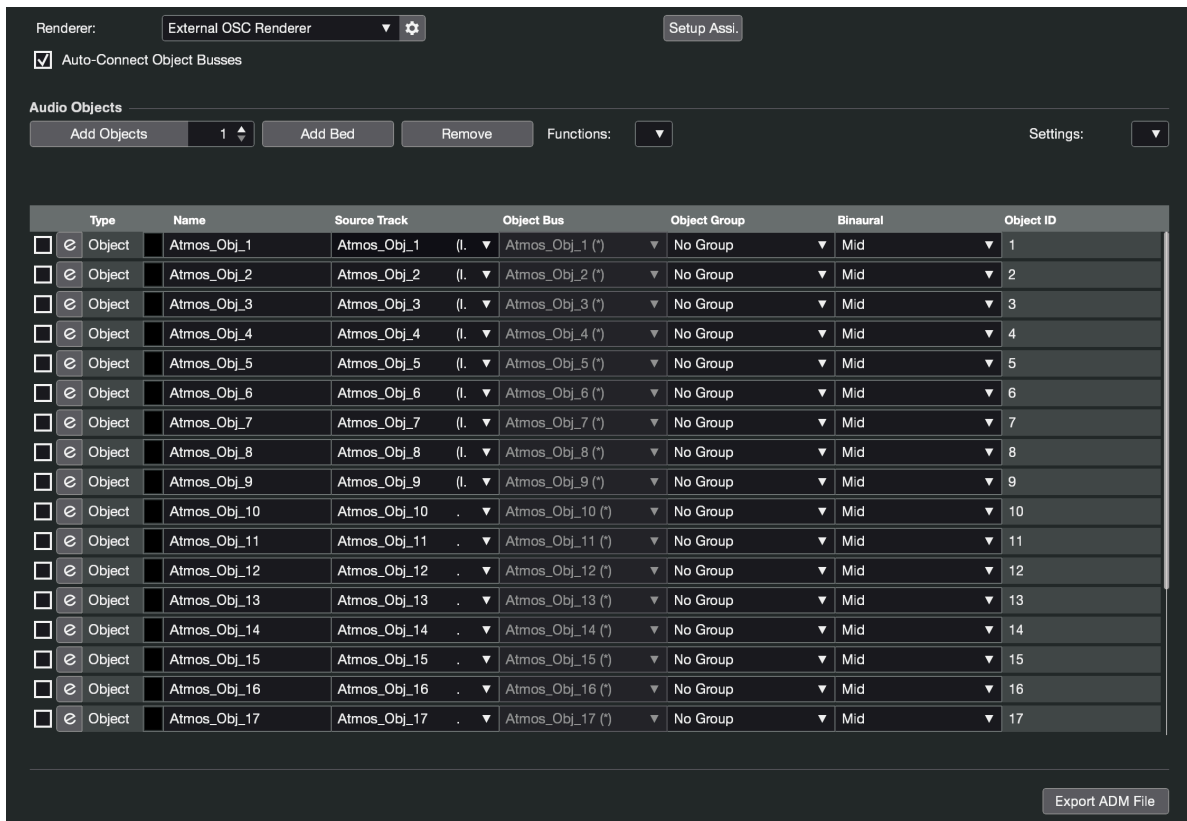


Figure 45.10: Nuendo ADM Authoring for Dolby Atmos 2

Inputs		Outputs	Group/FX	External FX	External Instruments	Control Room
Add Bus		Presets	No Preset ▼		Expand All	Collapse All
Bus Name	Speakers	Audio Device	Device Port			
▶ 7.1.2 Bed	7.1.2	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 65			
▶ Atmos Master	7.1.4	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 75			
▶ Atmos_Obj_1 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 1			
▶ Atmos_Obj_2 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 2			
▶ Atmos_Obj_3 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 3			
▶ Atmos_Obj_4 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 4			
▶ Atmos_Obj_5 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 5			
▶ Atmos_Obj_6 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 6			
▶ Atmos_Obj_7 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 7			
▶ Atmos_Obj_8 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 8			
▶ Atmos_Obj_9 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 9			
▶ Atmos_Obj_10 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 10			
▶ Atmos_Obj_11 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 11			
▶ Atmos_Obj_12 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 12			
▶ Atmos_Obj_13 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 13			
▶ Atmos_Obj_14 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 14			
▶ Atmos_Obj_15 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 15			
▶ Atmos_Obj_16 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 16			
▶ Atmos_Obj_17 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 17			
▶ Atmos_Obj_18 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 18			
▶ Atmos_Obj_19 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 19			
▶ Atmos_Obj_20 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 20			
▶ Atmos_Obj_21 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 21			
▶ Atmos_Obj_22 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 22			
▶ Atmos_Obj_23 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 23			
▶ Atmos_Obj_24 (*)	Mono	FLUX:: Audio Bridge 96ch	FLUX:: Audio Bridge 96ch 24			

Figure 45.11: Audio Connection in Nuendo V11 with Bus and Object outputs



Figure 45.12: Nuendo Surround VST MultiPanner in Object Mode

i Note

At the time of writing, it is not recommended to configure the objects bi-directionally as some workflow challenges exist with object index ID when dealing with a mix of mono, stereo, or multichannel objects.

Go to the **Studio** menu and choose **OSC Object Position Tracking**.

- Make sure **Object Position Tracking** is activated.
- Make sure port # is set to 9001. This is the default ADM-OSC output port for SPAT Revolution preset.
- Stage Dimensions: This is the ability to scale incoming OSC into Nuendo. By default, it is currently at 0,1. This is not the default ADM-OSC specification. You can simply change this by entering -1.0 m as a minimum and 1.0 m as a maximum.
- Track Mapping lets you map the incoming Index from SPAT Revolution to the actual Nuendo object.

45.9 SPAT Revolution ADM-OSC settings for Nuendo

45.9.1 ADM-OSC Input setup

In the SPAT Revolution OSC Connection preferences:

- Choose **Input ADM-OSC** preset and select your network interface. If Nuendo and SPAT Revolution are on the same computer, choose **Localhost 127.0.0.1**, otherwise choose the network interface where OSC messages are incoming.
- You can edit the transform by double-clicking on it. This is where you will define the SPAT Revolution automation zone range (scaling to). For example, here we are using -3, 3.

i Note

The “scaling to” is you defining what will be the maximum position value when the Nuendo panner is in its extreme corners.

You are done for the input!

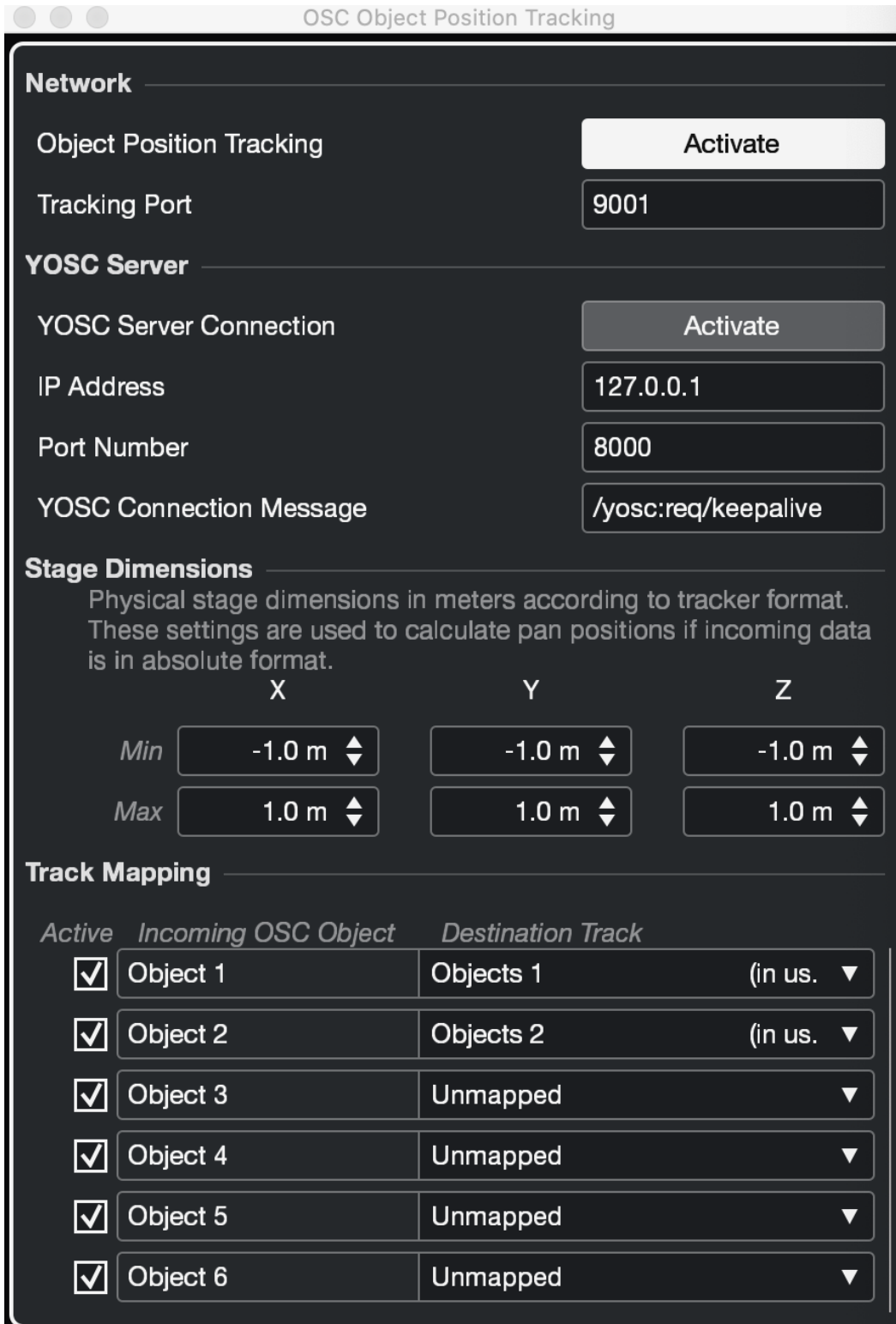


Figure 45.13: Nuendo OSC Object Position Tracking

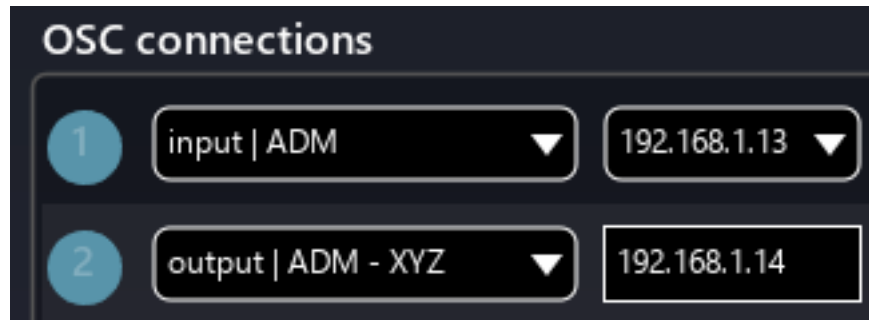


Figure 45.14: SPAT OSC settings for Nuendo ADM-OSC

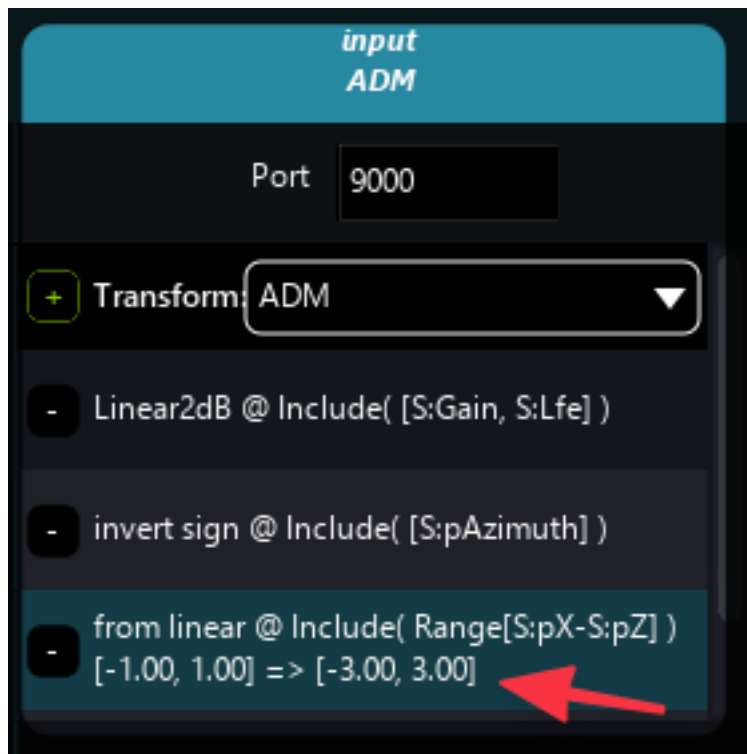


Figure 45.15: Modifying (edit) the transform of scaling

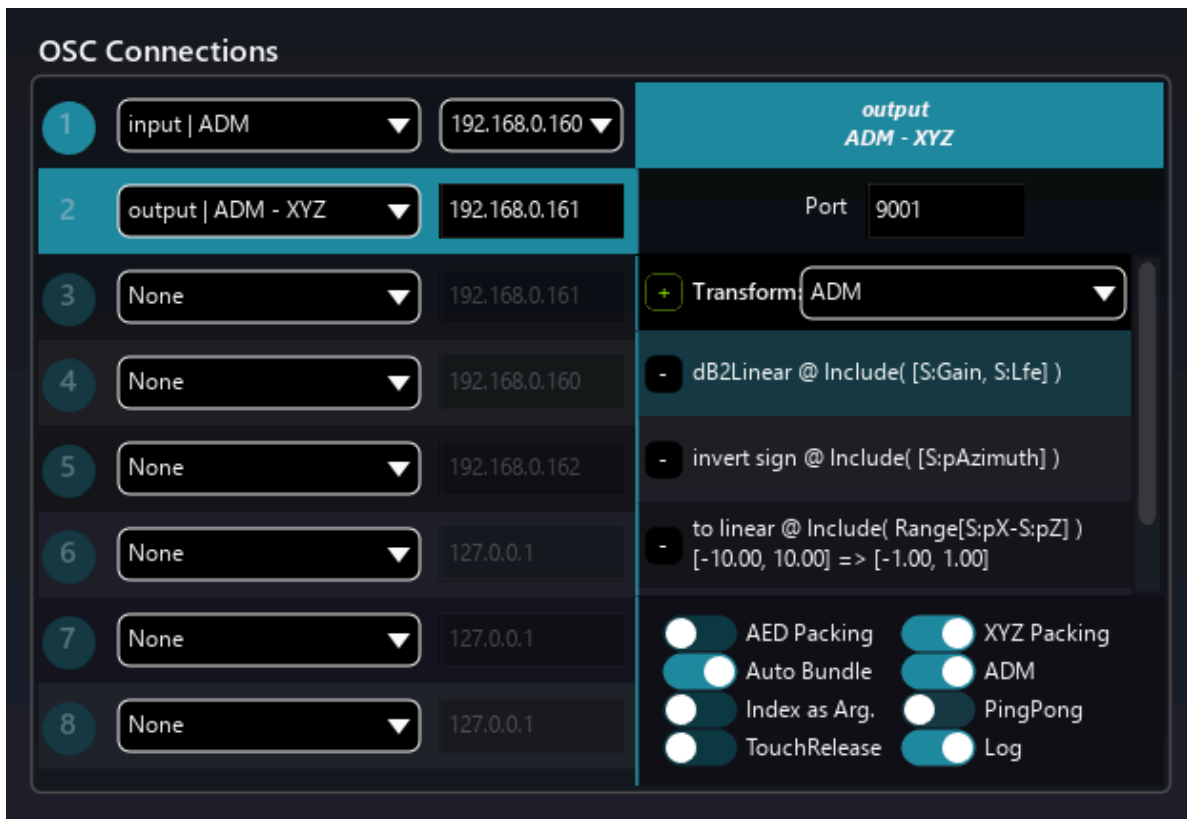


Figure 45.16: ADM XYZ Output

45.9.2 ADM-OSC XYZ Output to Nuendo

This configuration is for sending SPAT source object data to Nuendo.

- Choose the Output **ADM-XYZ** preset and set the IP address of the Nuendo computer.(if local, use localhost 127.0.0.1)
- The transformation preset is configured by default, as long as your OSC Object Position Tracking setup in Nuendo has stage dimensions -1.0, 1.0.
- You can edit the automation zone (range) you are sending to Nuendo. This is the same as used for the input range—for example, -3, 3.

i Note

The “scaling from” is you defining the SPAT Revolution position stage zone where objects are sent to Nuendo in a normalized manner.

Et voila! You are set.

Ready to move info in an object-oriented workflow!

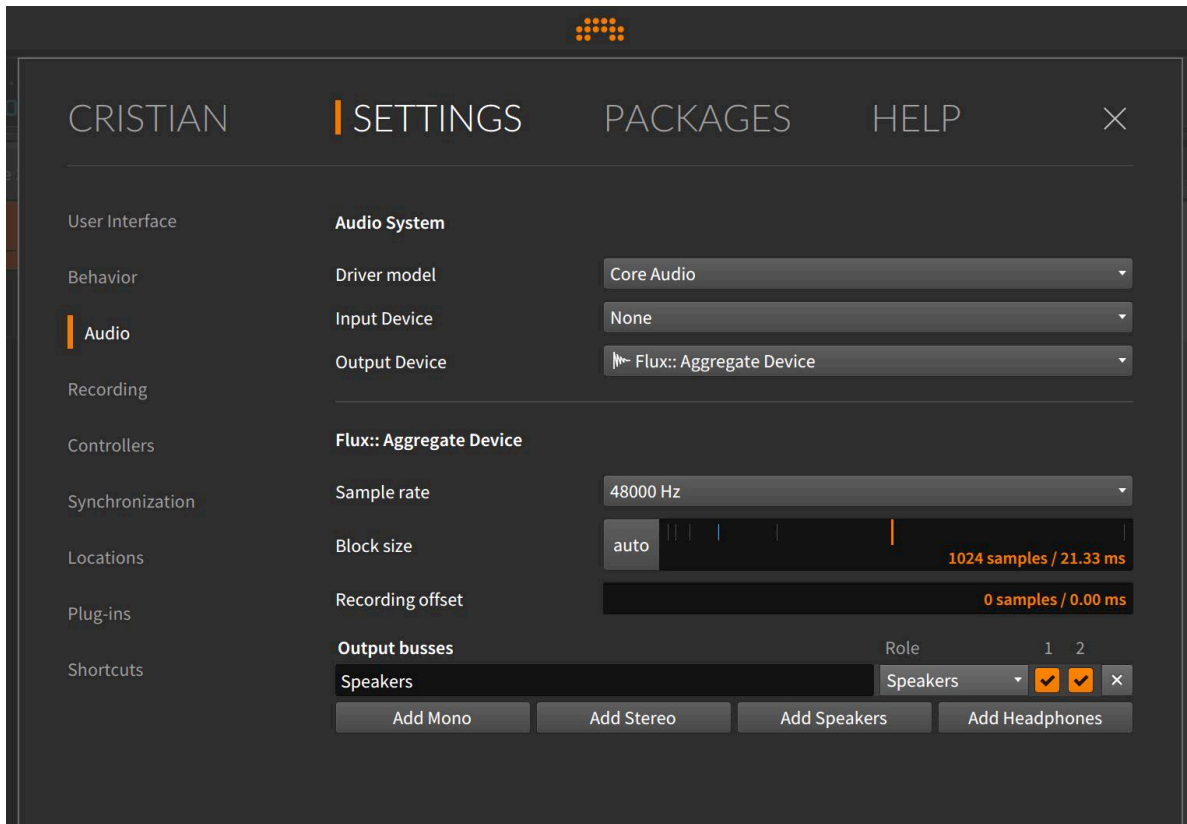
46 Bitwig Studio

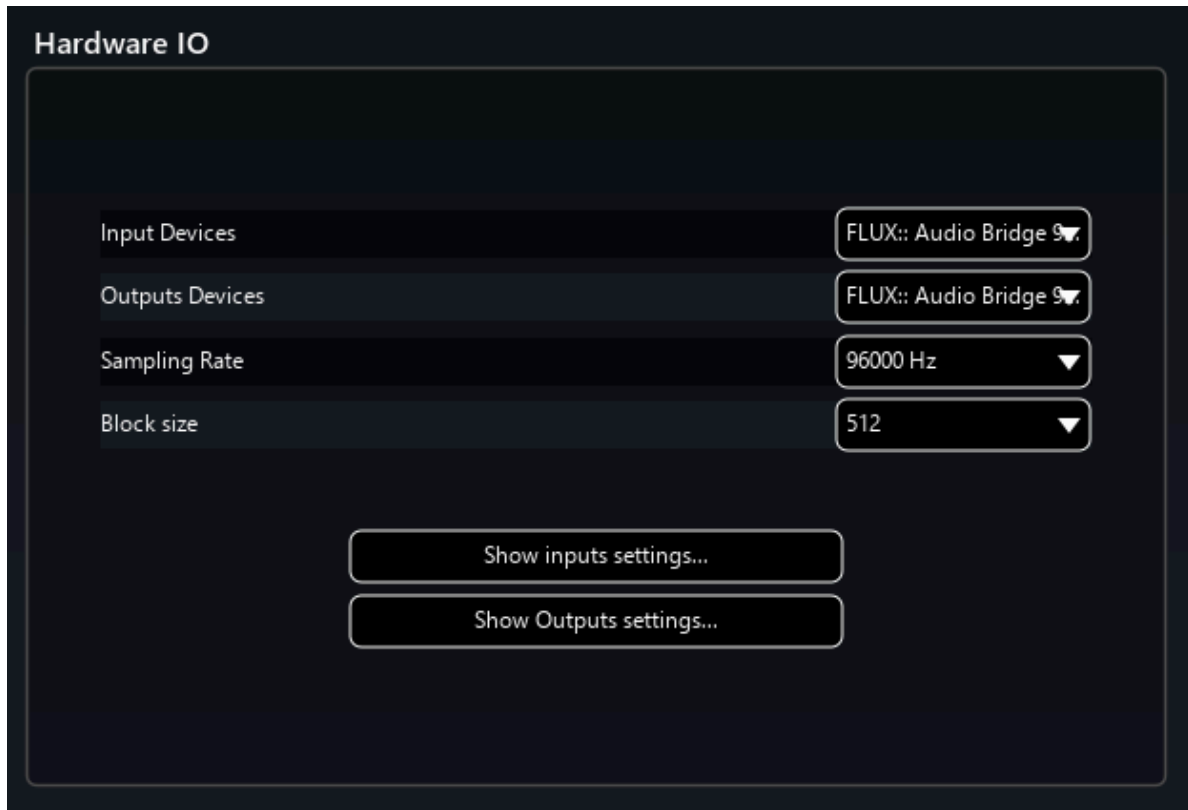
Bitwig Studio is a DAW designed in Berlin with a live, hands-on philosophy. Along with its highly animated and intuitive graphic interface, it offers well-designed clip-based and timeline arrangement paradigms for composing. The Bitwig designers have included a complete suite of powerful and great-sounding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



Setting Up Sync in BitWig

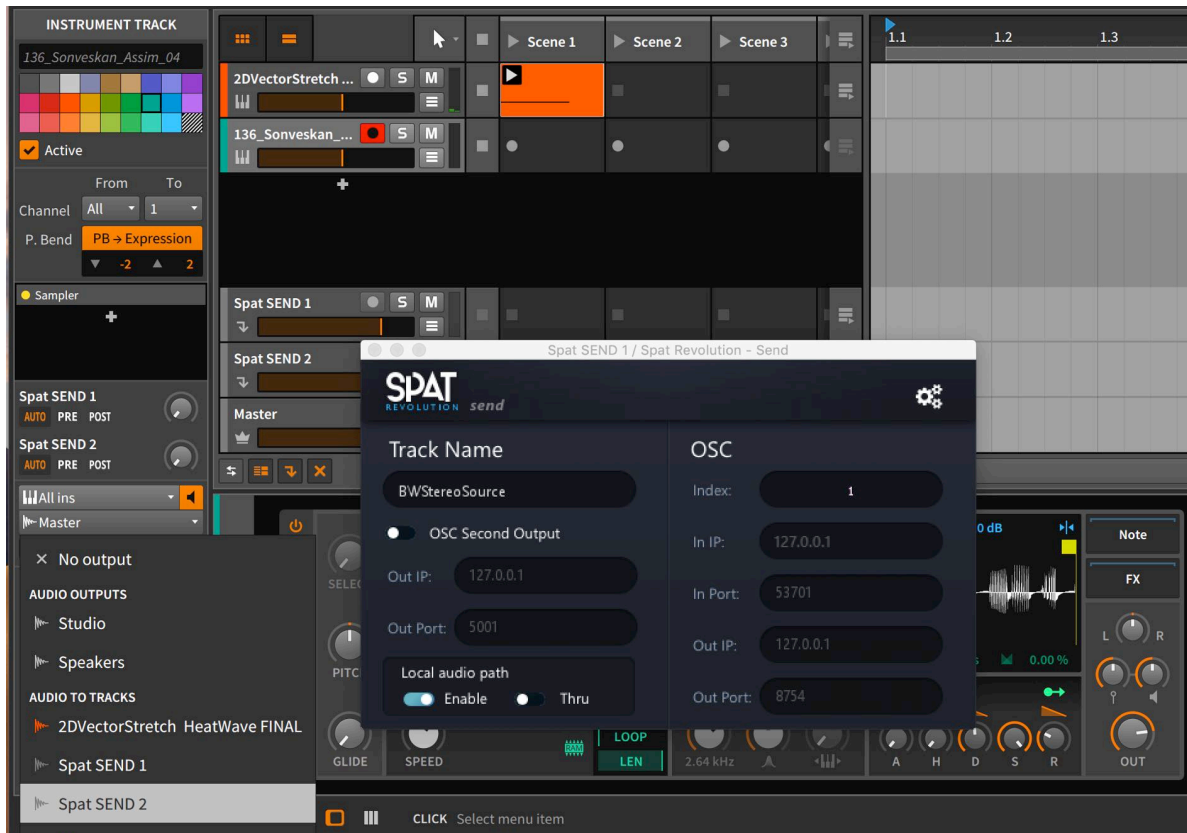
When using the Local Audio path (LAP), the buffer size and sample rate must be matched in both *SPAT Revolution* and Bitwig Studio. In SPAT, you do this in the preferences, and in Bitwig, in the audio engine settings. If they don't match at first, you *may* need to restart both applications to get the correct green sync status between the apps.





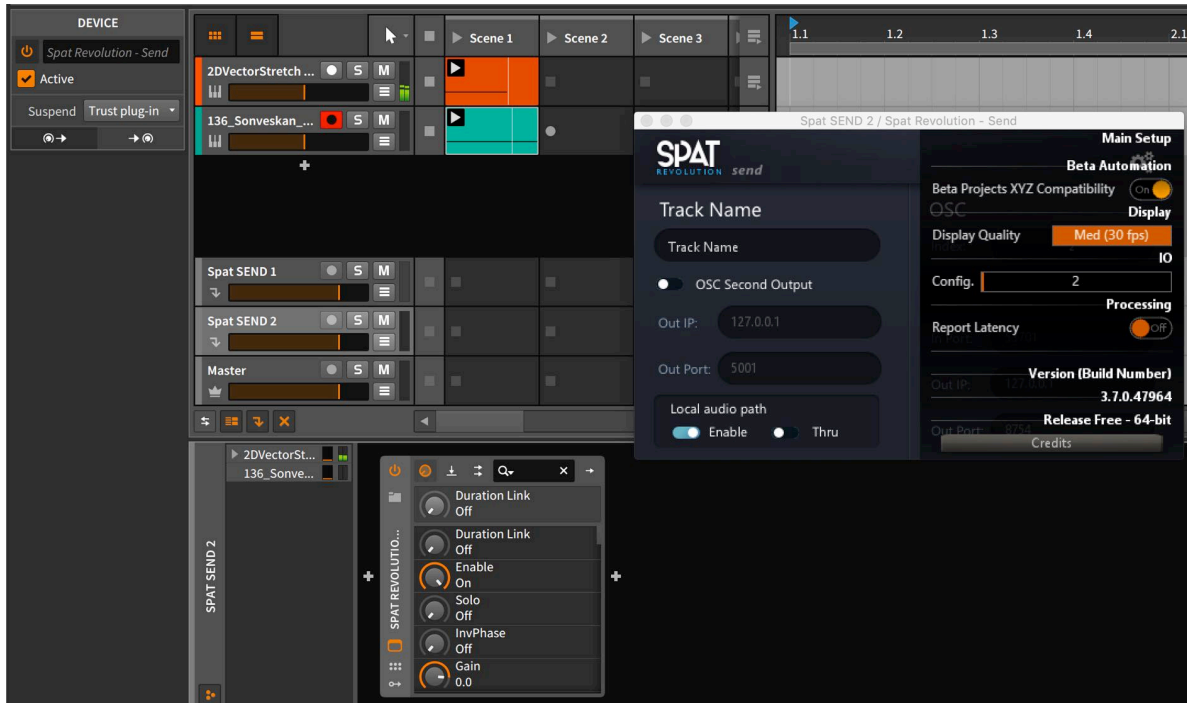
Setting Up Tracks in BitWig

One good way to work with Bitwig and SPAT is to set Bitwig tracks to output their audio to *Effect Track* types - 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*, and enable the local audio path with **THRU** set off, so all audio streams are rendered to output in SPAT Revolution.

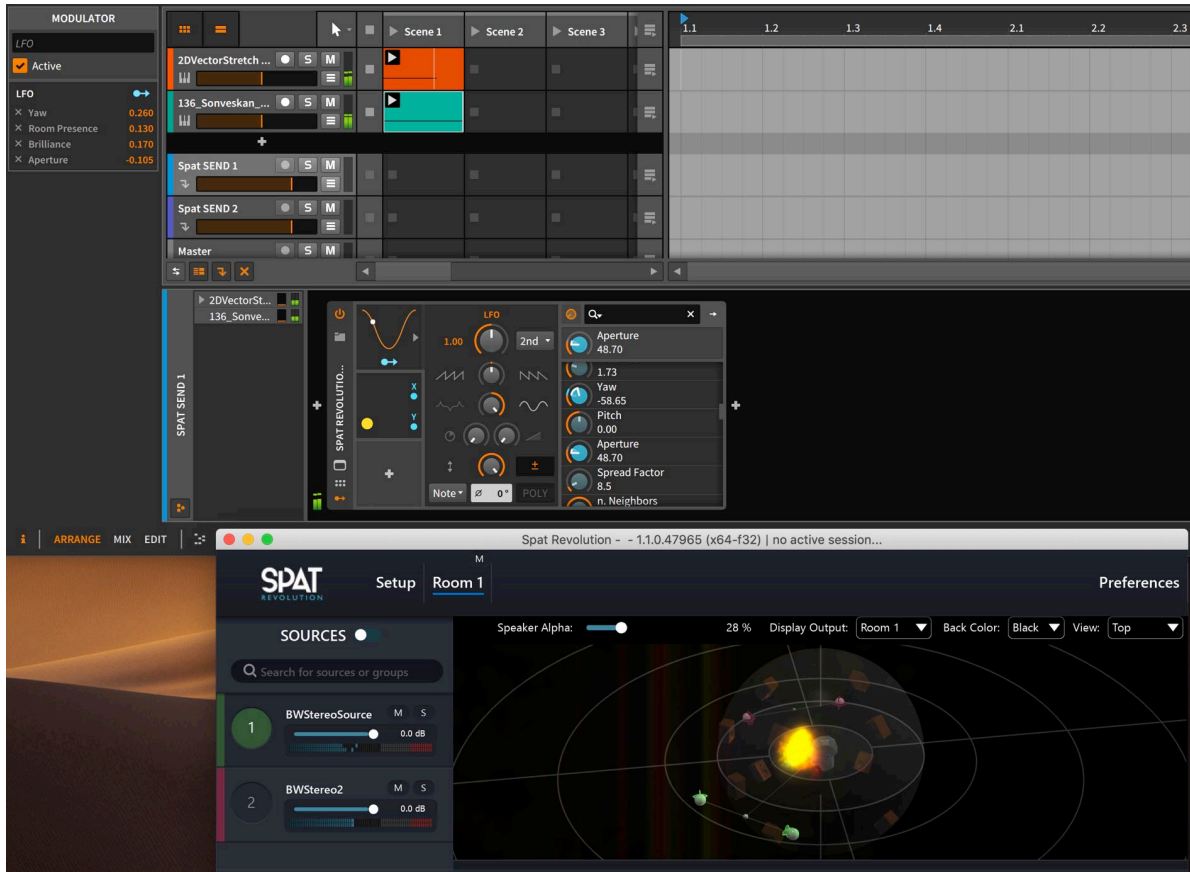


Like in all Local Path Audio workflows, you should see SPAT SEND inputs appearing in the SPAT Environment Setup, which relate directly to the plug-ins hosted in the other software environment, reflecting their TrackName and channel count.

SPAT Source Automation in BitWig

Now the fun starts - on the Bitwig Send tracks, which host the SPAT SEND plug-ins, 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

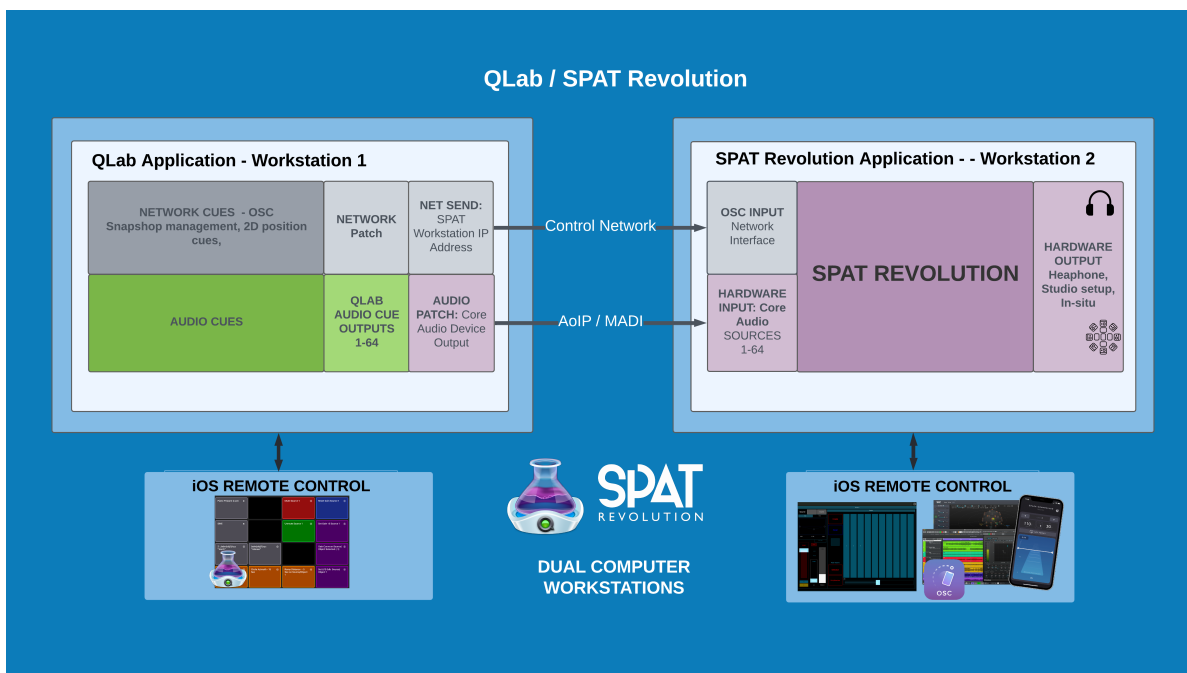
BitWig has comprehensive support for many popular MIDI controller surfaces. These can be very easily mapped to SPAT SEND parameters for hands-on control of virtual objects and rooms.

Troubleshooting

For troubleshooting, please review the [Appendix B - Troubleshooting C](#)

47 Figure53 QLab

Rock solid and fully featured show control program, based under a ‘cue’ based paradigm, QLab is excellent 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 a great integration partner to *SPAT Revolution*.



QLab Dual Computer setup

47.1 QLab 5

Starting with QLab5, the network type cue (OSC messages) has been substantially revamped and now includes support to control SPAT Revolution directly. Over 100 predefined messages for SPAT Revolution are currently possible, including project, source, room, master, and snapshot control messages.

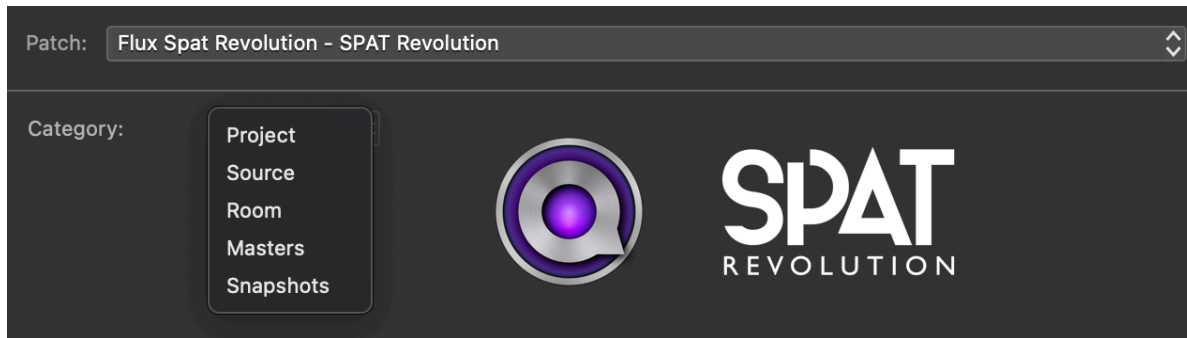


Figure 47.1: QLab5

Earlier provided templates (QLab 4) of actual examples of remote control messages for SPAT Revolution are no longer needed to assist with programming.

47.2 QLab 5 Integration template

New QLab 5 template will be provided shortly.

47.3 QLab 4 Integration template (V4 validated)

This template shows how to manage *SPAT Revolution* and QLab integration on the same machine, using *SPAT Revolution* Send plugins. Both QLab and *SPAT Revolution* sessions are included, with 16 Mono and 8 Stereo SPAT Send on cue outputs. This template is *SPAT Revolution* Essential compliant for binaural and Channel-based setups.

QLab templates are available in EURO and US versions, using commas or periods for denoting the decimal. (such as the decimal used in interpolation time)

[QLab 4_SPAT Revolution_ integration EURO.qlab4](#)

[QLab 4_SPAT Revolution_ integration US.qlab4](#)

47.4 Generic QLab Templates (V4, compliant with V5)

This template shows how to manage *SPAT Revolution* snapshots within QLab and have some carts for quick actions. It demonstrates how interpolation time value can be used in the snapshot recall messages.

[Qlab SPAT Revolution snapshot carts EURO.qlab4](#)

Qlab *SPAT Revolution* snapshot carts US.qlab4

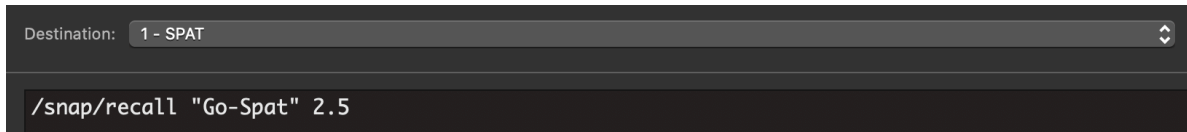


Figure 47.2: SPAT Snapshot message with interpolation time

QLab *SPAT Revolution* OSC Message examples

This template is our updated template and includes many cue examples using various message types in SPAT Revolution.

Beyond direct cue send actions, it brings 1D and 2D fade (Parameter ramp, 2D trajectories, and more). As of QLab4, there is a time interpolated 2D fade system for creating spatial XY gestures or similar multi-parameter control ideas. X/Interpolation time value can always be used directly in your messages, as shown in some template examples. With the latest release of *SPAT Revolution*, the ability to send messages to the currently selected source (s) with index -1 is also shown in this template.

Qlab *SPAT Revolution* examples EURO.qlab4

Qlab *SPAT Revolution* examples US.qlab4

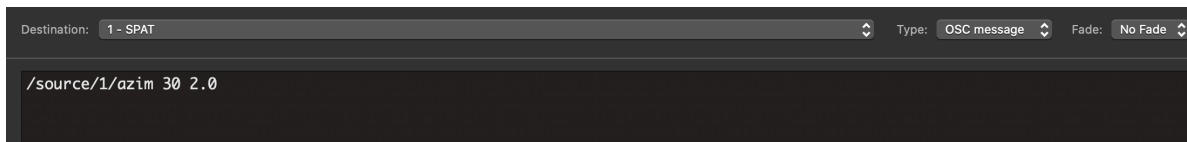


Figure 47.3: SPAT Message with interpolation time

47.5 Configuring QLab -> SPAT for messages via network cues

In a real-time situation, where performers or sounds are being spatialized live by *SPAT Revolution*, and cues need to be sent in the right running order with the rest of the show, Network OSC type cues can be sent from QLab to *SPAT Revolution* to control all aspects of the SPAT rendering software. To do this interaction, setting up the OSC communication is necessary. It is relatively straightforward. *In the SPAT Revolution preferences, ensure the OSC Enable is engaged.*

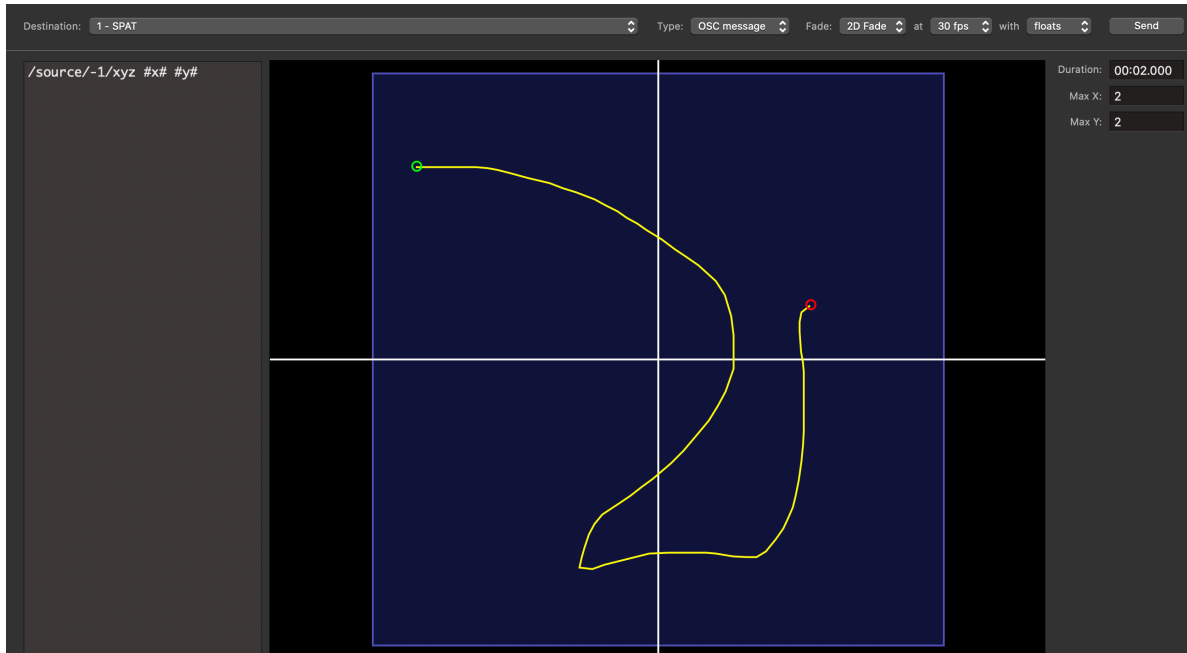


Figure 47.4: 2D Fade - trajectory messages

i Note

Enable commands log to view the commands and confirm you are receiving data (**Shift + F7** will open the log window). It is recommended to leave it active only sometimes, as it takes some system resources.

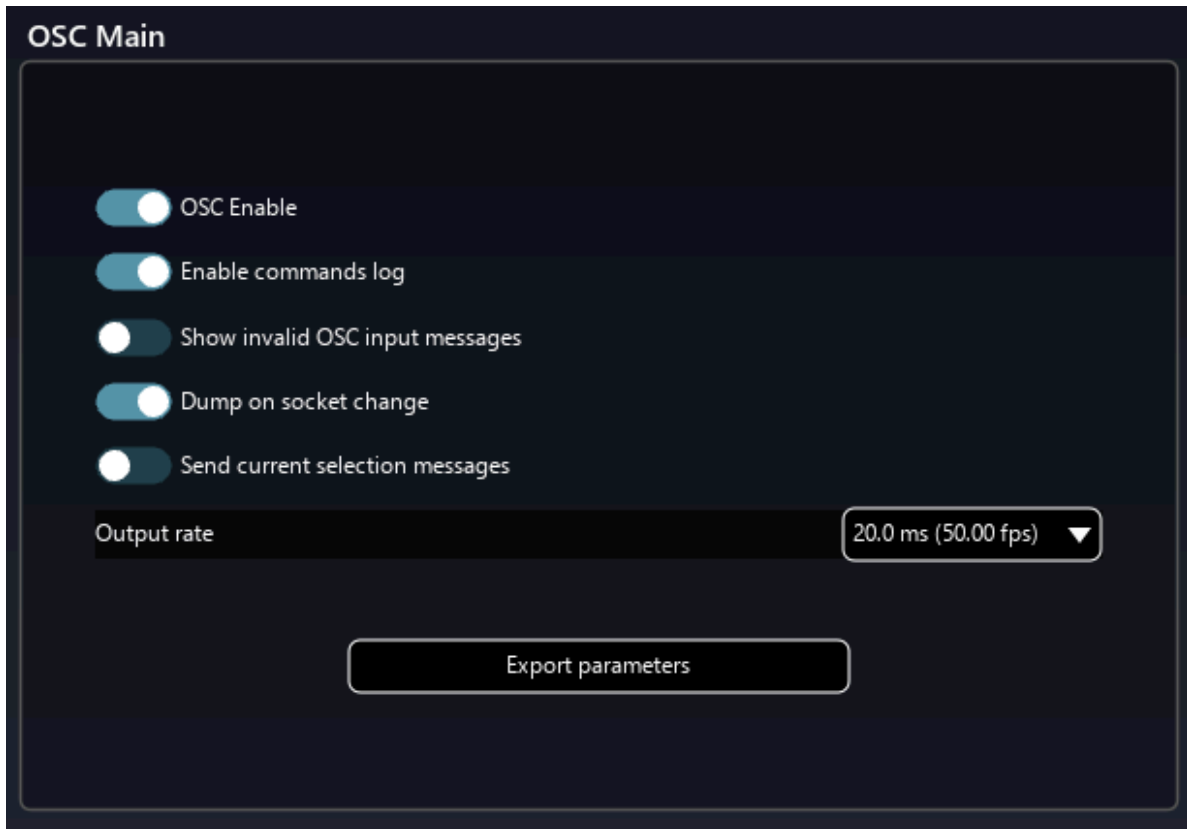
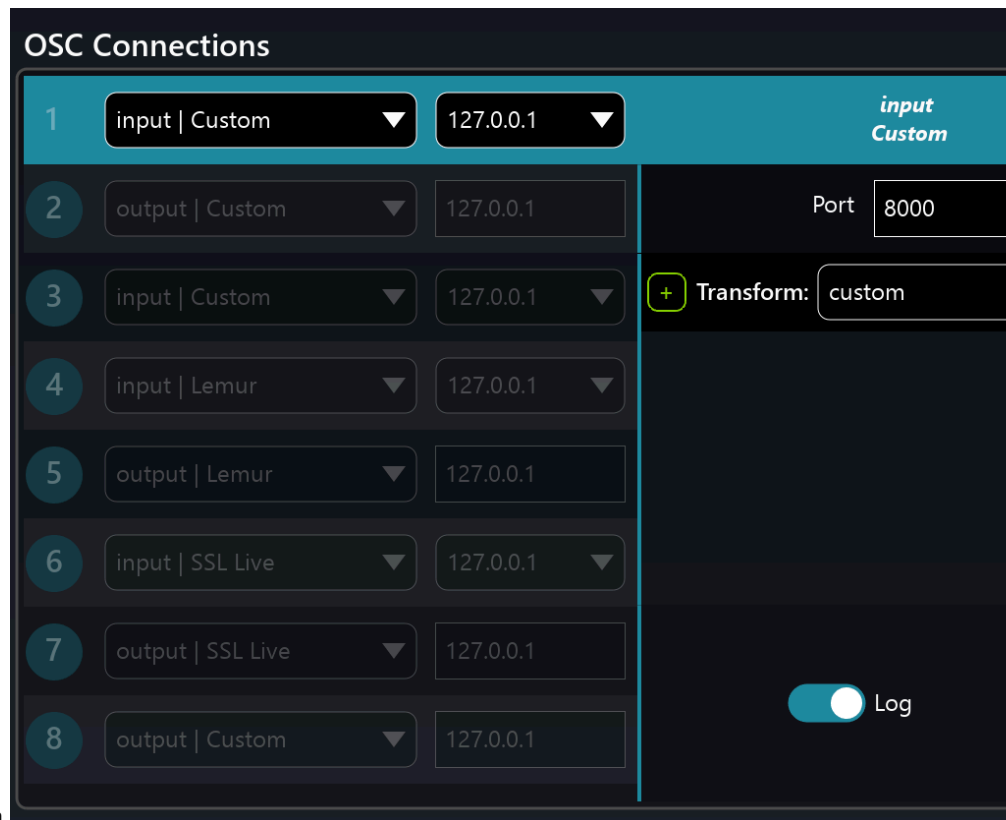


Figure 47.5: Setting the OSC Connection



Setting the OSC Connection

Go to the OSC connection section of *SPAT Revolution* and:

- Change the pull-down menu from *None* to the *Input / Custom* preset (meaning you are setting an OSC Input of Spat).
- Select the network interface you want to be receiving the commands from. Doing a local integration of QLab will require you to choose the localhost/loopback address `127.0.0.1`.
- Unless you already use the Port 8000 proposed, you do not need to change it. This is the corresponding QLab Network Patch Output Port used in templates.

QLab Workspace Settings / Network

General	Network Cue Destination Patches							New Patch ⌘N
Audio	Patch	Name	Type	Network	Destination		Passcode	
Video	Patch 1:	SPAT	address	Automatic	127.0.0.1	8000	Passcode	
Light	Patch 2:	OSCulator	address	Automatic	192.168.1.201	8010	Passcode	
Network	Patch 3:	ADM-OSC	address	Automatic	192.168.1.202	9000	Passcode	
MIDI	Patch 4:	Name	address	Automatic	Address	0	Passcode	
Cue Templates	Patch 5:	Name	address	Automatic	Address	0	Passcode	
Key Map	Patch 6:	Name	address	Automatic	Address	0	Passcode	
MIDI Controls	Patch 7:	Name	address	Automatic	Address	0	Passcode	
OSC Controls	Patch 8:	Name	address	Automatic	Address	0	Passcode	

On the QLab side, use the Network Patch settings to configure OSC destinations. One of them can be SPAT.

You can now send OSC network cues from QLab to SPAT, and control **most if not all parameters** of this virtual environment using Appendix C - OSC and ADM-OSC Table D. Once you get the hang of it, this is really very straightforward.

The screenshot shows the QLab software interface. At the top, there is a 'GO' button and a search bar for 'Position and Radiation'. Below that is a 'Notes' field. A toolbar contains various icons for cue management. The main area is a cue list table with columns: Number, Q, Target, Pre Wait, Action, and Post Wait. The first cue is 'Position and Radiation' with a target of '00:00.00' and a post wait of '00:00.00'. It is followed by several cues for setting source parameters like Azimuth, Distance, Elevation, and XYZ location. A section titled 'Level Changes' is expanded, showing cues for moving sources and processing changes. On the right side, there are playback controls (back, stop, play, mute) and a cue list sidebar showing '3 Cue Lists' and '0 Active Cues'.

Number	Q	Target	Pre Wait	Action	Post Wait
	Position and Radiation		00:00.00		00:00.00
	Set Azimuth Source 1 to 30 degree with 2 sec interpolation		00:00.00	00:00.00	00:00.00
	Set Distance Source 1 to 2 Meters with 5 sec interpolation		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
	Source 1 - XY Location preset		00:00.00	00:00.00	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 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
	Enable Relative Directivity - Source 1		00:00.00	00:00.00	00:00.00
	Disable Relative Directivity - Source 1		00:00.00	00:00.00	00:00.00
	Level Changes		00:00.00	00:00.00	00:00.00
	Movements on selected source (index -1)		00:00.00	00:10.00	00:10.00
	Move XY 2 Meter Square - 2 Sec		00:00.00	00:02.00	00:00.00
	Ping Pong Fx - 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		00:00.00	00:00.00	00:00.00
	2D Map Test		00:00.00	00:00.00	00:00.00
	Move XY 2 Meter Square - 2 Sec		00:00.00	00:02.00	00:00.00
	Processing Changes		00:00.00	00:10.00	00:00.00

...

48 Avid VENUE S6L

Spat Send is the only available plugin from FLUX:: Immersive for the *SPAT Revolution* integration to **Avid VENUE S6L** platform. The mono or stereo plugin instances provide the integration of the Spat source/object parameters into the channel strip (insertion on the channel for source parameter control, no audio processing). This channel strip can be a pure input channel or an actual bus (aux, group) for working with stem objects.

The plugin offers the ability to control the Spat source/object parameters directly on encoder, and automate them within the VENUE show file and snapshot system. And ultimately, having a completely integrated control of the objects in the 3D space. Added to the SPAT source/object parameters is a **time** value where the user can set a linear morphing between two parameter states over a time period to create smooth transitions, thus creating movements with interpolation time with the VENUE Snapshots.

New with version 20.12:

- Single X-Time (Interpolation) value for all parameters of PI
- X-Link allows to link all SPAT PI plugins interpolation time
- Page Table updated.
- Scale (for stereo) and LFE Send parameters were added.
- Communication improvements and optimization.

The OSC (Open Sound Control) part is the crucial aspect of this integration as when properly configured, the console will be sending OSC commands to *SPAT Revolution* software engine (OSC Out) and receiving OSC commands into the S6L console (OSC In). Thus, for a bidirectional configuration. This is what the SPAT Send plugin does on the console. Optionally, a second OSC output from the console can be enabled to send to a backup *SPAT Revolution* computer engine (unidirectional). The bidirectional aspect will mean that creating a sound scene in *SPAT Revolution* interface (or from third-party interfaces) will automatically push the information to the plugin instances on the console. Furthermore, the pushed data can be captured in a console snapshot. The console PI snapshots can capture automation prepared in pre-production (third party like DAW or controllers).

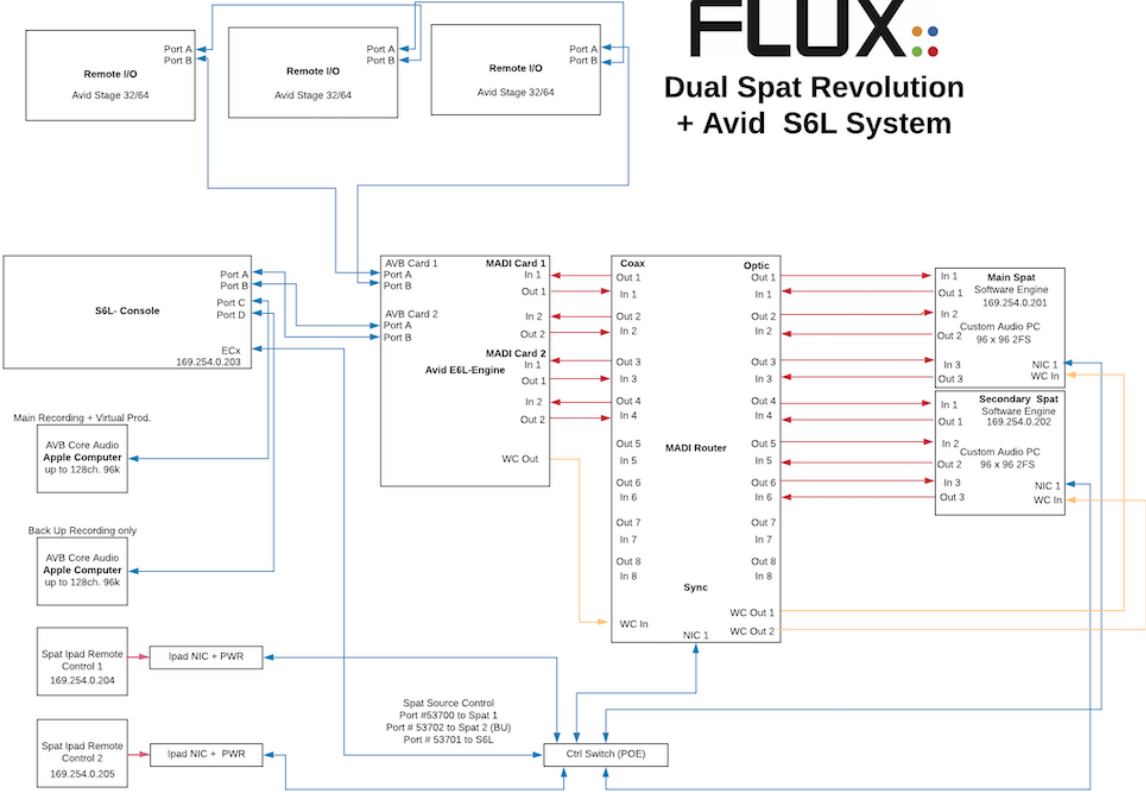


SPAT plugin on S6L CKM module – Page 1 of 3

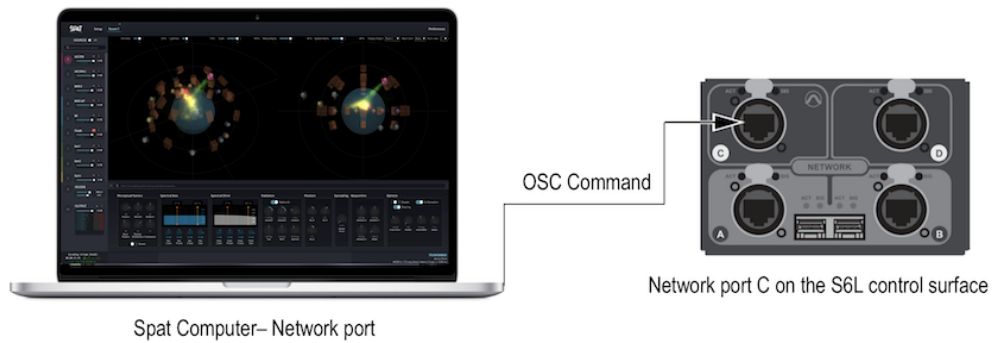
The integration will ultimately include two parts: the control portion and the audio portion. Our control will always be using OSC and will be communicating via either the ECx port of the console or via the AVB network interface (port C/D). You can choose the network interface the control is flowing in. The audio routing portion can take different scenarios. An example could be using the AVB audio entity capability of the S6L system integrated into a Mac-qualified computer. In this first case, audio and control will flow on the same network port (Port C/D on the console). As this AVB is often used for other uses (virtual production), people will frequently rely on legacy MADI integration to leave the AVB entity dedicated to Recording and playback scenarios. New scenarios will be possible shortly as the new Avid MLN-192 (Milan AVB) expansion card becomes available and qualified interfaces join the Milan party. If the route of the MADI integration is taken, then control will happen from the EXc port on the console. This makes it as well a cross-platform solution for Windows or Mac.

FLUX

Dual Spat Revolution + Avid S6L System



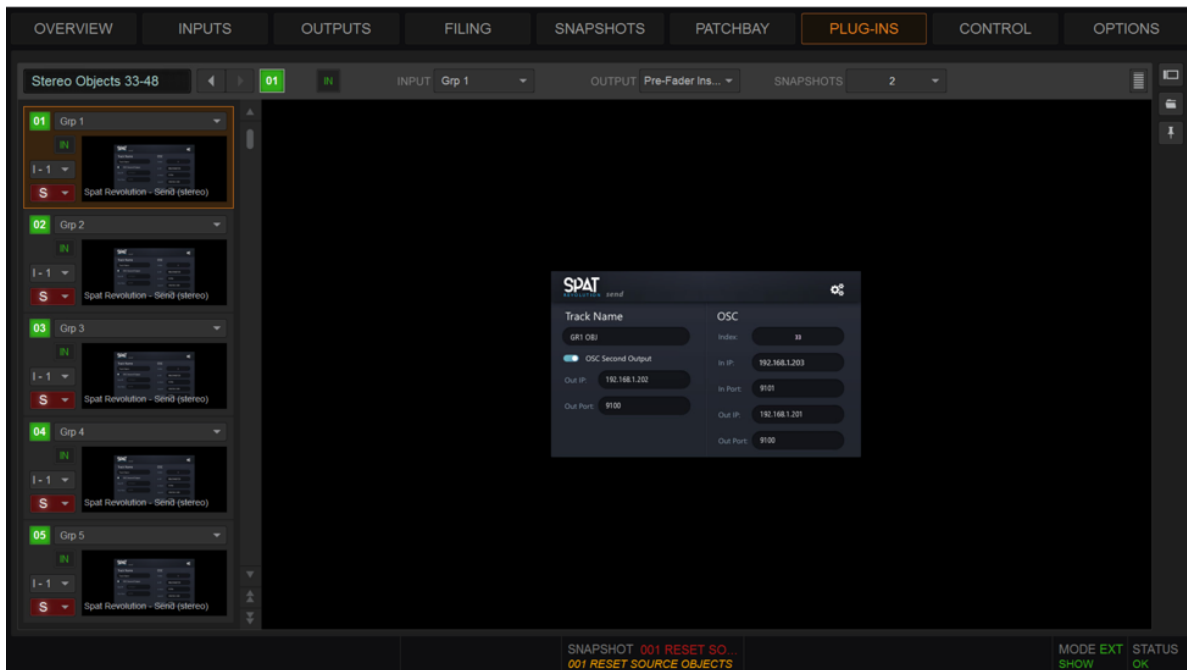
Dual SPAT Revolution integration to S6L via MADI



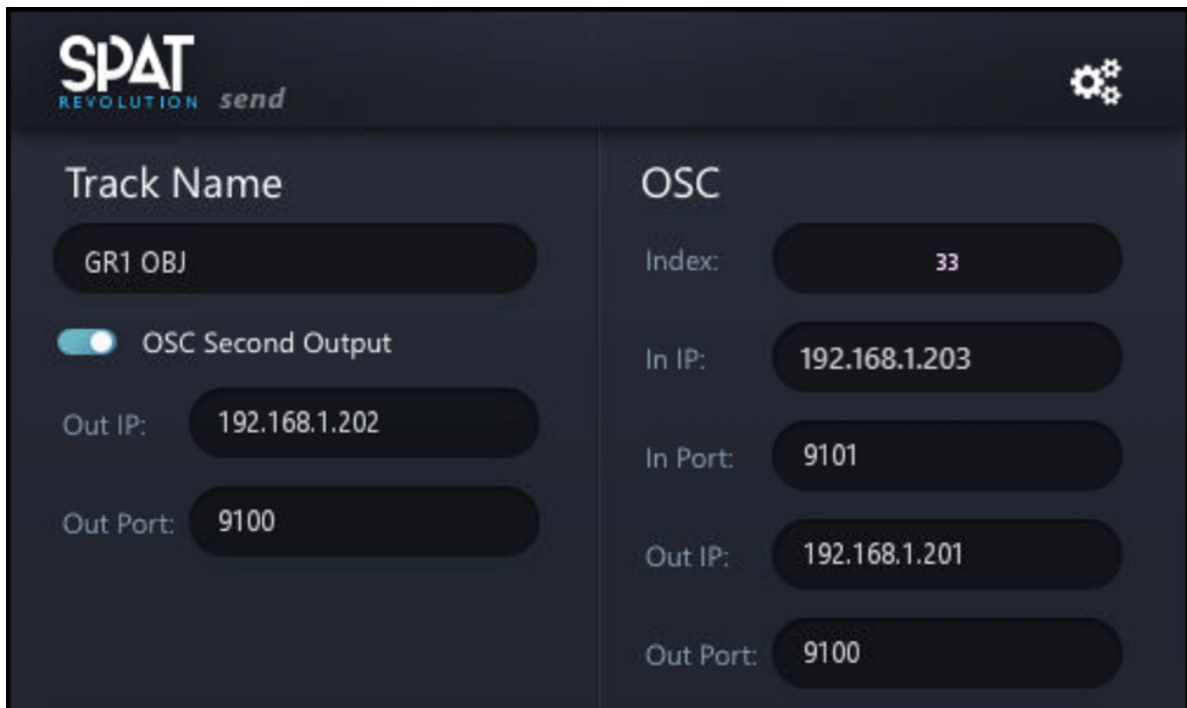
SPAT Revolution integration to S6L via the AVB port for audio and control

48.1 Configuring OSC

Now let's look at configuring this integration's control (OSC). The OSC settings (SPAT send plugin and *SPAT Revolution*) are specific to your console IP address and the *SPAT Revolution* preferences.



SPAT Send plugin in the Plugins rack of VENUE



SPAT Send plugin setup

The PI interface doesn't have much and is straightforward:

S6L doesn't populate the track name when inserting it into a channel. That said, if you are connected to *SPAT Revolution*, changing the track name to your desired name will rename your source/object name in *SPAT Revolution* software! Names are part of snapshots as well. The below SPAT / S6L templates have all the start naming set!

- Index refers to the object/source number in *SPAT Revolution* application. This gets automatically generated every time a new instance of SPAT sends plugin is instantiated in the console plugin rack. Snapshots can change it, but it is unique, and no instance can take over a currently used index. As the Index number is part of snapshots, be careful not to change them after creating or using them.
- In IP: The pull-down menu will allow you to choose the ethernet interface with which you will be listening to OSC commands coming from *SPAT Revolution*. You will need this address for setting up SPAT preference later. This can be the AVB port 169.254.x.x, or your ECx port network interface of the console, depending on the integration route. In the above example, the ECx interface with IP 192.168.1.203 is chosen.
- In Port#: This port #9101 is already set for you and matches the corresponding Port# used by the *SPAT Revolution* OSC preferences / Avid S6L preset. It can be left to default unless it conflicting with other OSC traffic.
- Out IP: This IP address is the manual address (above example 192.168.1.201) in which your Primary *SPAT Revolution* computer is configured.
- Out Port#: This port #9100 is already set for you and matches the corresponding Port# used by the *SPAT Revolution* OSC preferences / Avid S6L preset. It can be left to default unless it conflicting with other OSC traffic.
- OSC second output activates OSC messages to a second SPAT computer (such as your backup/ redundant computer engine). Output IP is the destination IP, so it is the IP of the second SPAT computer. The same port# as the primary is already set for you and matches the corresponding Port# used by the *SPAT Revolution* OSC preferences / Avid S6L preset. Note that the backup computer is not bidirectional to S6L and will only receive messages. So, the SPAT Primary computer is the one that is bi-directional (so updating plugins live with movements so you can take a snapshot of it).

i Note

Note that any changes you make to properties to the SPAT Send interface get done across the board (across all PI instances, except the unique index).

48.2 Setting the preferences of the SPAT Revolution Application

Let's now setup your preferences on *SPAT Revolution* preferences page:

- Hit preference in the top right corner. Look for the OSC Main section. First, you want to make sure the checkbox OSC Enable is engaged.

SPAT OSC Main Preferences

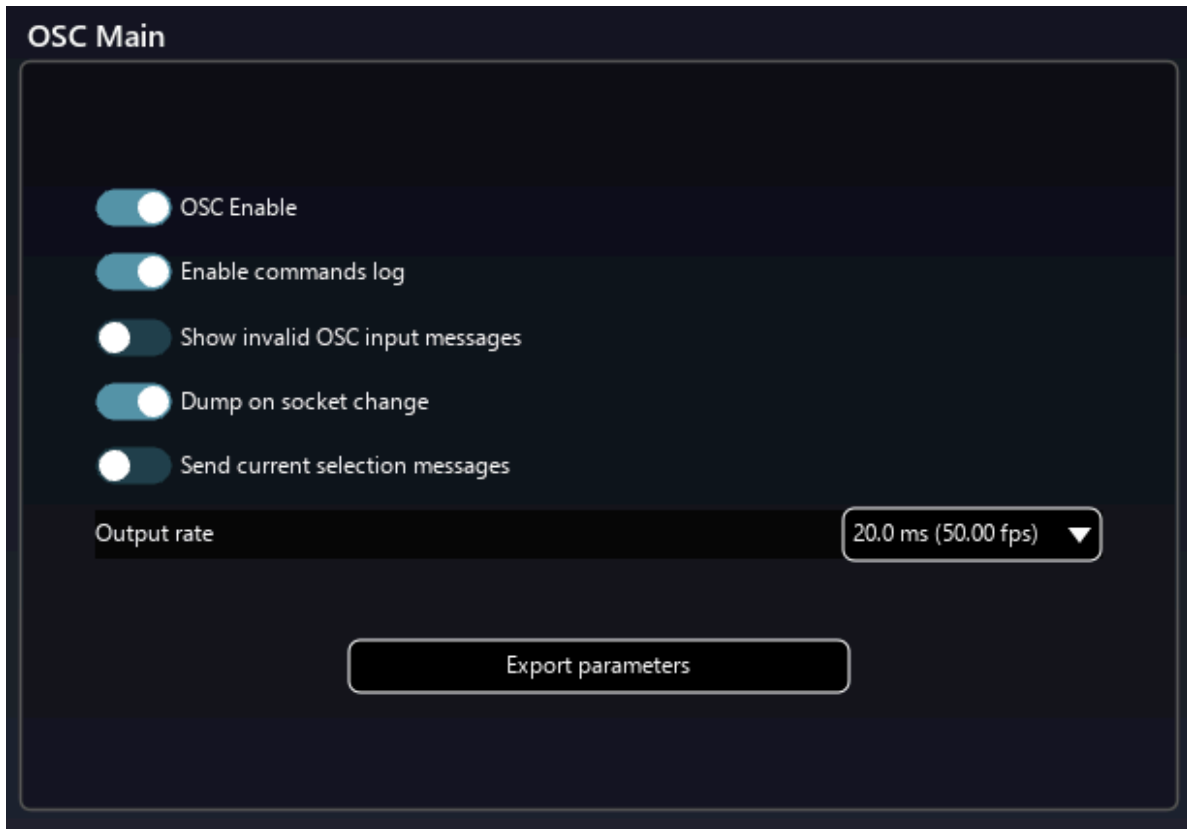


Figure 48.1: SPAT OSC Main preferences

- In the OSC Connections section, you will see 8 OSC connection slots. Set one connection to Input Avid S6L and one to Output Avid S6L. We will use two slots for this. The output will be to go to S6L SPAT Send Plugin; the input will be to listen to S6L in *SPAT Revolution*.

SPAT OSC Connections Matrix

48.2.1 SPAT OSC Connections preferences

- Set the In IP address by pulling then the menu and choosing the network interface of your SPAT engine computer (for example, the 192.168.0.201 of whatever network

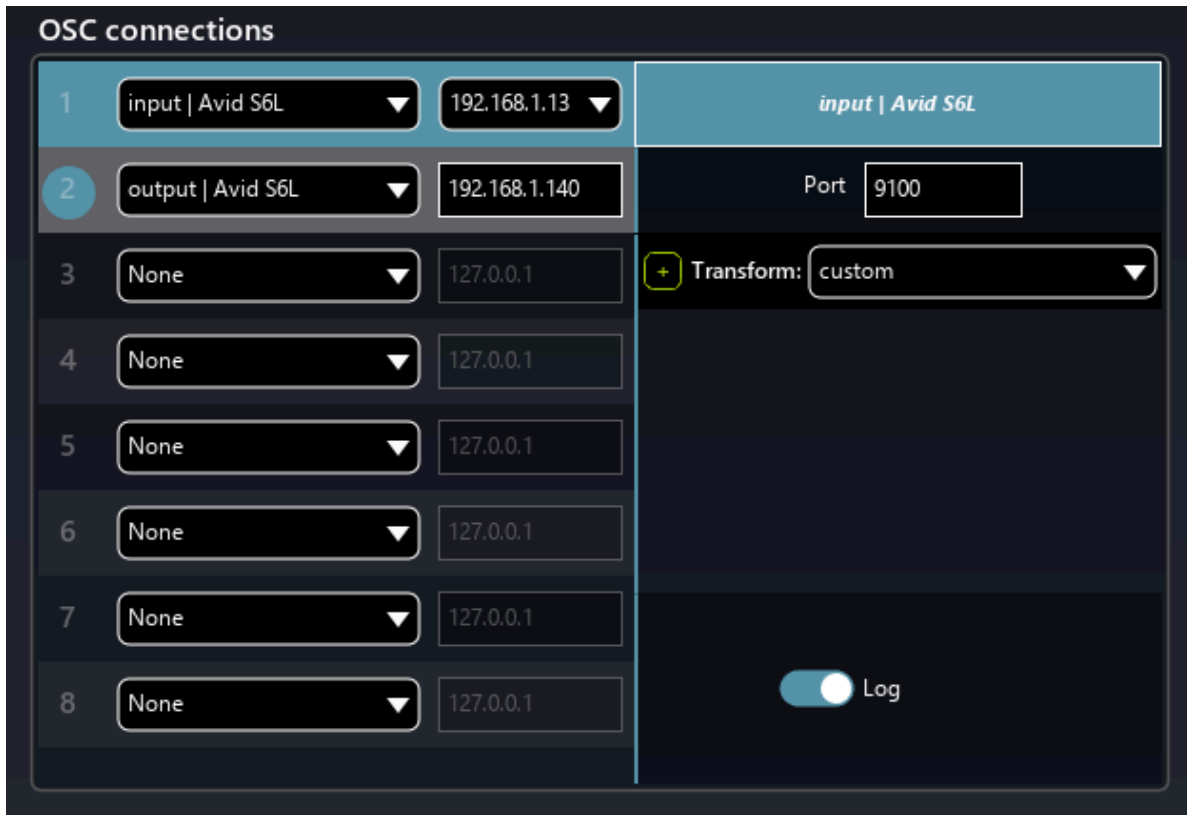


Figure 48.2: SPAT OSC Connections Matrix

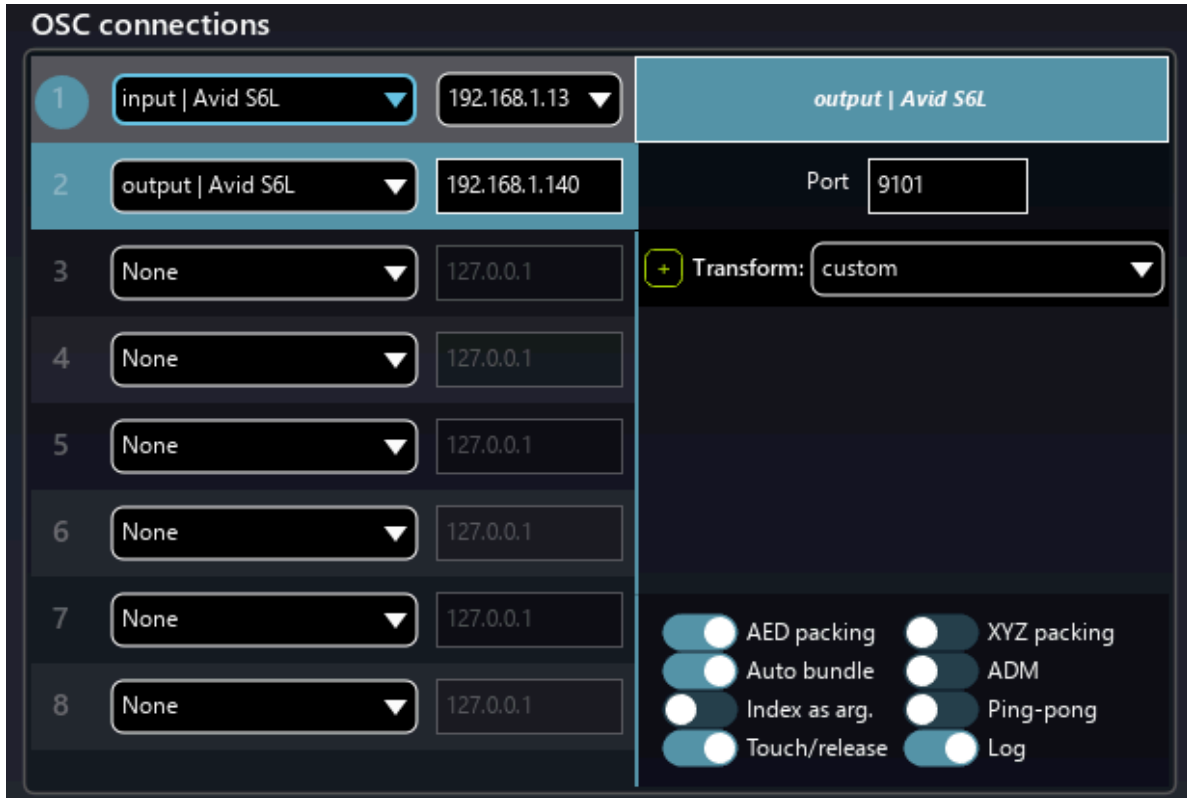


Figure 48.3: SPAT OSC Connections Matrix

interface/IP you are using for this communication as long as it is in the same network range as the console IP address).

- The input side (receiving the OSC parameters from the Out IP of the console) doesn't need anything specific as the OSC input is ready for Avid S6L. You simply need to choose the **Input | Avid S6L** preset. As mentioned above, in the S6L setup, the port # is predefined to port #9100 on both sides (PI and SPAT OSC preset for S6L).
- On the OSC output connection, you will simply need to choose the **Output | Avid S6L** preset and put the destination IP of the console, see above 192.168.1.203. The port # is already predefined (9101) and matches the Avid SPAT Send PI Input. Port #9101. This IP address is the IP address found in the PI interface (usually the ECx port IP setting). The default output option checkboxes are set with the preset for S6L.

i Note

If desired, the OSC Log can be seen for testing. In the SPAT OSC Main section, you can use the **Enable commands logs** option to confirm OSC communication. Pressing **Shift + F7** or going to the help menu of SPAT will give you the option for a mini log window to see if the traffic is flowing as you move a source either in SPAT software or on the parameters on the PI encoders. It is not recommended to leave command logs enabled past the confirmation testing step. This will ensure that you don't take up resources.

48.3 Configuring Audio Connections

Let's now look for the Hardware I/O connection. This is where you will configure the hardware input and output (audio interface) for *SPAT Revolution*.



SPAT Hardware IO (Audio) preferences – AVB example

- In Devices, please select your Core or ASIO audio device. In this example, we are using the AVB Core Audio E6 Engine, but this could also be your MADI interface of choice, such as an RME MADIface, MADIface XT, Soundgrid MADI device, or any preferred interface. Not that the choice of interface will impact your overall possible system latency combined with the computer hardware.

The sample rate should be set at 96000 Hz (unless the sample rate converts from the S6L to 48K), and the buffer can be set to your desired buffer (mind lower buffer requires a qualified computer and optimized resource).

Congratulations, you are done! You're ready to go to the setup page in SPAT and configure your Spatial audio system.

48.4 Templates

As of *SPAT Revolution* version 20.12.0, we provide our users with three different start templates to help them build their session or simply review an example of *SPAT Revolution* integration. Our latest release has updated them, built with S6L v6.3 and qualified/tested for S6L V7.

To note that show files built with SPAT Send version 48020 or 48049 can't use SPAT Send 20.12.0. Version 48049 is the version to use for these older showfiles. That said, *SPAT Revolution* 20.12 release works well with these older sessions (nothing prevents a user from updating to the latest *SPAT Revolution* just keeping the old SPAT Send version for show file (snapshot) compatibility.

[SPAT_S6L_V7_64MONO_20_12](#)

SPAT_S6L_V7_64MONO_20_12 is the exact same template file we provided before. For the model, all channels are being sent to the spatial renderers. Channel 1-64 have a SPAT instance inserted, and its direct out post-fader signal is assigned to MADI card 1, 1-64, and in parallel connected to the PTAVB entity channel 65-128 (for alternative routing that some have used). MADI 1-2 is patched to Stereo channel 65 as a means of returning the binaural room (and headphone mix) coming from the SPAT template session (.JSON file). The S6L file contains three snapshots: - **Reset all**, resetting all sources to default position without reverb (acoustic simulation) and two other test snapshots called - **Circle and Horizontal Line**. Plugins are set with a 2 second interpolation time, showing how time value is used to smooth out transitions. The accompanying SPAT session file contains the 64 sources/objects connected to a Binaural Room as a starting point.

[SPAT_S6L_V7_32MONO_16ST GROUPS_20_12](#)

SPAT_S6L_V7_32MONO_16ST GROUPS_20_12 contains an S6L file with 32 Mono SPAT send sources/objects ready to be inserted on the channels you are using as direct out to *SPAT Revolution*. These are name source/object 1-32. The file is all well-ready with 16 Stereo groups for your source/object stems (33-48). Two test snapshots are preconfigured to **reset all source positions** and one as a snapshot test recalling all sources in a **horizontal line**. Plugins are set with a 2-second interpolation time, showing how time value is used to smooth out transitions. The accompanying SPAT session file contains the 48 sources (32 mono and 16 stereos) connected to a Binaural Room as a starting point (it's a 64-channel audio setup). The audio routing is not applied yet on this file, but the post-fader pick-off point option is set to be ready to route direct-out post-fader out of the channels.

[SPAT_S6L_V7_8MONO_8ST GROUPS_20_12](#)

SPAT_S6L_V7_8MONO_8ST GROUPS_20_12 is a basic entry S6L file with eight mono SPAT send objects ready to be inserted on the channels you are using as direct out to SPAT. These are source/object 1-8. The file is all well-ready with 8 Stereo groups for your source/object stems (9-16). Two test snapshots are preconfigured to **reset all source positions** and one as a snapshot test recalling all sources in a **horizontal line**. Plugins are set with a 2-second interpolation time, showing how time value is used to smooth out transitions. The accompanying SPAT session file contains the 16 sources (8 mono and eight stereos) connected to a Binaural Room as a starting point (it's a 24-channel audio setup). The audio routing is not applied yet on this file, but the fader pick-off point option is set to be ready to route direct-out post-fader out of the channels.

49 Setting up DiGiCo console with SPAT Revolution

Templates can be found here: [DiGiCo v1280 - SPAT Revolution 20.12 templates](#)

- Before starting the SPAT Revolution software, make sure you have connected your Ethernet cable between the mixing desk and the computer running SPAT. Ensure you set the correct IP settings for your local network card. In our example, we are using 192.168.0.200.
- Start the SPAT Revolution software and open the preference page.
- Enable OSC.
- In the OSC Connections section, use the pre-configured DiGiCo OSC presets. In the pull-down menu, choose **input | DiGiCo** and select your local IP address to communicate with the desk. Then, set a second OSC Connection for the output from SPAT Revolution to the desk. In the pull-down menu, choose **output | DiGiCo**. You will then enter the desk IP address that we will see in the following steps.

49.1 Configuration and Templates

The DiGiCo template files were created on a DiGiCo SD series using version 12.2.1280. File versions are provided for all the different console models DiGiCo makes (user can do their own conversion as well). The session files contain all the custom OSC configurations, but you need to set up the external control from the desk to SPAT Revolution, as this is specific to the hardware. *This is a one-time setup for each specific console.*

- Go to the **Setup / External Control** section of the desk.

You will want to ensure **Enable External Control** is set to **YES** and the **Suppress OSC retransmit** is **active**. You will be using OSC generic for channel controllers. You are now ready to **add device** to configure the OSC connection. It is considered **other OSC** .

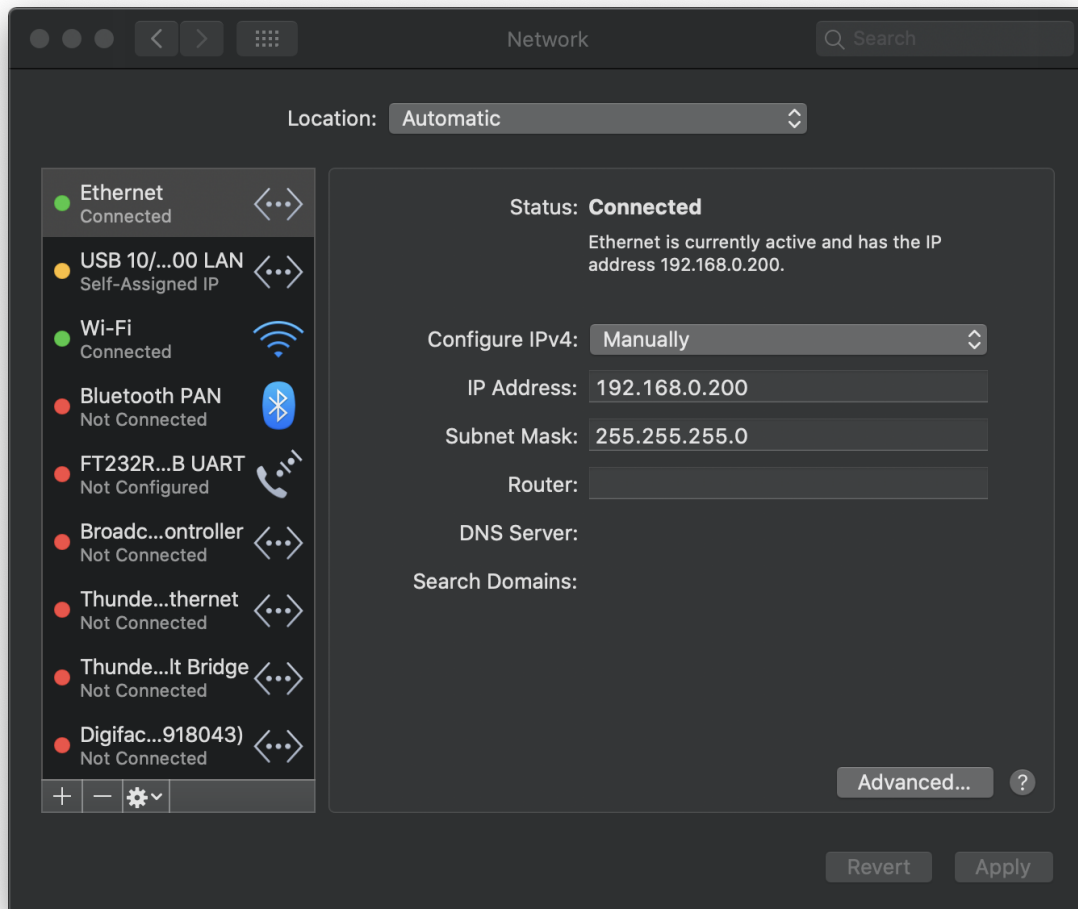


Figure 49.1: Network

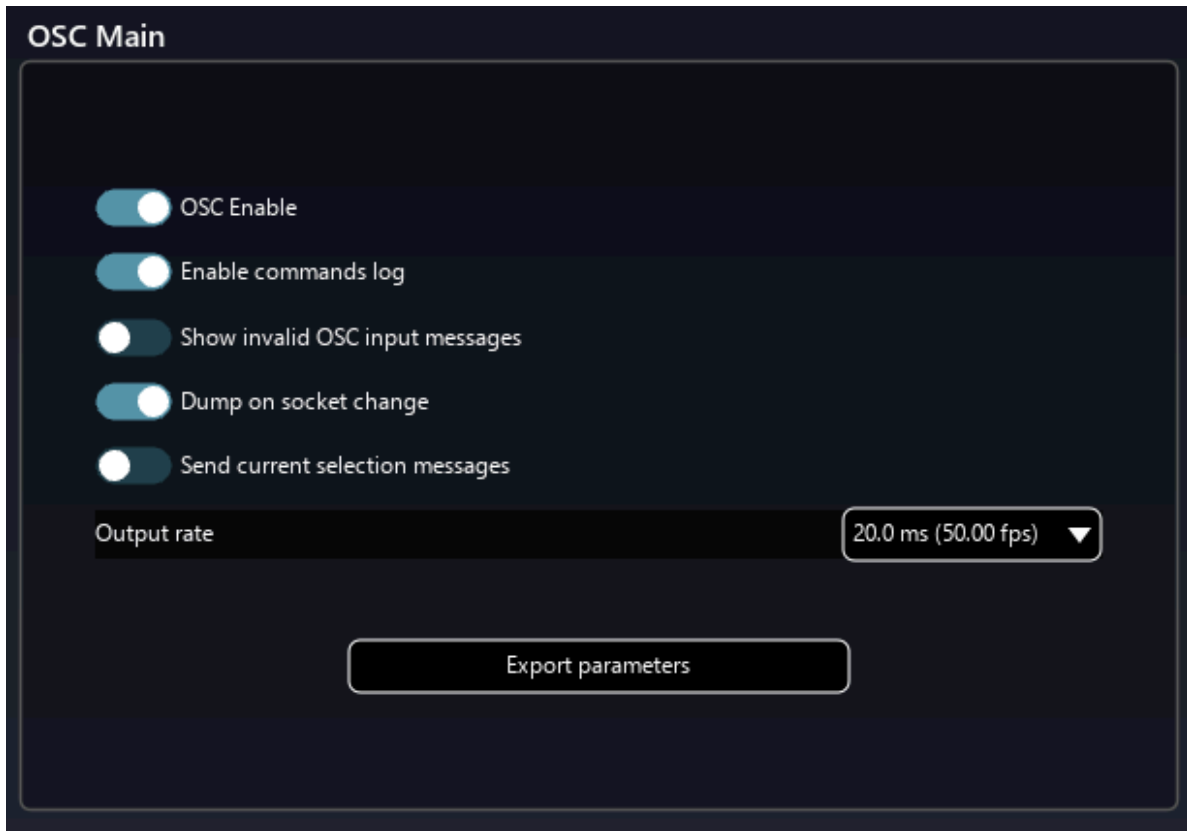


Figure 49.2: Enable OSC

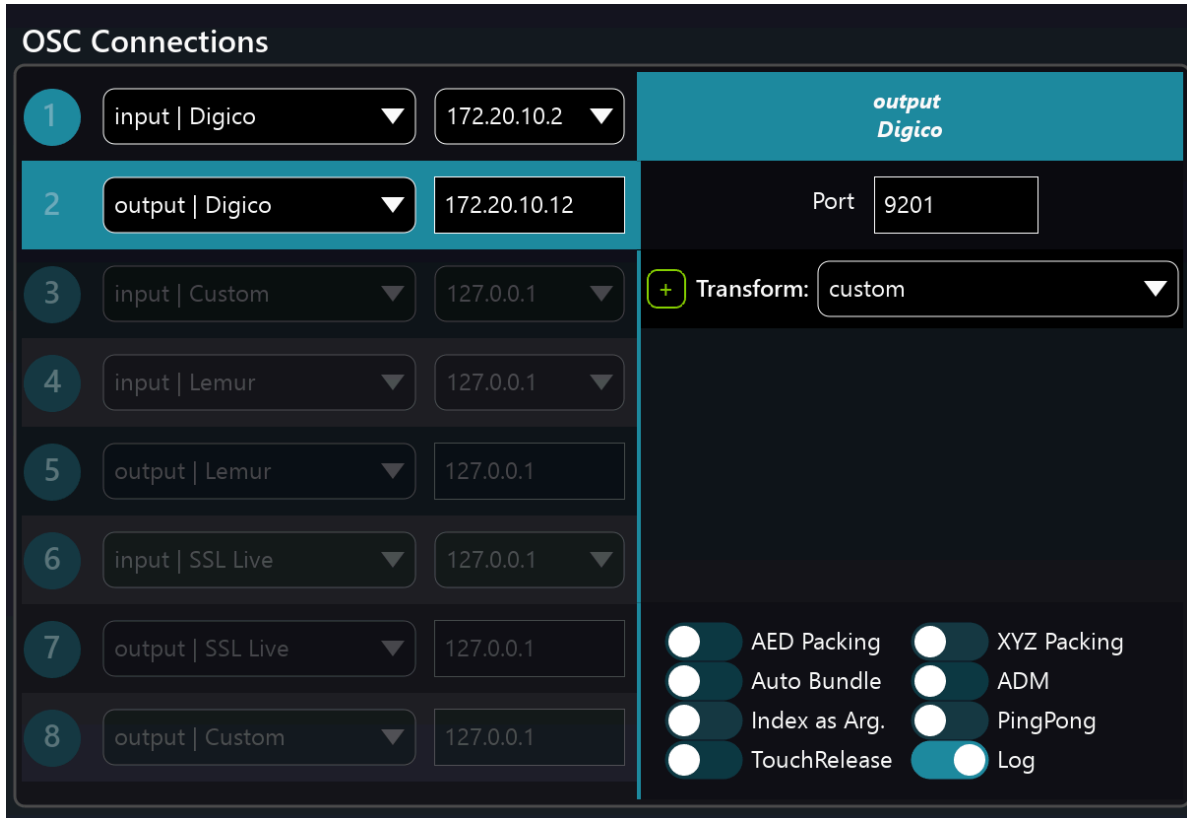


Figure 49.3: OSC Connections DiGiCo Presets



Figure 49.4: Network

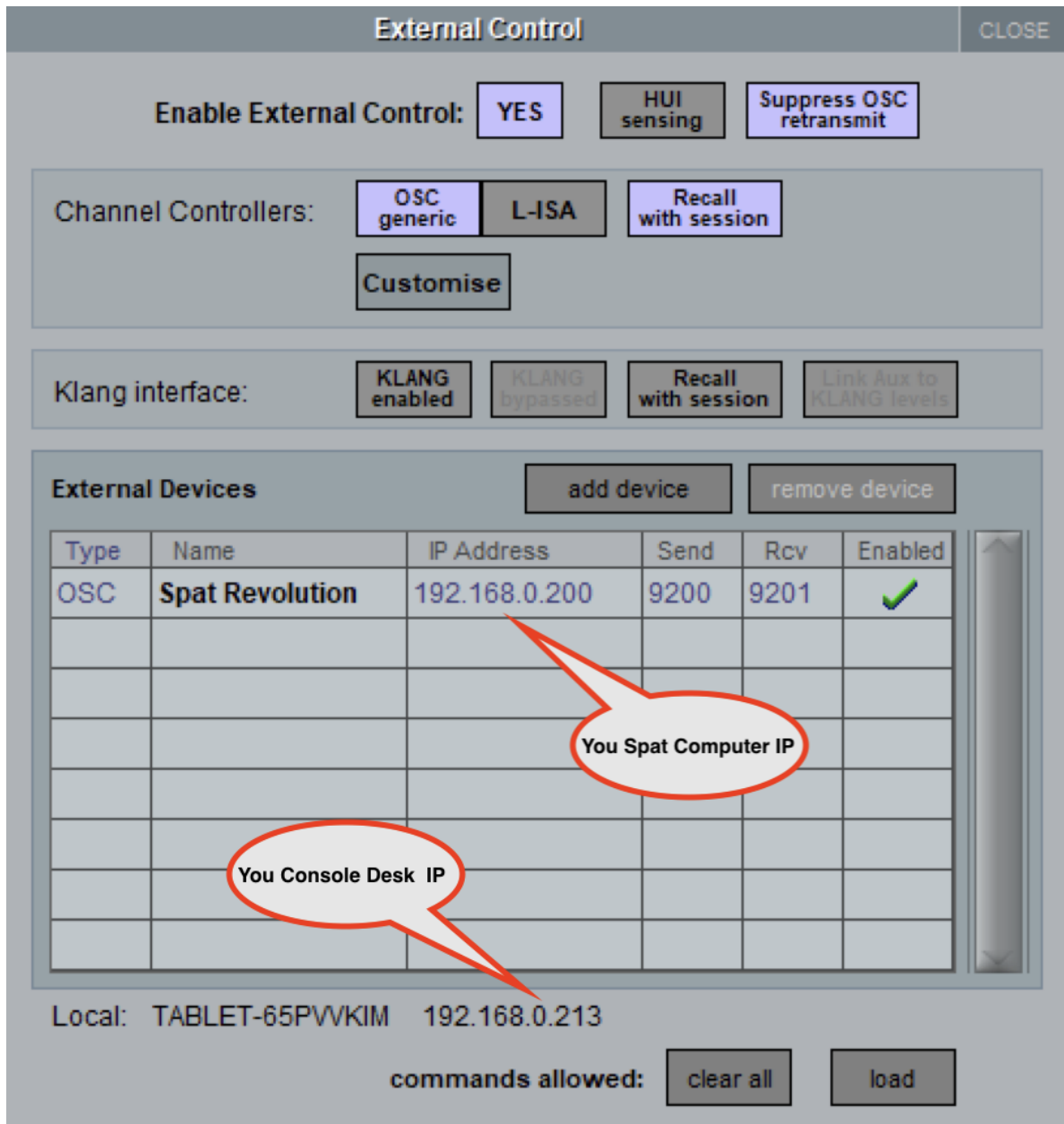


Figure 49.5: Network

- Press **add device**, and give your device a name: for example, SPAT Revolution. Enter the IP address of your SPAT Revolution computer: in this example, 192.168.0.200. For port #, these are already configured in SPAT Revolution in the DiGiCo preset. On your External Device port on the DiGiCo, set the **Send** port to 9200, **Rcv** Receive port is 9201. Make sure that you enable the OSC device with the green check!

i Note

Settings must be done individually because they are not part of a session file. These are global settings of the desk - they stay even when you're loading a new session file.

Now, you are ready to load the template session provided. Two session models are provided. One is for using SPAT control on all input channels; the other is for using it on groups. **Group template** is the one to send audio to SPAT Revolution from group stems.

After your file is loaded, you can see, for example, how to access these parameters on the desk.

Select an Input channel and open the channel's output page. Select **external control**. It will then open the external control tab. Here, you will find the predetermined most common parameters for the SPAT Revolution. A user could decide to change these if needed (for example, if rather wanting to use XYZ commands).

Configured in the templates are:

- Rotaries:
 - 360 Pan (Azimuth)
 - Distance (0-10M, default 2M like SPAT)
 - Elevation (-90 / 90 degree)
 - Presence (Direct Source presence)
 - Spread (0-100% Spread)
 - Scale (To scale stereo objects)
 - LFE/Aux (Your 0.1 system aux send. The LFE)
 - Room P (The presence of the room reverb/tail)
- Switches:
 - Select (to select the source, by default, it will be selected when opening the External Control pane)
 - Solo
 - Mute
 - Early (To turn on/off the localized early reflections - default on)
 - Cluster (To enable/disable cluster - default on)
 - Tail (To turn on/off the reverb diffused tail - default on)
 - Reverb (To completely enable/disable the reverb for this source - default on)



Figure 49.6: Network



Figure 49.7: Network

DiGiCo OSC Generic External Control - Rotaries

DiGiCo OSC Generic External Control - Switch

49.2 Audio routing

Your audio routing will happen either from you sending the post-fader channel direct out or the group outputs to MADI, for example, and receiving it in SPAT Revolution using a USB MADI interface (or any qualified MADI interface). These audio inputs will feed your sources/objects.

49.3 Useful tips for integration

Important: The SPAT Revolution's source/object index numbers match the physical channels or group (1 to x) of the DiGiCo desk. This means channel or group 1 on the desk will be source/object index 1 in SPAT Revolution. The same applies to a stereo channel or a stereo group. It is a single index for a stereo element. The communication between SPAT Revolution and DiGiCo is bidirectional - so when moving an object in *SPAT Revolution*, you will see that the parameters are changing at the external control window of the desk and vice versa.

You can use console snapshots to create your various scenes/cues. When recalling snapshots, the external parameters will be recalled as well. DiGiCo brings a crossfade time for these snapshots, which means interpolating from one scene to the other. This is a powerful function to do crossfades.

External Control
CLOSE

Enable External Control: YES HUI sensing Suppress OSC retransmit

Channel Controllers: OSC generic L-ISA Recall with session

Customise

Klang interface: KLANG enabled KLANG bypassed Recall with session Link Aux to KLANG levels

RESET WIDTHS
External Channel Controllers OSC
CLOSE

INPUT CHANNELS
 GROUP OUTPUTS

rotaries
switches
rotaries
switches
use asterisk (*) in place of channel number

	Label	OSC	Min	Max	Default
1	360 Pan	/source*/azim	-180	180	0
2	Distance	/source*/dist	0	10	2
3	Elevation	/source*/elev	-90	90	0
4	Presence	/source*/pres	0	120	100
5	Spread	/source*/spread	0	100	0
6	Scale	/source*/scale	0	10	1
7	LFE/Aux	/source*/lfe	-144.5	24	0
8	Room P	/source*/prer	0	120	48

commands allowed: clear all load

Figure 49.8: DiGiCo OSC Generic External Control - Rotaries

CLOSE

External Control

Enable External Control: YES HUI sensing Suppress OSC retransmit

Channel Controllers: OSC generic L-ISA Recall with session

Customise

Klang interface: KLANG enabled KLANG bypassed Recall with session Link Aux to KLANG levels

CLOSE

External Channel Controllers OSC

rotaries
switches
rotaries
switches

use asterisk (*) in place of channel number

	Label	OSC	Min	Max	Default
1	Select	/source*/select	0	1	1
2	Solo	/source*/solo	0	1	0
3	Mute	/source*/mute	0	1	0
4			0	1	0
5	Early	/source*/early	0	1	1
6	Cluster	/source*/cluster	0	1	1
7	Tail	/source*/tail	0	1	1
8	Reverb	/source*/reverb/enable	0	1	1

commands allowed: clear all load

Figure 49.9: DiGiCo OSC Generic External Control - Switch

! Important

Keep in mind that these are just global OSC recall/crossfade times. It's impossible to set individual crossfade times for each OSC parameter by itself.

Using the **Group template session**, you have the exact same behavior on your groups instead of your input channels. You can send either a single channel or multiple ones as a stem to these groups and control them as a single source/object, e.g., your drum set.

Then the fun begins! Feel free to add our Lemur or any other external touch control to the equation. The fact that you can, on the desk itself, send a select command for a source means that you can rapidly select the source, and using the external control set to the -1 source / selected source gives you control right away at your fingertips. (see OSC -1 select message).

50 Solid State Logic SSL Live Consoles

As of version 4.11, SSL now supports Open Sound Control (OSC). SSL Live consoles can now remotely control other devices over their connectivity network ports. This is a bidirectional integration for *SPAT Revolution*.

50.1 Template

A simple template is available for downloading, providing a starting point to configure this integration.

[Solid State Logic SSL Live v4.11 - *SPAT Revolution* 20.12 template](#)

50.2 Setting up SSL Live console

- Go to MENU/Setup Option /EXTERNAL CONTROL to configure remote control.
- Rename the Generic OSC to SPAT OSC and enter the SPAT Computer IP address.
- Set In port # to 9301 and Host RX Port # to 9300. They will match the SSL Live OSC Input and Output connection presets in *SPAT Revolution*.
- Make sure to Enable OSC and note the Control IP Address. This address will be needed when configuring SPAT OSC.

SSL OSC Settings

50.3 External control fader and switches

SSL Generic OSC - External Control

With Solid State Logic live consoles, up to eight *Fader* (i.e., continuously variable) and eight *Switch* parameters per third-party device channel may be defined for control by the console. These control parameters (OSC messages) going out to *SPAT Revolution* source/objects are accessible from each console audio path and dedicated OSC paths with no associated console

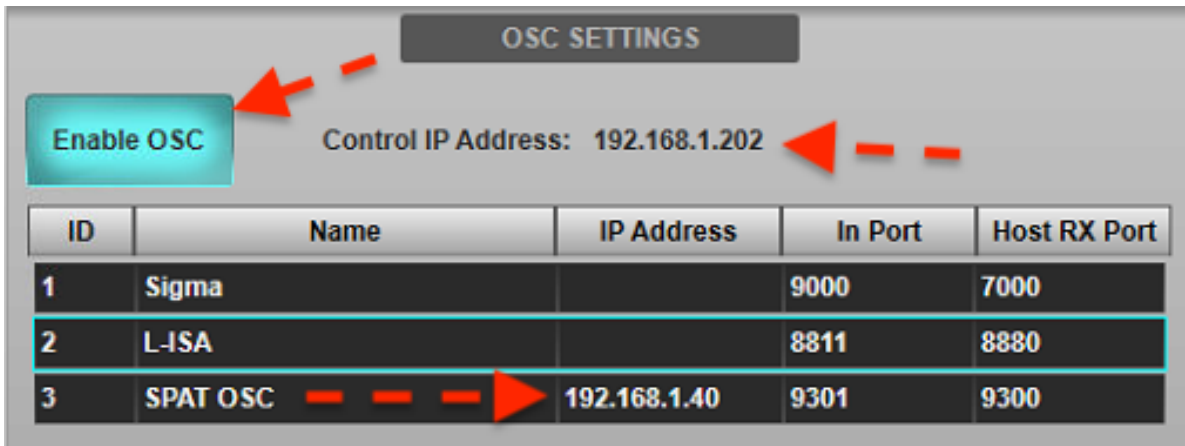


Figure 50.1: SSL Live SPAT OSC Switches

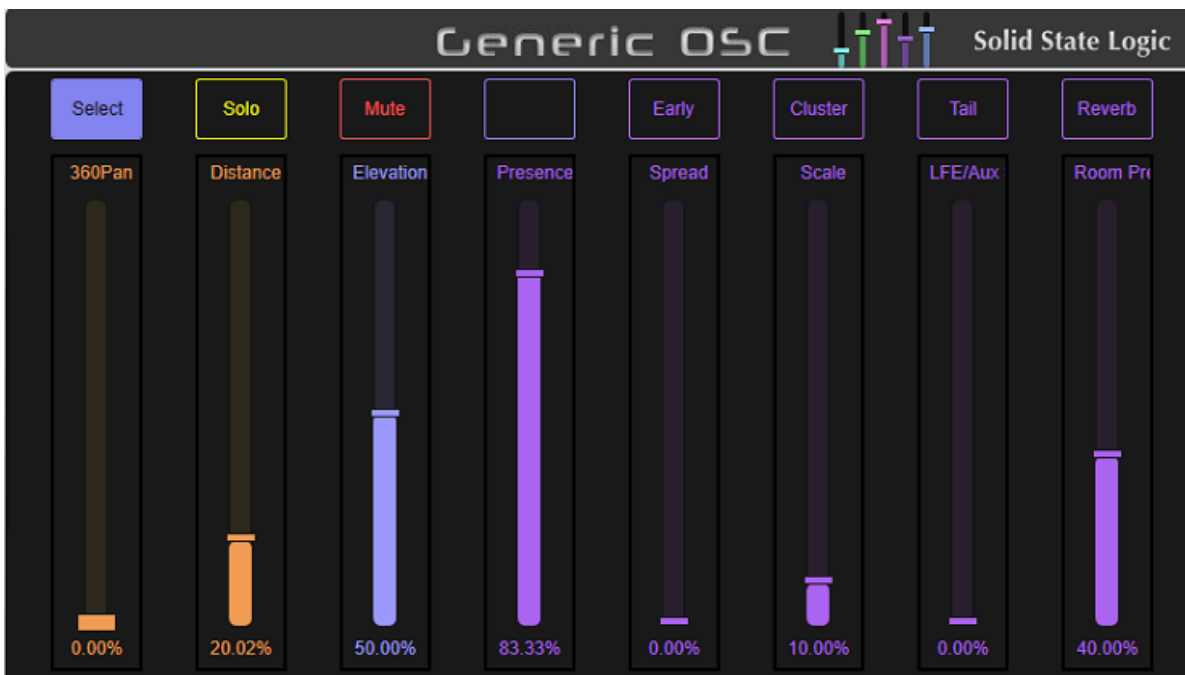


Figure 50.2: SSL Live SPAT OSC Switches

audio processing. The provided template has all these configurations done for you, but you can decide to customize with different or fewer parameters.

SSL OSC Device Configuration and Address for Switch

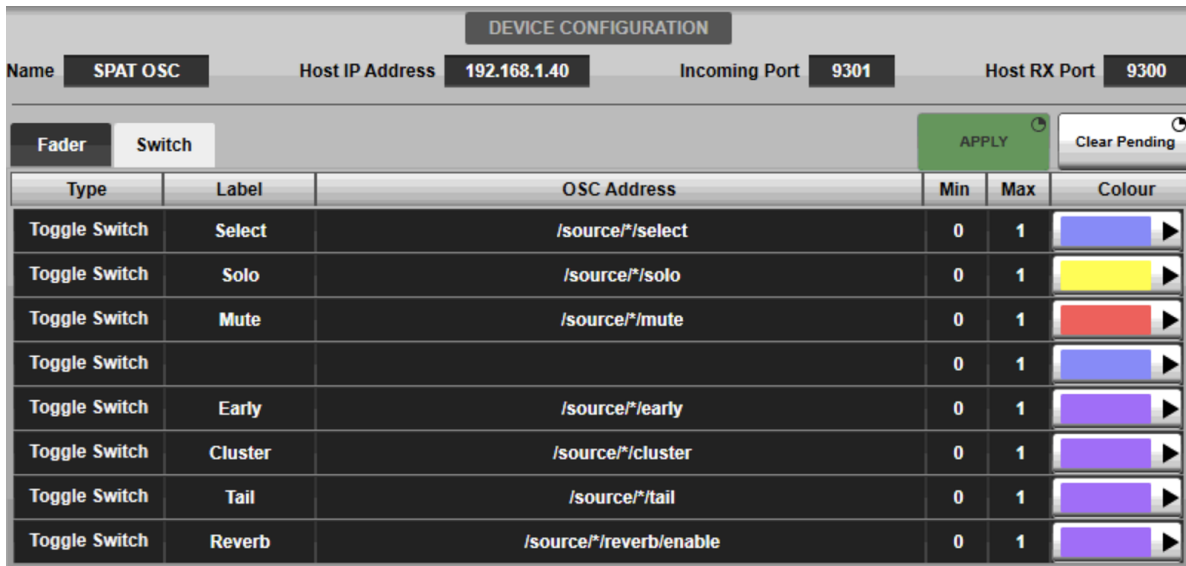


Figure 50.3: SSL Live SPAT OSC Switches

SSL OSC Device Configuration and Address for Fader

50.4 Setting up SPAT Revolution

- Start the *SPAT Revolution* software and open the preference page.
- Enable OSC.
- In the OSC Connections section, use the pre-configured SSL Live OSC presets. In the pull-down menu, choose **input | SSL Live** and select your local IP address to communicate with the desk. Then, set a second OSC Connection for the output from *SPAT Revolution* to the desk. In the pull-down menu, choose **output | SSL Live**. You will enter the desk IP address as found in the OSC settings. Ports # 9300 (Input) and 9301 (Output) are pre-configured.

DEVICE CONFIGURATION

Name **SPAT OSC** Host IP Address **192.168.1.40** Incoming Port **9301** Host RX Port **9300**

Fader **Switch** APPLY Clear Pending

Type	Label	OSC Address	Min	Max	Colour
Fader	360Pan	/source*/azim	-180	180	
Fader	Distance	/source*/dist	0	10	
Fader	Elevation	/source*/elev	-90	90	
Fader	Presence	/source*/pres	0	120	
Fader	Spread	/source*/spread	0	100	
Fader	Scale	/source*/scale	0	10	
Fader	LFE/Aux S	/source*/lfe	-144.5	24	
Fader	Room Presence	/source*/prer	0	120	

Path Number = *

Figure 50.4: SSL Live SPAT OSC Faders

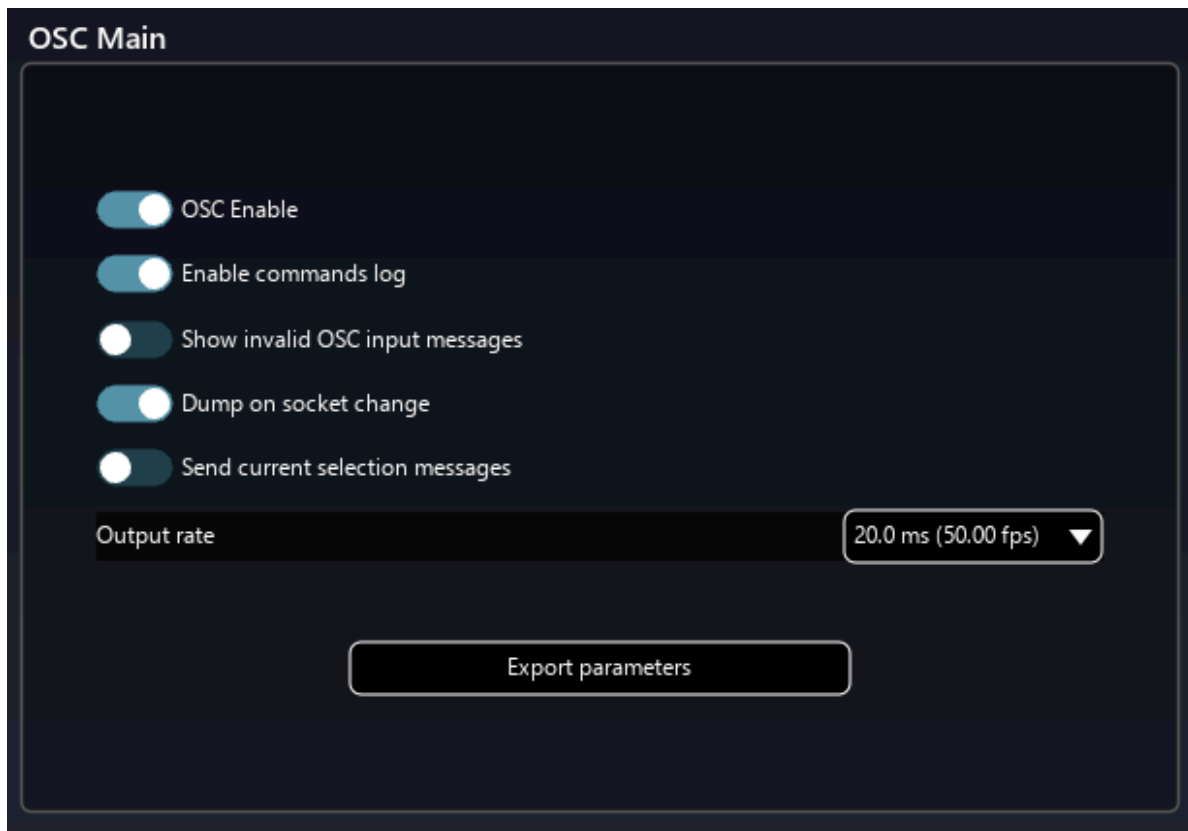


Figure 50.5: Enable OSC

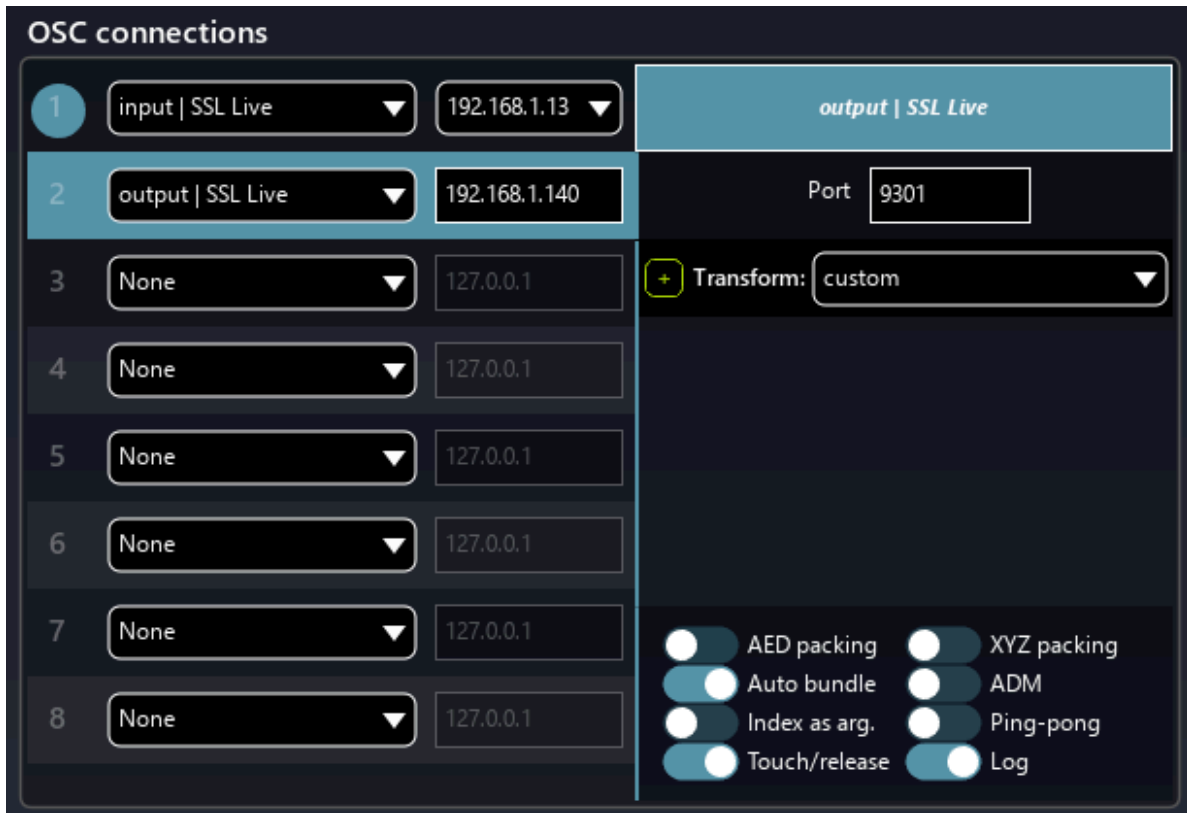


Figure 50.6: Enable OSC

51 SPAT Revolution Remote

Entirely supported by the recent release of SPAT Revolution 22.9 and powered by [Open Stage Control](#), a modular OSC controller/server, this new tool provides a web-based solution for remotely controlling *SPAT Revolution*. Being a web-based solution, it makes it compatible with various operating systems (OS), such as tablets and iPads, without excluding computers, even better with touch screens. It even supports simultaneous clients (remote browser) if need be.

The SPAT Revolution Remote communication adopts the basic [ADM-OSC](#) normalized grammar for source positioning while expanding it to most SPAT Revolution OSC control messages.

51.1 Installation

The first installation step is to download the resources. A config and a patch files are available for download via the [FLUX:: Center](#)

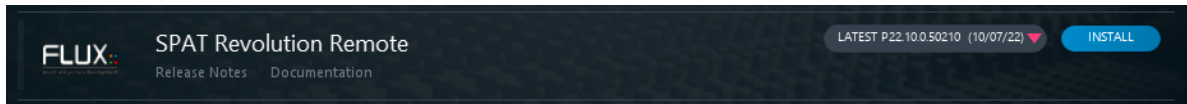


Figure 51.1: FLUX:: Center SPAT Revolution Remote

This installs the resources in:

- macOS: `Library/Application Support/Flux/SPAT Remote Server`
- Windows: `Program Files/Flux/SPAT Remote Server`

To operate this server, you will need to download and install [Open Stage Control](#), The application (server) that runs in the background and allows a web remote client access.

- [MacOS](#)
- [Windows \(64-bit\)](#)

51.2 Configuration

Once installed, you can simply run the Open Stage Control application and configure it. According to your system, a Windows or macOS config file is downloaded. This config file can be simply loaded to Open Stage Control by pressing the three dots on the right side of the interface.

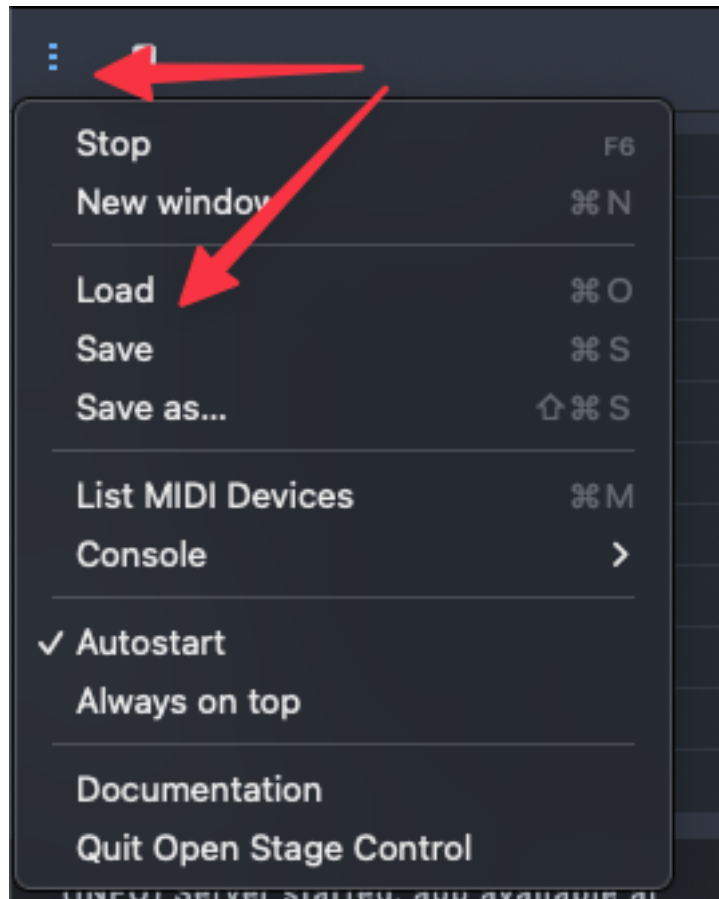


Figure 51.2: Load Config

Simply locate the config file in your folder:

- macOS: Library/Application Support/Flux/SPAT Remote Server
- Windows: Program Files/Flux/SPAT Remote Server

This config file will provide a default configuration with predefined OSC (9400 and 9401) and Web server (9410) port numbers.

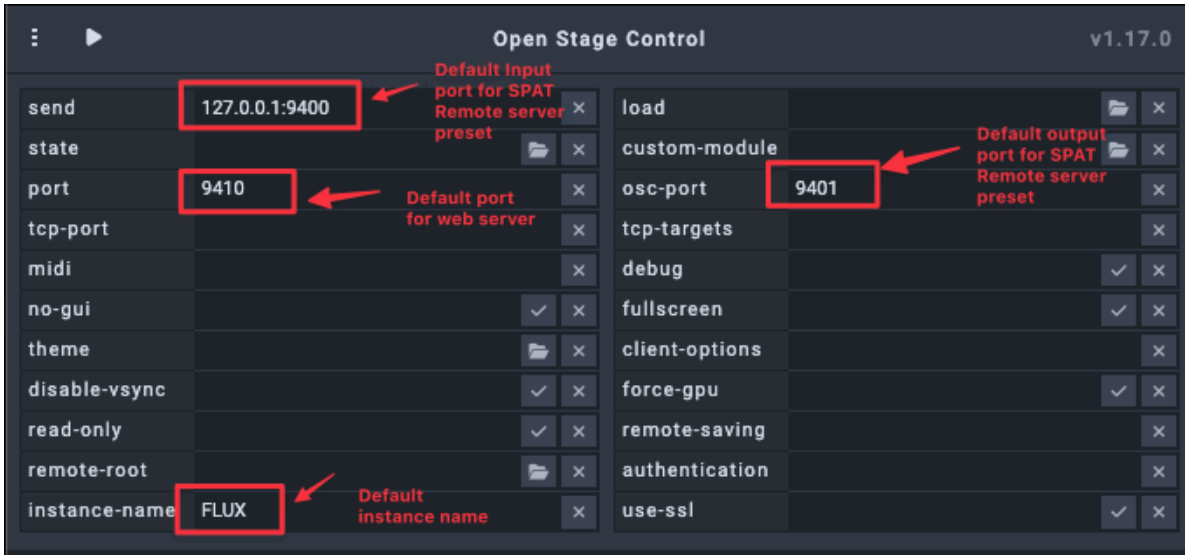
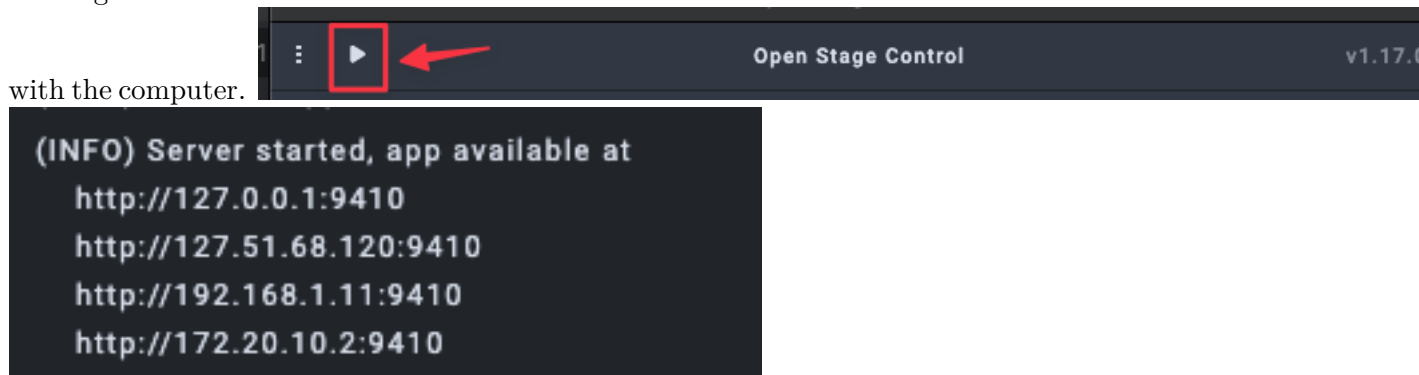


Figure 51.3: SPAT Revolution Remote Config

The last step is to run the server. Simply press the Play button, and the server will be up and running. The console will show all the client session info for all the available network interfaces



with the computer. By default, the actual user interface of the Remote control will appear on your computer. You can actually use it but can simply close it if you strictly intend to use it remotely.

If you would rather not see this interface every time the server starts, you can add `true` to the `no-gui` field. But you are serving any browser in all cases.

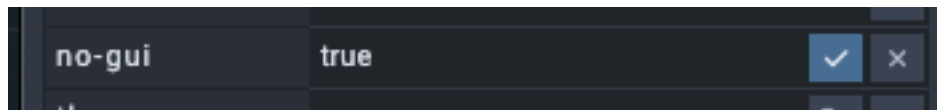


Figure 51.4: No GUI

Warning

Important: *Please ensure to open the server before opening SPAT Revolution.*

51.3 Autostart

You can optionally engage the autostart so the server will run every time Open Stage Control is started. You can also make Open Stage Control part of the launch of your computer so it will always be there running for you.

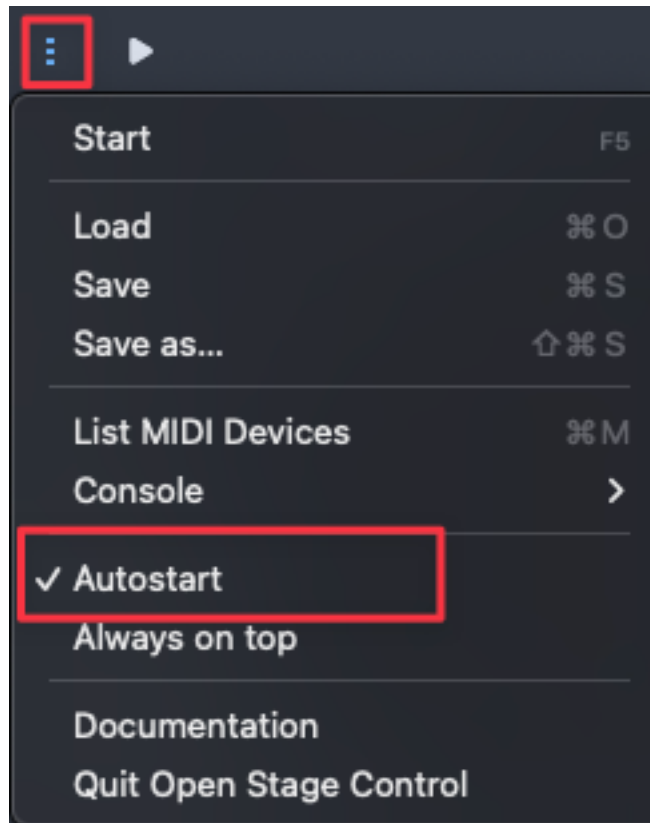


Figure 51.5: Autostart

51.4 SPAT Revolution configuration

Configuring SPAT Revolution requires to set the OSC Connection Input and Output ports for the SPAT Remote Server. Predefined presets simply need to be chosen for the input and

the output. These presets include the ADM-OSC transformation presets, which can scale the normalized positioning data to the desired range (the object positioning zone). By default, it scales to a scene of 10 m3 and a max distance of 10m.

- Access the SPAT Revolution preference page and reach the OSC Connection section.
- Add the `input | SPAT Remote server` preset

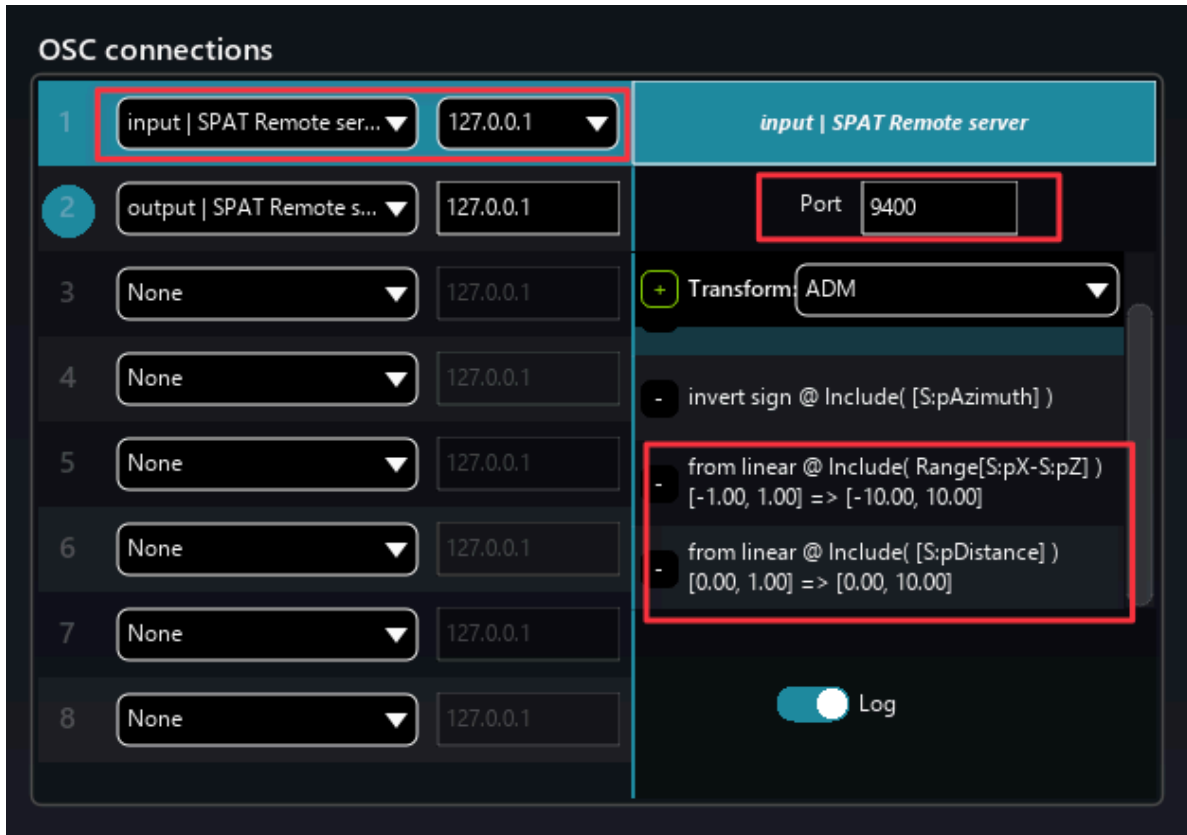


Figure 51.6: SPAT Revolution OSC Connection Input

- Add the `output | SPAT Remote server` preset

i Note

These above presets use the default ports and address of the supplied Open Stage Control Config files. By default, the local 127.0.0.1 address is set as it is assumed that the Remote server (Open Stage Control) is used locally on the same computer as SPAT Revolution. This is the most common scenario. Suppose your server is running on another computer. In that case, you can simply change the IP address on the server

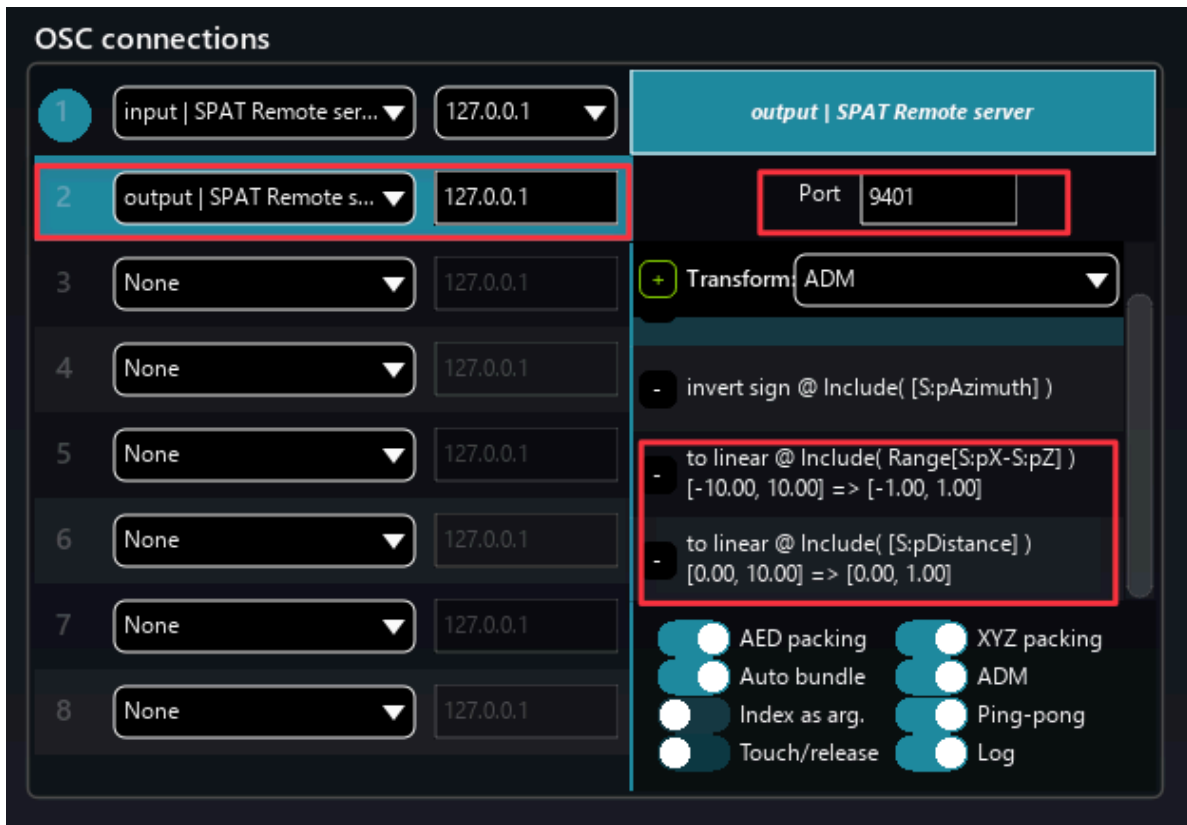


Figure 51.7: SPAT Revolution OSC Connection Output

before launching it, giving the IP of the SPAT Revolution computer Ethernet interface NIC.

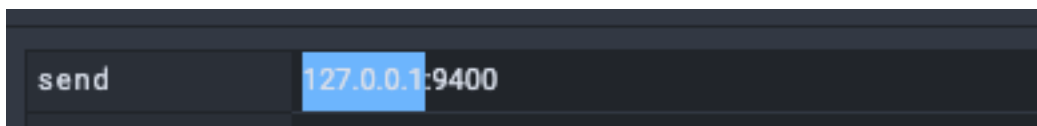


Figure 51.8: Changing the IP address in the server

51.5 Reaching the Remote interface from the browser

In the above example, we saw that after launching the server, we saw the addresses that the app was available at:

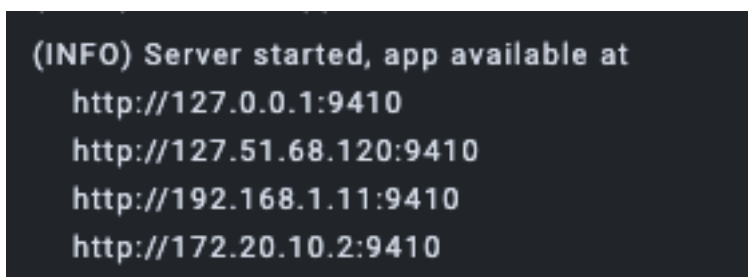


Figure 51.9: Server Info

This is basically giving you all the addresses of your computer's network interfaces that can reach the server as a client (in a browser), and reach the SPAT Revolution Remote. For example, the last one, the 172.20.10.2 represents the hardware connection of an iPad - yes, they form a network when you connect them to USB. This makes a more than ever robust remote instead of relying on Wi-Fi (which still works well).

With this, we can then turn to the iPad and start Safari. It is the preferred browser; simply dial the address to show where the server lies.

- <http://172.20.10.2:9410>

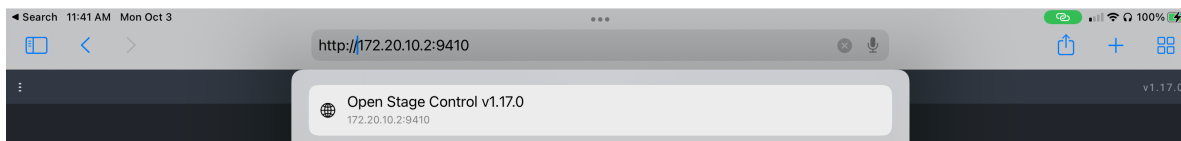


Figure 51.10: Reaching the server from iPad

51.5.1 Adding the SPAT Revolution Remote as a shortcut(Ipad)

If you want a shortcut on your iPad to reach the SPAT Revolution remote, you can do this with these simple steps. It is important to note that if you change the server's IP address, you will need to redo this step again.

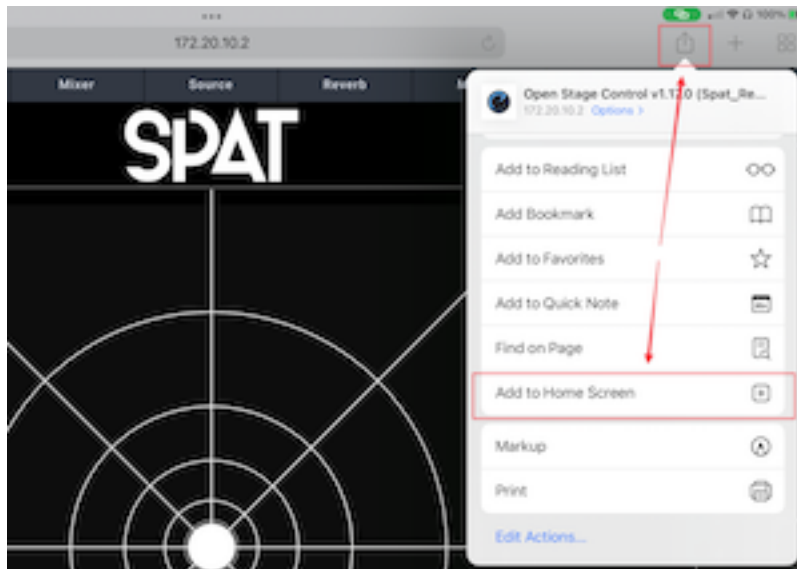


Figure 51.11: Adding Page to iPad Home Screen 1 25%

51.6 SPAT Revolution Remote page

Although most pages are pretty much self-explanatory, here is a review of the Main and Snapshot pages

51.7 Release Notes

51.7.1 22.10.0.50212

Release of the SPAT Revolution Remote package

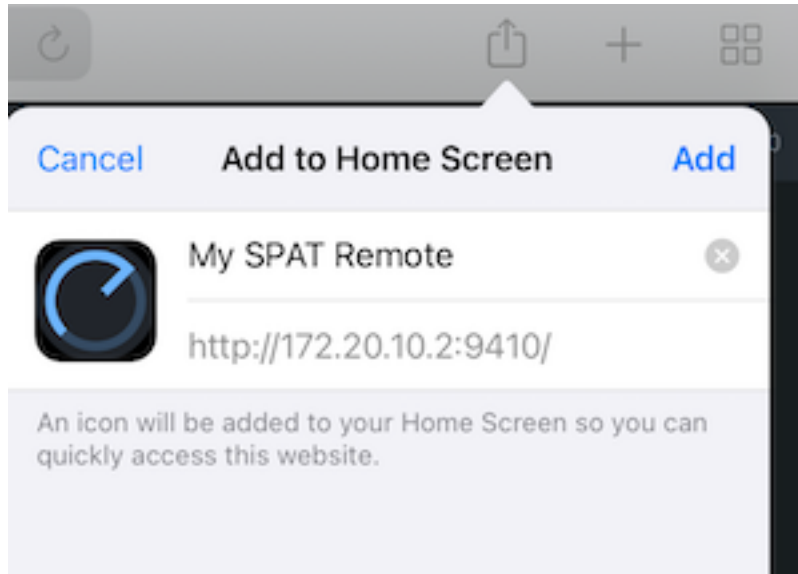


Figure 51.12: Adding Page to iPad Home Screen 2

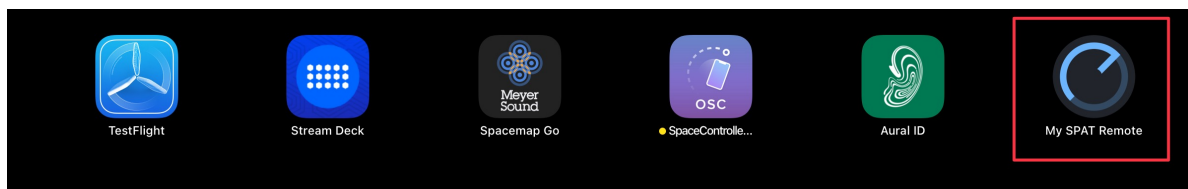


Figure 51.13: Adding Page to iPad Home Screen 3

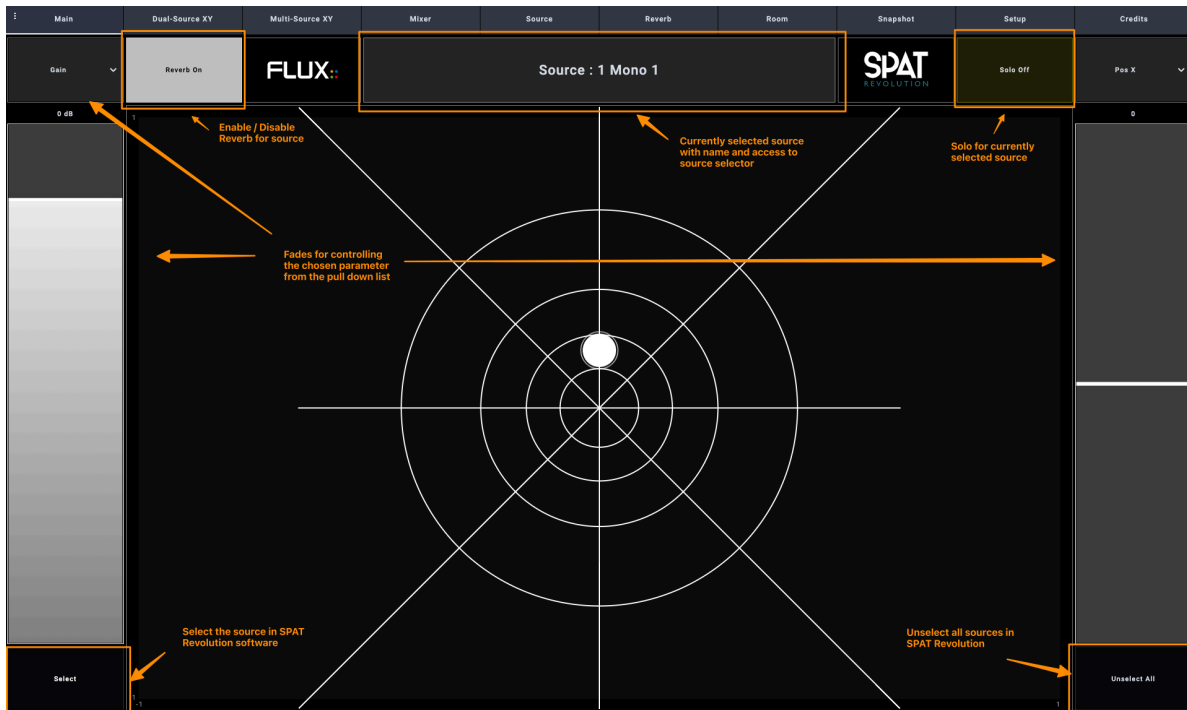


Figure 51.14: SPAT Revolution Main Page

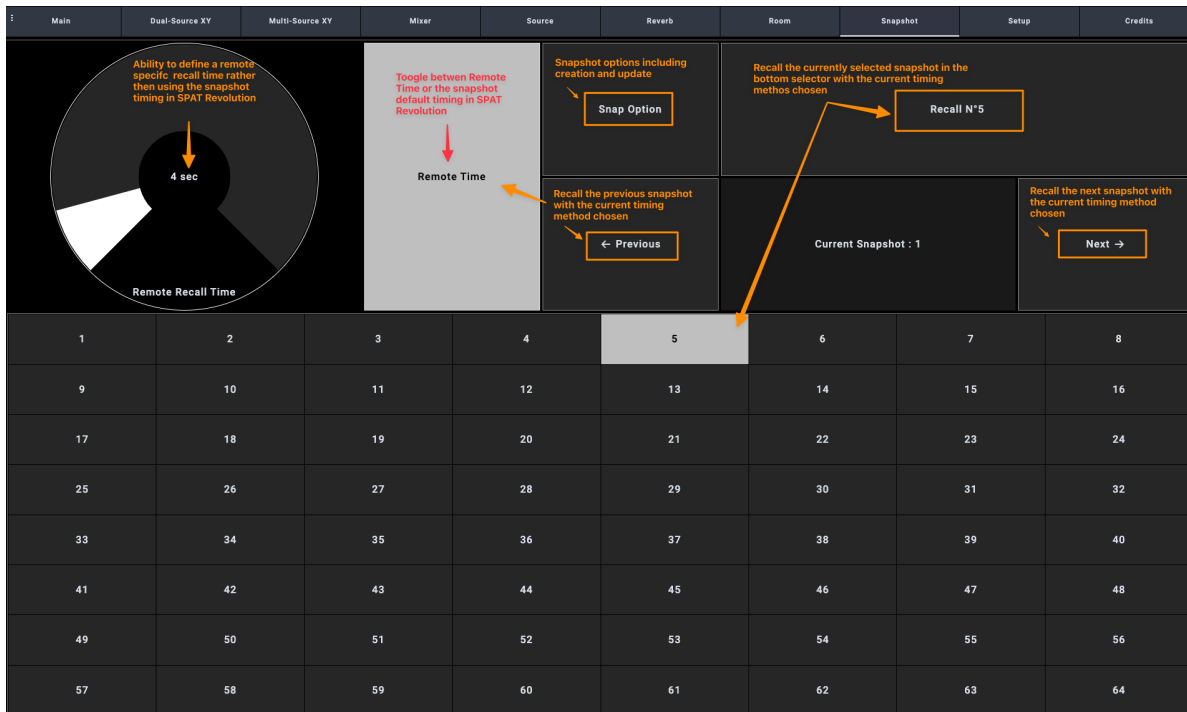
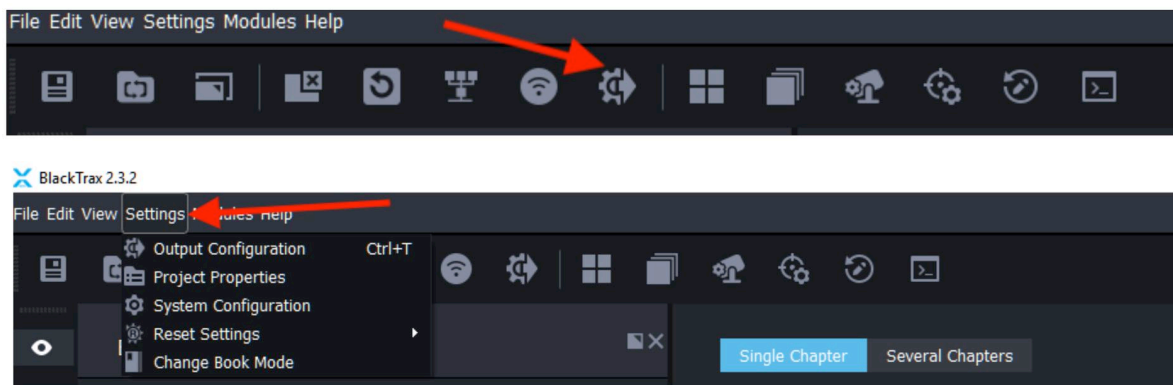


Figure 51.15: SPAT Revolution Main Page

52 BlackTrax integration

Starting in SPAT version 1.1, a new UI implementation is available to integrate the BlackTrax RTTrPM tracking protocol in SPAT. This provides a way to map BlackTrax tracking beacons with SPAT source objects and SPAT room listeners. 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 must create with the + sign 1 (or multiple if running a main and a backup SPAT computer engine) output that will send 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 your output to the desired name (ex: SPAT Main). The output type 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 the BlackTrax server for RTTrPM is 10.133.3.9 255.255.255.0. They have multiple NICs for various uses, such as lighting and the actual BlackTrax network itself. 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 attempting to change IP.

In the below example, the addresses of the Blacktrax NIC was changed to 192.168.1.200 Communication should be set to Unicast. Please refrain from using Broadcast unless there are specific circumstances. Currently, Multicast is not implemented with SPAT software. The port number should be left to the default port # 24002 as it is the BlackTrax system's 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 options. 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 want to send tracking data. In the Active Output window, select your SPAT Main output. On the bottom right, select the Beacon you want data sent to 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

Output Options

Trackable Settings Output Modules

Smoothing (-4 to 2 inclusive):
Smooth

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

Enable Selected Disable Selected

When done 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 sending data by returning 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.

Blacktrax RTTrPM

Enable

Max Beacon

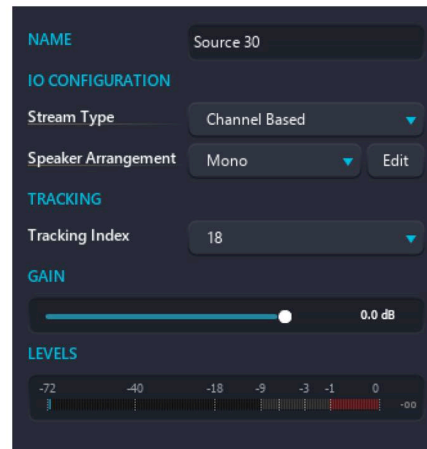
Interface

Port

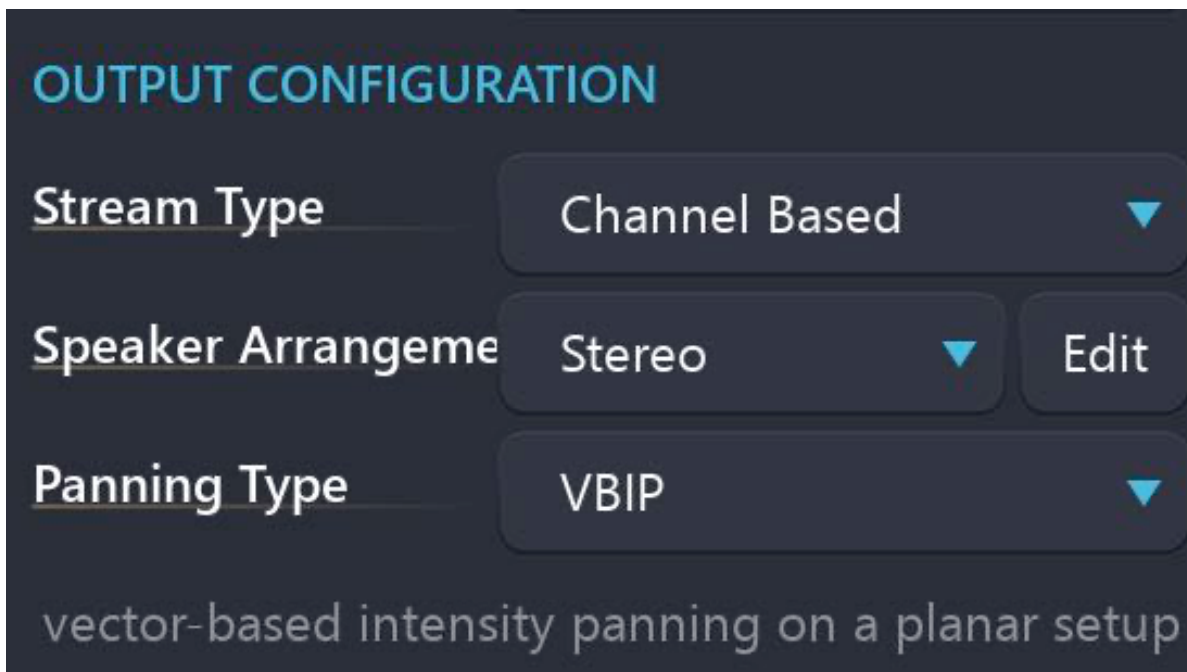
You can, from there, Enable BlackTrax RTTrPM tracking and select the amount of Max Beacon your system may need (1, 2, 4, 8, 16, 24, 32, or more). Please select the network interface you are using on your computer for the BlackTrax integration (Or leave it to Automatic) and then the port #, which is already set at the default port # used by BlackTrax RTTrPM output. You are set. These new preferences will be stored with all your preferences and remain after closing and restarting SPAT Revolution software.

The first option is assigning a Beacon ID to an audio source object. Selecting a source will give you the source menu options. In the option are now Tracking options and a pull-down menu to choose the Tracking Index (the beacon ID number you want for this audio source).

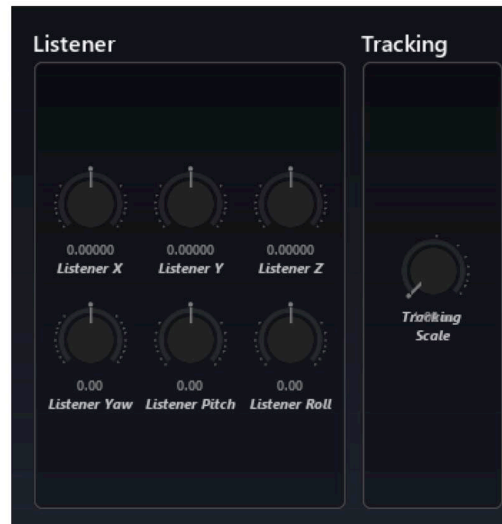
The list will be based on your Max Beacon preference. It is possible to have the same Index for multiple sources, allowing multiple sources to track a single beacon.



Your second option is to assign a Beacon ID to an actual room to track the listener's position. Selecting a room will give you the room menu options. In the option are now Tracking options with a pull-down menu 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 needed.



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



A SPAT Revolution - Credits

A.1 Version 22.09

A.2 User Guide and Documentation

Cristian Vogel, Hugo Larin, Jean-Loup Pecquais and Nicolas Erard

Additional contributions: Gaël Martinet, Vincent Carlier, Florie-Anne Lafaye, Thibaut Carpentier, Anders Tveit and the IRCAM team.

A.3 Software development

A.3.1 Project Manager and Designer:

Gaël Martinet

Additional contributions:

Hugo Larin

A.3.2 Application Development

Gaël Martinet Florie-Anne Lafaye Alexis Gentil Nicolas Erard Siegfried Hand Antoine Lorence

A.3.3 Flux:: DSP Design and Development

Gaël Martinet Maxence Grandidier Lorcan Mc Donagh

A.3.4 IRCAM “Spat~” Design of digital signal processing algorithms and implementation in Max

Jean-Marc Jot (Espaces Nouveaux / Ircam).

A.3.5 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

A.3.6 IRCAM “Spat~” C++ dsp code

Thibaut Carpentier Remy Muller

A.3.7 Flux:: Framework development

Gaël Martinet Florie-Anne Lafaye Alexis Gentil Bastien Prevosto Anthony Belard Siegfried Hand Antoine Lorence

Spatialisateur and Spat~ are trademarks of Ircam and Espaces Nouveaux.

A.4 Graphic design

Nicolas Philippot Gaël Martinet

Additional contributions:

Vincent Carlier Jean-Loup Pecquais Nicolas Erard Jean Cruypenynck Pablo Arias Samuel Tracol

A.4.1 Flux:: Framework graphic engine

Emmanuel Julien (GS lib) Gaël Martinet

A.5 Translation

German: Nils Hahmann Spanish: Ralph Killhertz French: Jean-Loup Pecquais & Nicolas Erard Korean: Gong Eunju & Kim Joo-ha Italian: Gianni Tamanini

A.5.1 And

thanks to all fantastic testers...

A.5.2 FLUX:: Special Thanks to

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

A.6 FLUX::

www.flux.audio COPYRIGHT (C) 2024, HARMAN INTERNATIONAL. ALL RIGHTS RESERVED. FLUX:: is a trademark of HARMAN International.

A.7 Additional libs

- GS lib Emmanuel Julien, Gael Martinet (Copyright 2013 Emmanuel Julien)
- ThorVG, Copyright (c) 2020 - 2023 notice for the ThorVG Project (see AUTHORS)
- r8brain free - Copyright (c) 2013-2023 Aleksey Vaneev
- LibJpeg - Copyright (c) 1991-2016, Thomas G. Lane, Guido Vollbeding
- libpng :
 - Copyright (c) 1995-2023 The PNG Reference Library Authors.
 - Copyright (c) 2018-2023 Cosmin Truta.
 - Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson.
 - Copyright (c) 1996-1997 Andreas Dilger.
 - Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.
- Freetype 2 - Copyright (c) 2006-2023 by David Turner, Robert Wilhelm, and Werner Lemberg.
- Zlib - Copyright (c) 1995-2022 Jean-loup Gailly and Mark Adler
- bzip2 - Copyright (c) 1996-2010 Julian Seward jseward@bzip.org
- Boost - BSL-1.0
- ni-media - Copyright (c) 2022 Native Instruments
- pfft:
 - Copyright (c) 2004 the University Corporation for Atmospheric Research (“UCAR”)
 - Copyright (c) 2013 Julien Pommier (pommier@modartt.com)
 - Copyright (c) 2019 Hayati Ayguen (h_ayguen@web.de)
 - Copyright (c) 2020 Dario Mambro (dario.mambro@gmail.com)
- pybind11 - Copyright (c) 2016 Wenzel Jakob wenzel.jakob@epfl.ch

- rtaudio - Copyright (c) 2001-2021 Gary P. Scavone
- rtmidi - Copyright (c) 2003-2021 Gary P. Scavone
- concurrentqueue - Copyright (c) 2013-2016, Cameron Desrochers.
- readerwriterqueue - Copyright (c) 2013-2021, Cameron Desrochers.
- assimp - Copyright (c) 2006-2021, assimp team
- enet - Copyright (c) 2002-2020 Lee Salzman
- fmt - Copyright (c) 2012 - present, Victor Zverovich and {fmt} contributors
- functions (Erik Rigtorp) - Copyright (c) 2015 Erik Rigtorp erik@rigtorp.se
- inplace_function - BSL-1.0
- gtest - Copyright 2008, Google Inc.
- jsoncpp - Copyright (c) 2007-2010 Baptiste Lepilleur and The JsonCpp Authors
- libcurl - Copyright (c) 1996 - 2023, Daniel Stenberg, daniel@haxx.se, and many contributors
- magic_enum - Copyright (c) 2019 - 2023 Daniil Goncharov
- nanobench - Copyright (c) 2019-2023 Martin Leitner-Ankerl
- nlohmann_json - Copyright (c) 2013-2022 Niels Lohmann
- openddl - Copyright (c) 1999-2022 Eric Lengyel
- openvr - Copyright (c) 2015, Valve Corporation
- poly2tri - Copyright (c) 2009-2018, Poly2Tri Contributors
- pugixml - Copyright (c) 2006-2022 Arseny Kapoulkine
- rapidjson - Copyright (c) 2015 THL A29 Limited, a Tencent company, and Milo Yip.
- shared_recursive_mutex - Copyright (c) 2019 konanM
- sentry-native - Copyright (c) 2019 Sentry (<https://sentry.io>) and individual contributors.
- sqlite3 - Public Domain License
- sqlite_orm - Copyright (c) 2012-2023 Eugene Zakharov and others
- stb - Copyright (c) 2017 Sean Barrett
- utfcpp - Copyright 2006 Nemanja Trifunovic
- unqlite - Copyright (c) 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019 Symisc Systems, S.U.A.R.L [M.I.A.G + Mrad Chems Eddine chm@symisc.net].

A.7.1 Additional Python modules

- sourcedefender - Copyright (c) 2018-2024 SOURCEdefender. All rights reserved.
- RTTrPM SDK (Blacktraxx) Copyright (c) 2019 CAST Group of Companies Inc
- argh (Andrey Mikhaylenko) - Copyright © 2010—2023 Andrey Mikhaylenko and contributors
- chardet (Mark Pilgrim, Daniel Blanchard)
- charset_normalizer - Copyright (c) 2019 TAHRI Ahmed R.

- docopt (Vladimir Keleshev) - Copyright (c) 2012 Vladimir Keleshev, vladimir@keleshev.com
- dotenv - Copyright (c) 2014, Saurabh Kumar
- ecdsa (Peter Pearson) - Copyright (c) 2010 Brian Warner
- environs (Steven Loria) - Copyright 2022 Steven Loria
- ftputil (Stefan Schwarzer) - Stefan Schwarzer sschwarzer@sschwarzer.net and others contributors
- numpy - Copyright (c) 2005-2023, NumPy Developers.
- pendulum - Copyright (c) 2015 Sébastien Eustace
- psutil (Giampaolo Rodola) - Copyright (c) 2009, Jay Loden, Dave Daeschler, Giampaolo Rodola
- watchdog:
- Copyright 2010 Malthe Borch mborch@gmail.com
- Copyright 2011 Yesudeep Mangalapilly yesudeep@gmail.com Copyright 2012 Google, Inc & contributors.
- tgcrypto - Copyright (c) 2017-2020 Dan <https://github.com/delivrance>
- idna - Copyright (c) 2013-2022, Kim Davies and contributors. All rights reserved.
- marshmallow - Copyright 2021 Steven Loria and contributors
- packaging - Copyright (c) Donald Stufft and individual contributors.
- paramiko (Jeff Forcier) - Copyright (c) 2003-2008 Robey Pointer robeypointer@gmail.com
- pathtools Copyright (c) 2010 by Yesudeep Mangalapilly
- ptyvsd (Microsoft) - Copyright (c) Microsoft Corporation. All rights reserved.
- pyga (Arun K.R.)
- pylightxl - Copyright (c) 2019 Viktor Kis
- pyparsing - Copyright (c) 2003-2019 Paul T. McGuire
- pytzdata - Copyright (c) 2015 Sébastien Eustace
- winshell (Tim Golden) - Copyright (c) 2012 Tim Golden
- six - Copyright (c) 2010-2020 Benjamin Peterson
- Request (Kenneth Reitz)

- sentry_sdk - Copyright (c) 2018 Functional Software, Inc. dba Sentry
- urllib3 - Copyright (c) 2008-2020 Andrey Petrov and contributors
- certifi (Kenneth Reitz)
- anytree (c0fec0de)
- boltons - Copyright (c) 2013, Mahmoud Hashemi
- msgpack (Inada Naoki) - Copyright (c) 2008-2011 INADA Naoki songofacandy@gmail.com
- dateutil - Copyright 2017- Paul Ganssle paul@ganssle.io and dateutil contributors
- PyYAML (Kirill Simonov)

A.8 Open source libraries

A.8.1 RTTrPM SDK (Blacktraxx)

Copyright (c) 2019 CAST Group of Companies Inc

A.8.2 ThorVG (replaces crossed-out libraries above) - MIT

<https://github.com/thorvg/thorvg/blob/main/LICENSE> Copyright (c) 2020 - 2024 notice for the ThorVG Project (see AUTHORS) Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.3 r8brain free - MIT

<https://github.com/avaneev/r8brain-free-src/blob/master/LICENSE>

A.8.4 ooura fft from lmdsp - MIT (or MIT-like)

<https://www.kurims.kyoto-u.ac.jp/~ooura/fft.html> Copyright Takuya OOURA, 1996-2001

A.8.5 LibJpeg - Custom (BSD-like)

<https://jpegclub.org/reference/> Copyright (C) 1991-2016, Thomas G. Lane, Guido Vollbeding

A.8.6 libpng - PNG Reference Library License version 2

<https://sourceforge.net/p/libpng/code/ci/master/tree/LICENSE> Copyright (c) 1995-2023 The PNG Reference Library Authors. Copyright (c) 2018-2023 Cosmin Truta. - Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson. Copyright (c) 1996-1997 Andreas Dilger. Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

A.8.7 Freetype 2 - FreeType License

<https://github.com/freetype/freetype/blob/master/LICENSE.TXT> Copyright (C) 2006-2023 by David Turner, Robert Wilhelm, and Werner Lemberg

A.8.8 Zlib - zlib

https://zlib.net/zlib_license.html Copyright (C) 1995-2022 Jean-loup Gailly and Mark Adler

A.8.9 bzip2 - Modified zlib

Copyright (C) 1996-2010 Julian Seward jseward@bzip.org

A.8.10 Boost - BSL-1.0

https://github.com/boostorg/boost/blob/master/LICENSE_1_0.txt

A.8.11 mimalloc - MIT

<https://github.com/microsoft/mimalloc/blob/master/LICENSE>

A.8.12 ni-media - MIT

<https://github.com/NativeInstruments/ni-media/blob/master/LICENSE>

A.8.13 pfft

<https://github.com/marton78/pfft/blob/master/LICENSE.txt>

A.8.14 pybind11

<https://github.com/pybind/pybind11/blob/master/LICENSE>

A.8.15 rtaudio

<https://github.com/thestk/rtaudio/blob/master/LICENSE>

A.8.16 rtmidi

<https://github.com/thestk/rtmidi/blob/master/LICENSE>

A.8.17 concurrentqueue - Simplified BSD

<https://github.com/cameron314/concurrentqueue/blob/master/LICENSE.md> copyright (c) 2013-2016, Cameron Desrochers. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”

A.8.18 readerwriterqueue - Simplified BSD

<https://github.com/cameron314/readerwriterqueue/blob/master/LICENSE.md> Copyright (c) 2013-2021, Cameron Desrochers

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”

A.8.19 assimp

<https://github.com/assimp/assimp/blob/master/LICENSE> Copyright (c) 2006-2021, assimp team All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the assimp team, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of the assimp team.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”

A.8.20 enet - MIT

<https://github.com/lsalzman/enet/blob/master/LICENSE>

A.8.21 fmt

<https://github.com/fmtlib/fmt/blob/master/LICENSE.rst> Copyright notice Copyright (c) 2012 - present, Victor Zverovich and {fmt} contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute,

sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.22 functions (Erik Rigtorp) - MIT

<https://github.com/rigtorp/Function/blob/master/LICENSE>

A.8.23 inplace_function - BSL-1.0

A.8.24 gtest - ” BSD 3-Clause “New” or “Revised” ”

<https://github.com/google/googletest/blob/main/LICENSE>

A.8.25 jsoncpp - MIT

<https://github.com/open-source-parsers/jsoncpp/blob/master/LICENSE> Copyright (c) 2007-2010 Baptiste Lepilleur and The JsonCpp Authors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF

CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.26 libcurl - MIT like

<https://curl.se/docs/copyright.html> Copyright (c) 1996 - 2024, Daniel Stenberg, daniel@haxx.se, and many contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.”

A.8.27 magic_enum - MIT

https://github.com/Neargye/magic_enum/blob/master/LICENSE Copyright (c) 2019 - 2024 Daniil Goncharov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE

FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.28 nanobench - MIT

<https://github.com/martinus/nanobench/blob/master/LICENSE>) Copyright (c) 2019-2023 Martin Leitner-Ankerl

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ““Software”“), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ““AS IS”“, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.29 nlohmann_json - MIT

<https://github.com/nlohmann/json/blob/develop/LICENSE.MIT> Copyright (c) 2013-2022 Niels Lohmann

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ““Software”“), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ““AS IS”“, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE

FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.30 openddl - MIT

<https://github.com/EricLengyel/OpenDDL/blob/main/LICENSE> Copyright (c) 1999-2022 Eric Lengyel

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.31 openvr - ” BSD 3-Clause “New” or “Revised” ”

<https://github.com/ValveSoftware/openvr/blob/master/LICENSE> Copyright (c) 2015, Valve Corporation All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES - LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”

A.8.32 poly2tri - ” BSD 3-Clause “New” or “Revised”“ ”

<https://github.com/jhasse/poly2tri/blob/master/LICENSE> Copyright (c) 2009-2018, Poly2Tri Contributors All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Poly2Tri nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”

A.8.33 pugixml - MIT

<https://github.com/zeux/pugixml/blob/master/LICENSE.md> Copyright (c) 2006-2024 Arseny Kapoulkine

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.34 rapidjson - MIT

<https://github.com/Tencent/rapidjson/blob/master/license.txt> - “Terms of the MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.35 shared_recursive_mutex - MIT

https://github.com/KonanM/shared_recursive_mutex/blob/master/LICENSE Copyright (c) 2019 konanM

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.36 sentry-native - MIT

<https://github.com/getsentry/sentry-native/blob/master/LICENSE> Copyright (c) 2019 Sentry (<https://sentry.io>) and individual contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.37 sqlite3 - Public Domain

<https://www.sqlite.org/copyright.html>

A.8.38 sqlite_orm - MIT

https://github.com/fnc12/sqlite_orm/blob/master/COMM-LICENSE Copyright (c) 2012-2023 Eugene Zakharov and others

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.39 stb - MIT

<https://github.com/nothings/stb/blob/master/LICENSE> Copyright (c) 2017 Sean Barrett
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.40 utfcpp - BSL-1.0

<https://github.com/nemtrif/utfcpp/blob/master/LICENSE> “Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the “Software”) to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.41 unqlite - BSD 2-clause

<https://github.com/symisc/unqlite/blob/master/LICENSE> ” * Copyright (C) 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019 Symisc Systems, S.U.A.R.L [M.I.A.G Mrad Chems Eddine chm@symisc.net]. * All rights reserved. Redistribution and use in source and binary forms, with or without * modification, are permitted provided that the following conditions * are met: * 1. Redistributions of source code must retain the above copyright * notice, this list of conditions and the following disclaimer. * 2. Redistributions in binary form must reproduce the above copyright * notice, this list of conditions and the following disclaimer in the * documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY SYMISC SYSTEMS “AS IS” AND ANY EXPRESS * OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED * WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR * NON-INFRINGEMENT, ARE DISCLAIMED. IN NO EVENT SHALL SYMISC SYSTEMS * BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF * SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE,

DATA, OR PROFITS - OR * BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, * WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE * OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN * IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.” 3rd Party Open Source Python Modules - - -

A.8.42 argh (Andrey Mikhaylenko) - LGPLv3

<https://pypi.org/project/argh/>

A.8.43 chardet (Mark Pilgrim, Daniel Blanchard) - LGPL v2.1

<https://github.com/chardet/chardet/blob/main/LICENSE>

A.8.44 charset_normalizer - MIT

https://github.com/Ousret/charset_normalizer/blob/master/LICENSE Copyright (c) 2019 TAHRI Ahmed R.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.45 docopt (Vladimir Keleshev) - MIT

<https://github.com/docopt/docopt/blob/master/LICENSE-MIT> Copyright (c) 2012 Vladimir Keleshev, vladimir@keleshev.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.8.46 dotenv - “BSD 3-Clause” “New” or “Revised”

<https://github.com/theskumar/python-dotenv/blob/main/LICENSE> Copyright (c) 2014, Saurabh Kumar (python-dotenv), 2013, Ted Tiekens (django-dotenv-rw), 2013, Jacob Kaplan-Moss (django-dotenv)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of django-dotenv nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT

NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”

A.9 Open source python libraries

A.9.1 ecdsa (Peter Pearson) -

<https://github.com/tlsfuzzer/python-ecdsa/blob/master/LICENSE> ““python-ecdsa” Copyright (c) 2010 Brian Warner

Portions written in 2005 by Peter Pearson and placed in the public domain.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ““Software”“), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ““AS IS”“, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.9.2 environs (Steven Loria) - MIT

<https://github.com/sloria/environs/blob/master/LICENSE> “Copyright Steven Loria

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.9.3 ffmpeg-python - Apache 2

<https://github.com/kkroening/ffmpeg-python/blob/master/LICENSE> ” Copyright 2017 Karl Kroening

Licensed under the Apache License, Version 2.0 (the “License”) - you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.”

A.9.4 ftputil (Stefan Schwarzer) - BSD

<https://pypi.org/project/ftputil/> Copyright (c) 2004-2024 Stefan Schwarzer sschwarzer@sschwarzer.net and others contributors

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE”

A.9.5 numpy - “BSD 3-Clause”“New”” or ““Revised”“ ”

<https://github.com/numpy/numpy/blob/main/LICENSE.txt> Copyright (c) 2005-2024, NumPy Developers. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the NumPy Developers nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT

NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”

A.9.6 pendulum - MIT

<https://github.com/sdispater/pendulum/blob/master/LICENSE> Copyright (c) 2015 Sébastien Eustace

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.9.7 psutil (Giampaolo Rodola) - “BSD 3-Clause” “New” or “Revised”

<https://github.com/giampaolo/psutil/blob/master/LICENSE> Copyright (c) 2009, Jay Loden, Dave Daeschler, Giampaolo Rodola All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the psutil authors nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES - LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”

A.9.8 watchdog - Apache 2

<https://github.com/gorakhargosh/watchdog/blob/master/LICENSE> “Copyright 2010 Malthe Borch mborch@gmail.com Copyright 2011 Yesudeep Mangalapilly yesudeep@gmail.com Copyright 2012 Google, Inc & contributors.

Licensed under the Apache License, Version 2.0 (the “License”) - you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.”

A.9.9 tgcrypto - LGPL v3

<https://github.com/pyrogram/tgcrypto/blob/master/COPYING.lesser> Copyright (C) 2017-2020 Dan <https://github.com/delivrance>

A.9.10 idna - BSD 3

<https://github.com/kjd/idna/blob/master/LICENSE.md> Copyright (c) 2013-2023, Kim Davies and contributors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”

A.9.11 marshmallow - MIT

<https://github.com/marshmallow-code/marshmallow/blob/dev/LICENSE> “Copyright Steven Loria and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.9.12 packaging - BSD

<https://github.com/pypa/packaging/blob/main/LICENSE> Copyright (c) Donald Stufft and individual contributors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE”

A.9.13 paramiko (Jeff Forcier) - LGPL 2.1

<https://github.com/paramiko/paramiko/blob/main/LICENSE> Copyright (C) 2003-2008 Robey Pointer robeypointer@gmail.com <https://raw.githubusercontent.com/paramiko/paramiko/main/LICENSE>

A.9.14 pathtools - MIT

<https://github.com/gorakhargosh/pathtools/blob/master/LICENSE> Copyright (C) 2010 by Yesudeep Mangalapilly yesudeep@gmail.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.9.15 ptvsd (Microsoft) - MIT

<https://github.com/microsoft/PTVS/blob/main/LICENSE> Copyright (c) Microsoft Corporation. All rights reserved.

Licensed under the Apache License, Version 2.0 (the “License”) - you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.”

A.9.16 pyga (Arun K.R.) - BSD

<https://pypi.org/project/pyga/> Copyright (c) Arun K.R. and individual contributors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE”

A.9.17 pylightxl - MIT

<https://github.com/PydPiper/pylightxl/blob/master/LICENSE> Copyright (c) 2019 Viktor Kis

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.9.18 pyparsing - MIT

<https://github.com/pyparsing/pyparsing/blob/master/LICENSE> Copyright (c) 2003-2019 Paul T. McGuire Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.9.19 pytzdata - MIT

<https://github.com/sdispater/pytzdata/blob/master/LICENSE> Copyright (c) 2015 Sébastien Eustace

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.9.20 winshell (Tim Golden) - MIT

<https://github.com/tjguk/winshell/blob/master/LICENSE.txt> Copyright (c) 2012 Tim Golden

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.9.21 six (Copyright (c) 2010-2020 Benjamin Peterson - MIT

<https://github.com/benjaminp/six/blob/master/LICENSE> Copyright (c) 2010-2020 Benjamin Peterson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.9.22 Request (Kenneth Reitz) - Apache 2

<https://github.com/psf/requests/blob/main/LICENSE> - modified <https://raw.githubusercontent.com/psf/requests/main/LICENSE>

A.9.23 sentry_sdk (Copyright (c) 2018 Sentry) - MIT

<https://github.com/getsentry/sentry-python/blob/master/LICENSE> Copyright (c) 2018 Functional Software, Inc. dba Sentry

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.9.24 urllib3 - MIT

<https://github.com/urllib3/urllib3/blob/main/LICENSE.txt> Copyright (c) 2008-2020 Andrey Petrov and contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

A.9.25 certifi (Kenneth Reitz) - MPL-2.0

<https://github.com/certifi/python-certifi/blob/master/LICENSE> “This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.”

A.9.26 anytree (c0fec0de) - Apache 2

<https://github.com/c0fec0de/anytree/blob/main/LICENSE>

A.9.27 boltons - BSD

<https://github.com/mahmoud/boltons/blob/master/LICENSE> Copyright (c) 2013, Mahmoud Hashemi

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * The names of the contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”

A.9.28 msgpack (Inada Naoki) - Apache 2

<https://github.com/msgpack/msgpack-python/blob/main/COPYING> Copyright (C) 2008-2011 INADA Naoki songofacandy@gmail.com

Licensed under the Apache License, Version 2.0 (the “License”) - you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.”

A.9.29 dateutil (Gustavo Niemeyer, Paul Ganssle) - Apache 2

<https://github.com/dateutil/dateutil/blob/master/LICENSE> “Copyright 2017- Paul Ganssle paul@ganssle.io Copyright 2017- dateutil contributors (see AUTHORS file)

Licensed under the Apache License, Version 2.0 (the “License”) - you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.”

A.9.30 PyYAML (Kirill Simonov) - MIT

<https://pypi.org/project/PyYAML/> Copyright (c) Kirill Simonov.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

B SPAT Revolution specifications

B.1 Availability

SPAT Revolution is a stand-alone application for macOS and Windows. SPAT Send/Return/Room plugins are available in AU/VST/AAX Native. SPAT Send plugin is available in AAX VENUE.

See [plugins specifications](#) for more details.

B.2 Processing

SPAT Revolution stand-alone software: - unlimited number of channels for Inputs and Outputs (**Ultimate** license only, hardware and audio interface dependent). - 32/64-bits internal floating-point processing. - Sampling rates up to 384kHz, block size starting at 16 samples.

B.3 OS Compatibility

- Windows 10 - 64 bits.
- macOS - All versions from 10.15 including macOS Big Sur and Monterey compliant. *

i Note

*See hardware details for more information on compatibility.

B.4 Hardware Recommendations and requirements

B.4.1 Single Computer (Creative Station with DAW and SPAT Revolution)

Single computer systems can provide an excellent option for portability but come with the importance of having a performing computer. As *SPAT Revolution* comes with a 3D graphic engine where audio objects are manipulated, GPU resources are required.

Apple based recommended hardware:

- Apple Silicon / ARM M2
 - Mac Studio M2 Max / M2 Ultra off the shelf models
 - Mac Mini M2 Chip
 - MacBook Pro (13/14/16 inches) M2 Chip
- Apple Intel
 - MacPro, 3.5GHz 8-core Intel Xeon W processor, Radeon Pro W5500X, 32GB RAM
 - iMac Pro, 3.2GHz 8-core Intel Xeon W Turbo Boost, Radeon Pro Vega 56 8 GB, 32GB RAM
 - MacBook Pro with dedicated GPU video graphic card

Generic hardware specification:

- Processing: Intel Core i9-9900K, i7-9700K or equivalent. Intel 8th generation or greater processor. (Minimum 6 cores - 8 preferred), 8 MB Cache.
- System Memory: 16 minimum GB DDR4 (32 preferred when integrating with DAW and *SPAT Revolution* on the same machine with local audio path - LAP)
- Chipset: High-quality / Professional main-board
- Graphic: GeForce GTX 1060 - 4GB GDDR5 Graphic Memory or greater. *Graphic card fully supporting OpenGL 2.0 is required. USB displays are not supported.*
- Audio Interfaces:
 - macOS: Core Audio compatible interface or virtual sound card
 - Windows: ASIO compatible interface or virtual sound card.

B.4.2 Hardware for Live Productions (Dedicated SPAT Revolution Computer)

Recommended audio interfaces:

- RME Digiface AVB or Dante, USB 3 Audio interface
- RME Madi HDSPe (PCIe), MADI FX or MADIface XT.

Apple based recommended hardware:

- Apple Silicon / ARM M2
 - Mac Studio M2 Max (10-Core CPU, 24-Core GPU, 32 GB RAM)
 - Mac Studio M2 Ultra (20-Core CPU, 48-Core GPU, 64 GB RAM)
- Apple Intel
 - MacPro, 3.5GHz 8-core Intel Xeon W processor, Radeon Pro W5500X, 32GB RAM

i Note

Rackmounted Apple Mac Studio is possible using Sonnet's xMac Studio or RackMac Studio 3U rackmount systems. xMac Studio can hold one Mac Studio and provide PCIe card expansion module while the RackMac Studio can hold 2 units such as in redundant configuration. MacPro can be installed in a Sonnet's RackMac™

Generic recommended hardware specification:

- Intel® Xeon® W family W-2200 or W-3200 Series CPU (preferred), E-22xx, or equivalent. Higher core speed. Minimum 8 Cores, 12 MB Cache
- Chipset: Professional workstation chipset C422, C621, or equivalent.
- System Memory: 32 – 64 GB. A system with ECC support is preferred.
- Graphic: Professional NVIDIA Quadro P4000 or equivalent - 8GB GDDR5 Graphic Memory (Graphic card with full support for OpenGL 2.0), with the latest graphic drivers.
- Network: Dual network interface (NIC) - Intel I210-T1 or equivalent
- OS drive (Operating System): NVMe Internal SSD
- Audio Projects Hard Drive: no specifics. For redundant systems, only certified RAID systems should be used.
- Real-Time optimized OS with minimum services.

B.4.3 Software License Requirements

To use the software, an iLok.com user account is required (the iLok USB Smart Key is not required, authorization can be made to hardware machines).

B.5 Certified configurations (SPAT Revolution v.22.02.50151)

These configurations are tested and validated on a computer dedicated to *SPAT Revolution*. All other applications have been closed, and computer has been optimized as indicated below.

Computer optimization: - Disable Wi-Fi and Bluetooth if not absolutely required. - Disable Airdrop - Disable Spotlight - Disable display sleep, disk spin-down, and system sleep - Disable screen saver - Disable Time Machine - Disable software update - Disable dashboard (Catalina and prior) - Disable photos processing and Cloud services - Log out of iCloud

The below configurations have been **tested at the worst scenario**, with the reverberation at the highest density for all rooms, all sources active with reverberation and all sources in full movement using OSC input. WFS and Binaural PanType are used as reference as they are the most processing heavy techniques.

⚠ Warning

Processing can be different according to the used HRTF. Kemar HRTF were used for these tests.

In the case of the *Dante Virtual Soundcard*, the network latency, reported into SPAT, have been set to 10ms.

B.5.1 48000 kHz

mac Mini M1 | Basic configuration

Number of sources	Number of speakers	Room and PanType	Buffer with RME Digiface Dante / AVB	Buffer with Dante Virtual Soundcard
64	32	1 WFS room	256 (11.5ms)	128 (25.33ms)
64	64	1 WFS room	512 (22.16ms)	256 (30.66ms)
64	16 + Binaural	2 rooms: 1 WFS + 1 Binaural	1024 (43.5ms)	512 (52ms)

mac Studio M1 Max | Basic configuration

Number of sources	Number of speakers	Room and PanType	Sound card	Buffer size / reported Latency
64	32	1 WFS room	RME Digiface Dante / AVB	64 (3.5ms)
64	64	1 WFS room	RME Digiface Dante / AVB	128 (6.16ms)
64	16 + Binaural	2 rooms: 1 WFS + 1 Binaural	RME Digiface Dante / AVB	128 (6.16ms)

B.5.2 96000 kHz

mac Mini M1 | Basic configuration

Number of sources	Number of speakers	Room and PanType	Sound card	Buffer size / reported Latency
32	16	1 WFS room	RME Digiface Dante / AVB	256 (6.05ms)
32	48	1 WFS room	RME Digiface Dante / AVB	512 (11.44ms)
32	32 + Binaural	2 rooms: 1 WFS + 1 Binaural	RME Digiface Dante / AVB	1024 (22.1ms)
64	16	1 WFS room	RME Digiface Dante / AVB	512 (11.44ms)
64	24	1 WFS room	RME Digiface Dante / AVB	1024 (22.1ms)

mac Studio M1 Max | Basic configuration

Number of sources	Number of speakers	Room and PanType	Sound card	Buffer size / reported Latency
32	16	1 WFS room	RME Digiface Dante / AVB	64 (2.1ms)
32	48	1 WFS room	RME Digiface Dante / AVB	128 (3.42ms)
32	32 + Binaural	2 rooms: 1 WFS + 1 Binaural	RME Digiface Dante / AVB	256 (6.05ms)
64	16	1 WFS room	RME Digiface Dante / AVB	128 (3.42ms)
64	24	1 WFS room	RME Digiface Dante / AVB	256 (6.05ms)

C Troubleshooting

C.1 Sync issues with LAP

When using Local Audio Path (LAP), synchronisation issues can be identified when the synchronisation indicator in *SPAT Revolution* (bottom left corner) turns to red. This can in some cases result in clicks, noises, and or loss of sound.

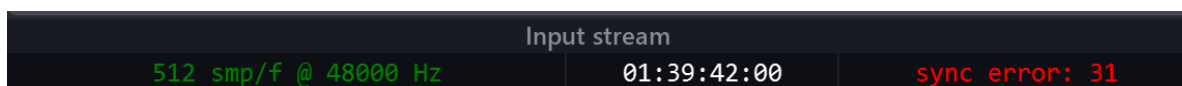


Figure C.1: Sync Error

All 'green' indicators mean the sync is correct.

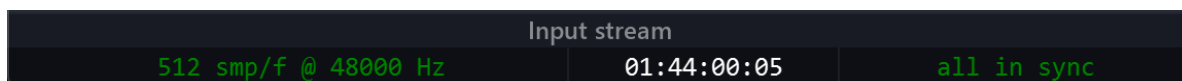


Figure C.2: Sync

Thanks to the DAW templates provided, this should not happen unless the required routing is not in place (session modified, routing unpatched). To work properly, the SPAT Send plug-ins instances must be processed by the DAW before the SPAT 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 'return' tracks, using DAW internal routing (ex: dummy busses in ProTools, direct routing in Reaper).

i Note

CPU performance overloading could generate some sync errors.

i Note

You can reset the sync error by double-clicking on the sync error counter.

If you see red indicators in the bottom left corner section of SPAT, this could be related to having a mismatch in frame size (buffer) or sample rate between the DAW and the *SPAT Revolution*.

Frame size (sometimes called buffer size or block size) should be matched in the host DAW and *SPAT Revolution*. A red message would identify a different frame rate the host DAW. Simply double-clicking on **smp/f** message in error will automatically change you SPAT block size setup to match the incoming audio.

If the audio processing is too demanding for your computer at the current block size and sample rate, you may also experience dropouts and sync problems due to CPU overload.

If you are experiencing lost sync when using Local Audio:

- Increase the block size in the SPAT preferences.
- Save your project and quit *SPAT Revolution*.
- Change the block size in your host DAW to match the new setting.
- Reopen *SPAT Revolution*.

Also, please be sure to carefully read the detailed advice of the various DAW in the **Third party integration** section of this guide and refer to the various provided templates.

C.2 SPAT Send and Return DAW

When dealing with SPAT Send and Return, you do not need to have a hardware device configured in the preference. In this case, *SPAT Revolution* will automatically adapt its sampling rate and synchronize to the incoming software audio I/O.

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. Thanks to the various DAW templates, this is done for the users.

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 plug-in, using DAW internal routing.

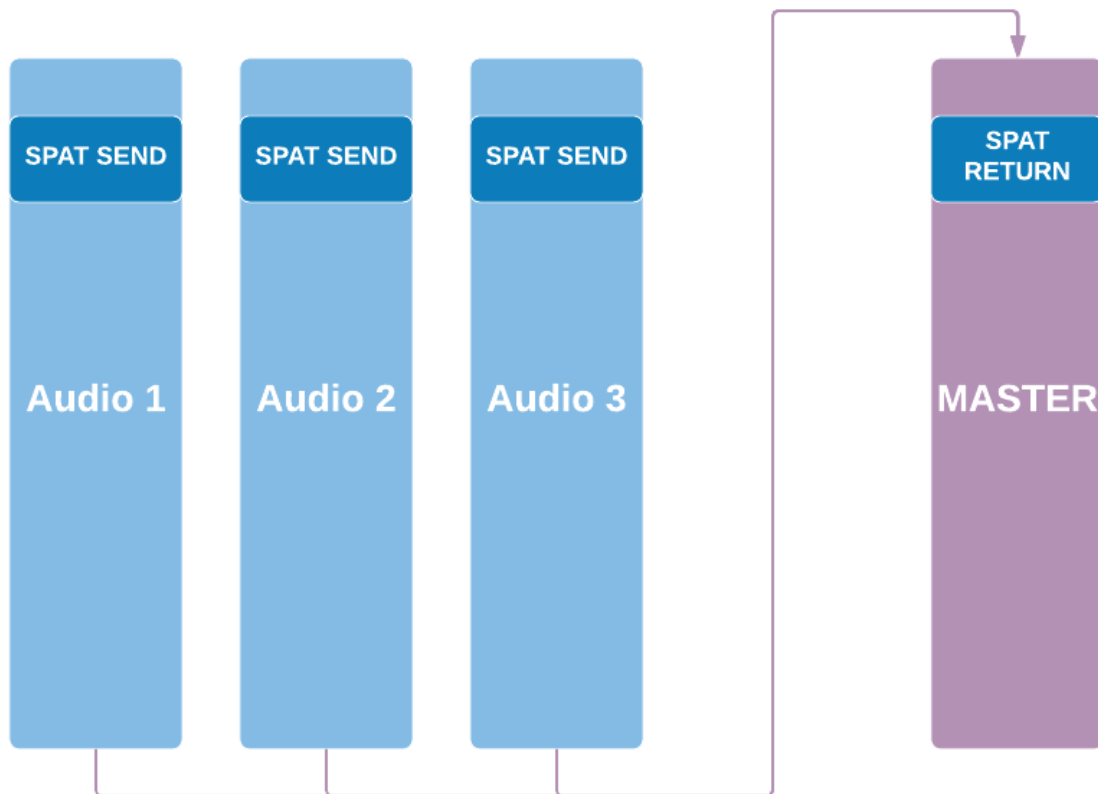
Following are four examples of recommended practice with DAW routing, which should cover the main use cases. If your problems persist even after implementing these suggestions, don't hesitate to drop us a line at FLUX:: support.

i Note

PLEASE NOTE: The mixing of HARDWARE inputs and LOCAL AUDIO PATH may report a sync loss, as *SPAT Revolution* cannot guarantee correct sync in this scenario. Proceed with caution if this is unavoidable. This is not officially supported. __

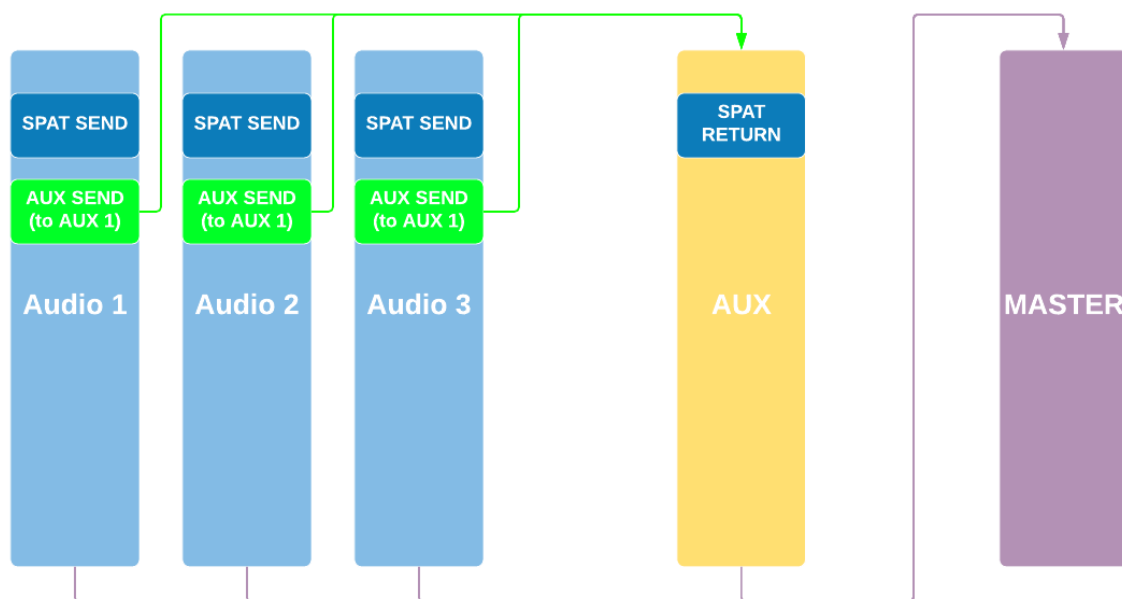
SPAT RETURN plug-in on master track

In simple projects, when having a single return plug-in on a 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). Here is an example using AUX send (see examples below).



i Note

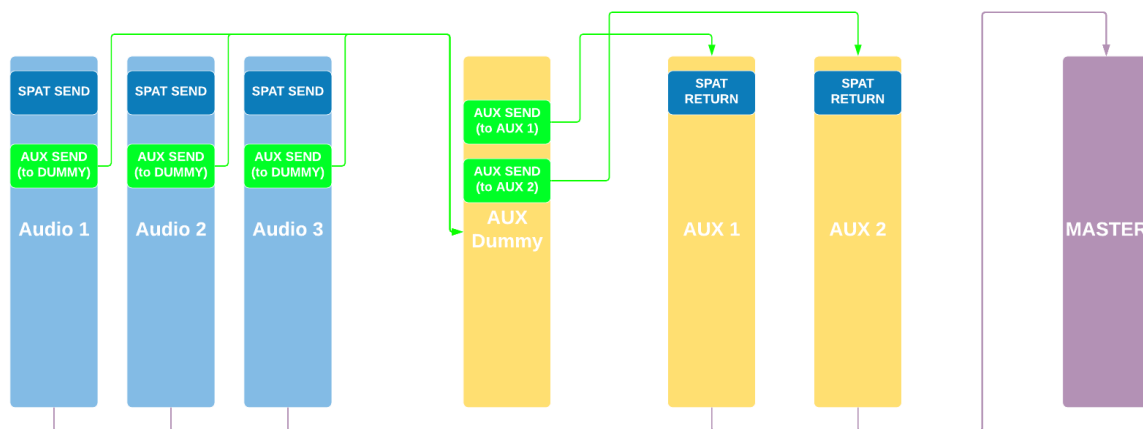
You might achieve the same by routing the track outputs to the bus.

Several RETURN on AUX tracks

When several RETURN tracks are needed (for example several rooms to render from *SPAT Revolution*, and/or several output stream formats), you will have to route each SEND track to each RETURN track, using the same technique.

As it can quickly become complicated as the project grows, in the following example, the use of a 'dummy' track, avoids using several AUX sends on the 'SEND' tracks. It makes routing clearer and easier to implement on larger projects. The 'dummy' AUX track is routed to all the RETURN tracks (using AUX sends or patching the output to a multichannel/multi-format 'dummy' bus).

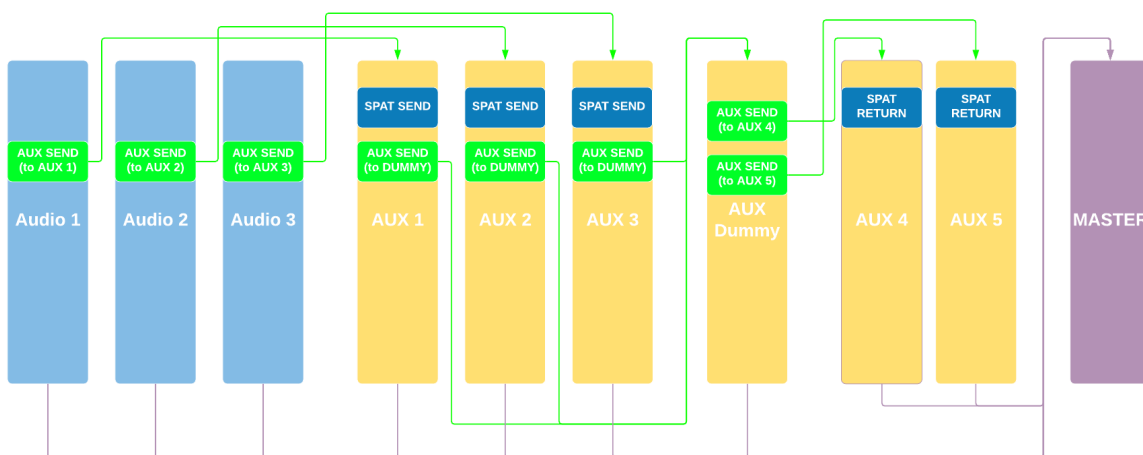
Then, simply route all your SEND tracks to this 'dummy' track by simply patching its output to the multichannel/multi-format 'dummy' bus or via the use of aux send.



Using specific tracks as your SPAT source/object

One of the good practices to deal with the source/object you are sending for external rendering is to use tracks as dedicated objects. (Similar to many object-based mixing workflows proposed by DAW.) This way you can leave the session audio tracks and their channel insertion as they are and simply send your audio track to the SPAT SEND object track. This allows you to send a single audio track or multiple ones (stem) to the SPAT SEND object track.

Doing this can segment your external rendering routing and is highly recommended to prevent audio track delay compensation systems in DAW to come and jeopardize the audio synchronization between the DAW and SPAT. This as well ensures that your audio track automation on levels for example is respected as some DAWs don't have post-fader insertion (a pre-fader insert with SPAT SEND PI will send audio to *SPAT Revolution* prior to your fader automation).



The above best practice ensures as well that you 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 object AUX track per audio track to be sent to SPAT, and insert the SEND plug-in on these object AUX tracks.

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

C.3 Clearing Shared Memory

Some users have experienced an issue where SPAT SEND and RETURN plug-ins are not cleared from the shared memory when used with certain third-party DAW hosts (or seen after DAW crashes). Although this should not happen and have been intensively improved in the latest releases of *SPAT Revolution*, a **Clean Shared Memory** option can fix some connection issues (ghost modules, duplicated modules, modules not connecting when opening session). This can sometimes be seen where SPAT Send and Return modules appear in the SPAT setup page when there is no DAW host software running in the background. It can cause problems, when a host with plug-ins is launched and more SEND and RETURN plug-ins appear to be doubled.

Although rebooting the computer would fix this issue, the workaround if this is happening with your particular third-party software, is to invoke a special debug action called *Clean Shared Memory*. It is available by the *Help* menu *Help/Clean Shared Memory*

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

C.4 Performance issues

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

- Lower UI frame rates.
- Turn the 'Nebula Alpha' to zero in your rooms.
- Lowering Reverb Density to 8x8 for all Rooms.
- Lower the automation rate via the engine preference
- Adjust the Multi-Core Parallel Computation Algorithm via the engine preference.

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

The latest release of *SPAT Revolution* includes a new Multi-Core Parallel Computation Algorithm. In the preference of *SPAT Revolution*, in the Engine section at the bottom, you can

choose various computer hardware presets that match your setup. This is a simple step to tuning the computation algorithm.

If you are still experiencing performance and sync issues, you may want to ensure that your hardware configuration meets the *SPAT Revolution* requirements and recommendations [B](#)

Consider as well to kill as many processes not required as possible (Wi-Fi/internet, background services and activity).

C.5 Display Performance (CPU) measurements

Included in release version 20.12 is the ability to display performance (CPU) measurements. This can be done in the **Help** menu **Display Performance** option. This can be accessed as well with shortcuts **Shift + Option + Command + P**.



Figure C.3: Display Performance

C.6 Engine Parallel Processing profile

SPAT Revolution v20.12 introduced a new Multi-Core Parallel Computation Algorithm that is key to optimizing your hardware. This is for both the automation and the audio processing. The **Display Performance** option stated above will be our best ally to monitor the results. This can be accessed with shortcuts **Shift + Option + Command + P**.

The Engine Preference section includes three Profiles for parallel processing

Automation Rate provides the ability to set the refresh rate (frequency) of the automation. By default, we use 10.0 ms (100.00 fps). This can be decreased to lower the frequency if you see that automation is becoming a burden for your system with such a fast refresh.

Max Number of Core is your ability to make more or fewer cores available to the *SPAT Revolution* algorithm. By default, it will be at your number of native core. Not hyper-threading. This can be increased or reduced. Watch not to take all cores for a multitasking computer.

Lastly are the 3 presets for you to choose from, to adapt the **parallel profile** to your use:

- **Max distribution** is a preset which helps to spread the load as much as possible to all cores. It is mainly recommended for Desktop systems and dedicated real-time SPAT computers.

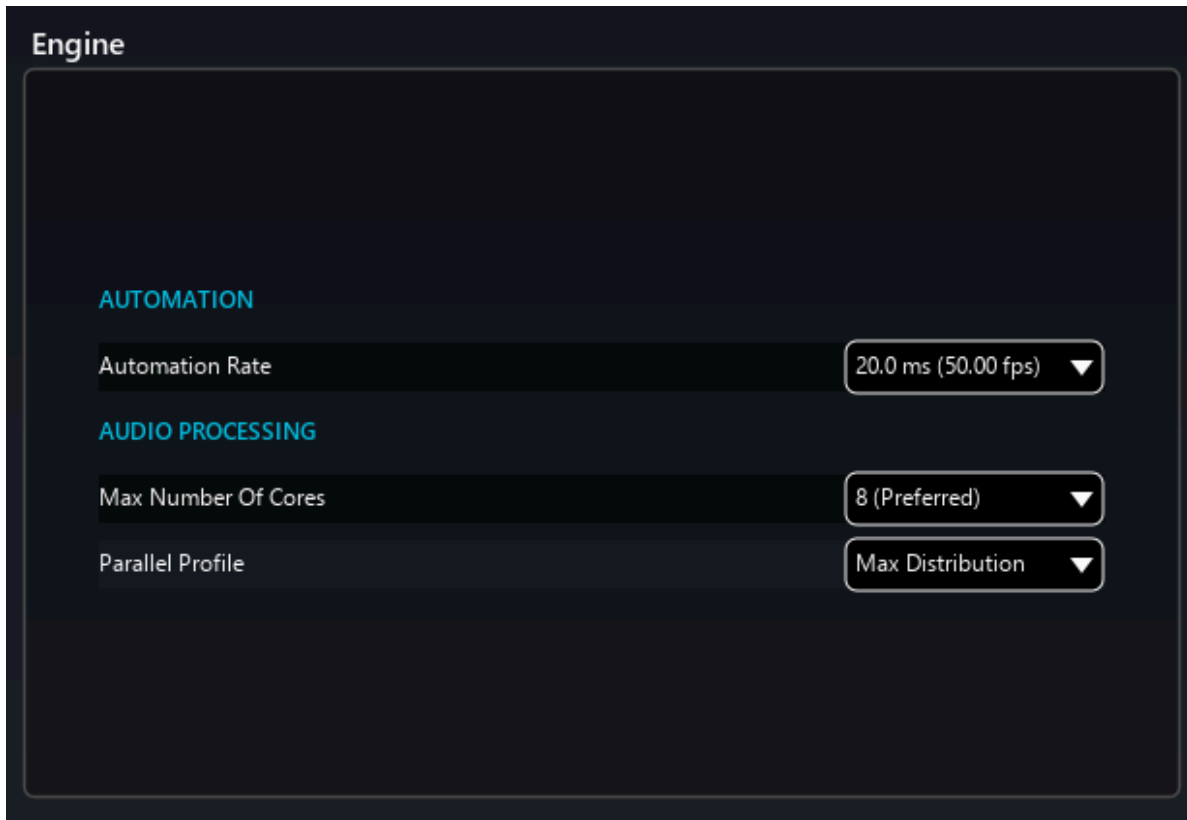


Figure C.4: New updated Engine Preference Section



Figure C.5: Automation Rate Optimization

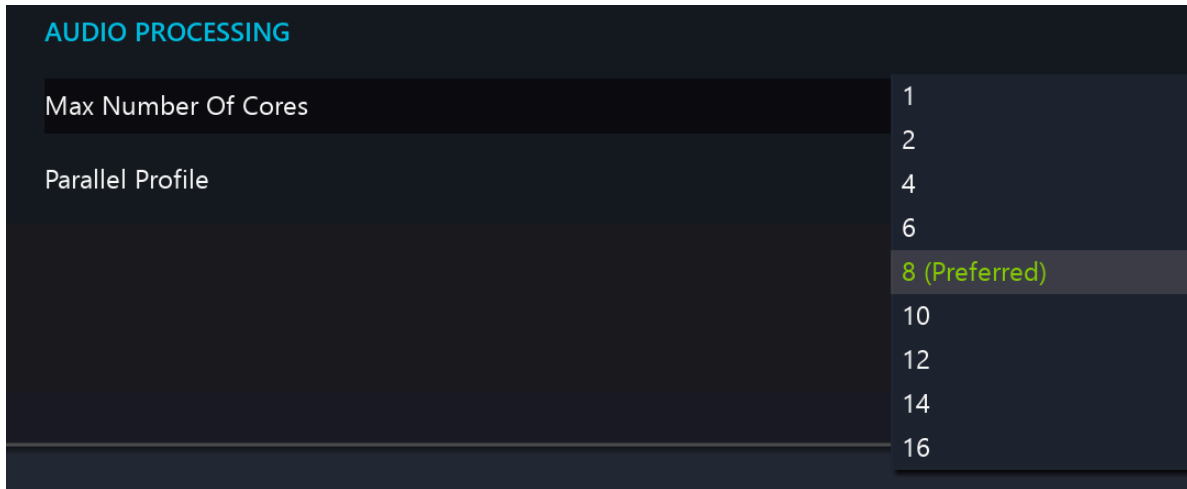


Figure C.6: Max Number Of Cores - Engine Preference

- **Favor first core** is a typical preset for laptop computers where we find that loading cores as much as possible, allowing us to get out of the way of some laptop optimization (power and cooling) that are playing with available processing speed.
- **Balanced distribution** is a preset somehow in the middle. It intends to be a balance between both above options.



Figure C.7: Parallel Processing Algo Profile

D Appendix C - OSC Table

This table lists all the OSC message supported in *SPAT Revolution*.

The table is divided into different messages' categories :

- Project-related messages
- Sources-related messages
- Rooms-related messages
- Masters-related messages
- Snapshots-related messages

The complete document can be downloaded here: [OSC Commands for *SPAT Revolution*](#)

D.1 OSC Grammar

FLUX:: *SPAT Revolution* supports by default 3 input grammars: FLUX:: OSC grammar, ADM-OSC grammar and IRCAM ADMix positional grammar.

D.2 OSC Output Options

For position / radiation messages such as XYZ and AED, it supports individual messages, abbreviated individual messages, packed messages (XYZ or AED) or partially packed messages (XY or AE or AD). On OSC Output, various OSC connections options are available to configure a custom OSC output. 1) AED or/and XYZ packing can be forced on output 2) Auto-Bundle features of OSC can be enabled or disabled. 3) Messages can be sent with Index as Argument rather than in the message (aka `/source/aed [1, 45, 45, 2]` instead of `/source/1/aed [45, 45, 2]`) 4) The ping-pong feature will force send messages coming from an OSC source back to it (by default, we send all message incoming to all output destination except destinations that are the source of the message. The destination which is defined as the source is the output with same IP address and a port number in the range of [input OSC port / input OSC port +10]. Example : for input 127.0.0.1 Port 8000, Output 127.0.0.1 port 8009 is considered as the same device, port 8011 as another device). 5) Touch/Release messages are used to nicely integrate with DAW Automation features, this can be disabled on output if required or if messages are not supported by destination 6) Alternatively, ADM-OSC grammar can be used on output rather the standard FLUX:: OSC grammar

Part X

SPAT Revolution - Release notes

Here are the release notes of the last SPAT Revolution versions.

- [Version 23.11.50341](#)
- [Version 23.08.50318](#)
- [Version 23.08.50316](#)
- [Version 22.10.50222](#)
- [Version 22.09.50217](#)
- [Version 22.09.50213](#)
- [Version 22.09.50206](#)
- [Version 22.09.50200](#)
- [Version 22.02.50151](#)
- [Version 22.02.50148](#)
- [Version 21.04.50030](#)
- [Version 21.04.50020](#)
- [Version 20.12.49943](#)
- [Version 20.12.49930](#)
- [Version 20.12.49890](#)
- [Version 20.12.49880](#)

Build 23.11.0.50341

New Features

- New record feature, allowing the audio record of the inputs, rooms, masters, transcoders and outputs audio, directly into SPAT.
- 3D View - New Nebula scaling option to auto-scale the nebula outside the speaker arrangement
- 3D View - New auto-hide option to hide sources without signal. The threshold can be chosen on the Preferences page
- 3D View - New zone alpha option to fade the protection and efficiency zones

Improvements

- 3D view - Increase scale max value to 1000%
- OSC - Add broadcast IP address on macOS to allow to listen to broadcast messages
- OSC - Send source color
- Preferences - Improve preferences version handling, to limit the removal of the preferences when changing version.
- Preferences - Add a warning to propose deleting the preferences when coming from a more recent version
- Snapshot - New action which deletes all unused snapshot history, in order to have lighter files
- Speaker arrangement - Add category with Apple and MPEG-H speaker configurations
- UI - Hide tracking parameters when the tracking is disabled

Bugs

- 3D View - Impossible to move a source muted: it is not considered as part of the selection
- 3D view - Issue with background file on the room page
- 3D View - Speaker orientation follows the (0,0) point when moving the listener
- 3D view - Speaker orientation follows the Center and not the listener when moving the listener

- 3D view - When sources are solo, groups and presence infos of other sources alpha are not managed
- Ambix decoding preset sometimes loads N3D instead of SN3D
- Audio-Technica BP3600 microphone arrangement routing is not correct
- Changing a source's speaker arrangement causes bad init of the sub-sources
- OSC - `"/source/-1/dump"` is caught as an invalid message
- OSC - `/source/1/select/?` does not work
- OSC - A snapshot count should be sent when deleting a snapshot
- OSC - Color and name are not sent when receiving `"/source/[index]/dump"`
- OSC - Crash when receiving `"/source/-1/select"`
- OSC - Issue with remote number and group
- OSC - Object name is sent before the object count
- OSC - On socket change, snapshot properties should be sent as other objects
- OSC - On source channel count change, OSC messages are sent for sub sources
- OSC - Selecting an OSC Output socket on the preferences should not dump all the parameters
- OSC - Snapshot next and previous should be sent on session opening
- OSC - Snapshots should send their properties on creation
- OSC - Some snapshot OSC message to ask for a property does not work
- OSC - Source select message does not send index as argument but -1
- OSC - Wildcard as index does not work for rooms
- Room view - `Cmd/Ctrl+A` select all sources even if there are some filters on the source list
- SPAT Send - Nuendo - Write automation is not possible, on both OSC and LAP on Windows, and on LAP only on macOS.
- Setup - No scrollbar when moving some blocks on the setup page
- Setup - Reorder room does not reorder the top bar
- Setup - input config badly set from speaker editor for Input and Source blocks (crash for sources)
- Setup page - Changing the speaker setup category of a master or a transcoder does update the speaker setup list
- Setup page - Issue with object naming and number
- Setup wizard - Speaker categories are not available
- Snapshot - A disabled snapshot can be recalled by the list on the snapshot toolbar
- Snapshot - It should be impossible to enter strings on snapshot timing
- Snapshot - Recall name of sources with snapshots does not work
- Snapshot - Reindex of snapshot does not work
- Snapshot - Remove some menu items
- Source list - Grouping sources open all groups
- Speaker editor - Duplicate and create a new speaker arrangement should trigger "custom" category field
- Speaker editor - Duplicating a speaker setup keeps the tag of the original setup
- Speaker editor - Group options should not be available when transforming speakers

- Speaker editor - It should be forbidden to delete a speaker setup used by other blocks
- Speaker editor - Sometimes, speakers are not unselected when clicking on another speaker, especially when the selected speakers are not visible on the list
- Speaker editor - Speaker orientation should not change when editing speaker cartesian coordinate
- Speaker editor - Speaker autoscaling is too big
- Speaker editor - When moving down or up speakers, side-bar should follow, especially when the target is not visible
- Speaker editor - When opening the speaker arrangement, audio gets briefly interrupted
- Speakers category can be wrong for custom speakers
- UI - “Reset all” of sources does not update the panel for non-position properties
- UI - Spreading and LFE are shown after collapsing and uncollapsing the panel

Known Issues

- App - Bad GUI initialization on Windows with intel graphics chipsets
- App - Core Audio headphones and loudspeakers cannot be used without an aggregate
- Nuendo - Room plugin cannot write automation
- PI - AAX - Renaming the send track in protocols cuts the audio if verb switches are not the default one.
- S6L integration - SPAT Send track name is not retained properly and sent every time the UI opens (erasing your SPAT Source name).
- SPAT Send - Renaming an instance cut the audio in Protocols/Windows10. Have to re-enable the LAP to fix
- App - A process still running after quitting SPAT Revolution when ASIO4ALL is used as audio driver on Windows
- Display issues when OS display scaling are under the default resolution
- On reverb preset density change, sound is cut
- SPAT Plugins - Crash on reloading mixer with Pyramix versions earlier than 14 & Ovation versions earlier than 10
- SPAT Plugins - Tail reverb is not reported to the plugins in LAP when IO devices are “None” in SPAT Revolution
- SPAT plugins - The second OSC output status is not retained in AAX / AAX VENUE after a reboot or an AAX host rack reset

Build 23.08.0.50318

New Features

- ARM support for SPAT Revolution and the plugins suite
- Add Auto Touch/Release for remotes that are not doing it
- Add OSC send option for selection
- Add absolute / trim when tweaking multiselection property
- Core - Propose to empty preferences when crashing on load
- DSP - Add phase inversion of the signal for Inputs, Sources, Rooms, and Masters
- Group of sources
- New speaker arrangements with all main DAW arrangements, with new arrangement categories
- Plugins suite available as VST3
- Speaker editor - Add AFMG EASE new XLD support to import speaker arrangement

Improvements

- 3D View - Display signal activity in real source position and group, like in standard sources
- Add Stereo to LCR encode on the input block
- Add signal generator options
- Better readability of Question and info Dialog box
- Better readability of Transform dialog
- Core - Performance improvements in automation, OSC, snapshots and audio processing
- Core - Source LFE channel is now sent to non-LFE channels if the room does not have an LFE
- Engine - Preferred number of cores on macOS is now performances core minus one. This provides better results with OSC, automation, and snapshots
- HRTF - Add a message to prevent when the default HRTF library is missing, and check if the HRTFs file exists before loading a session.
- OSC - Improve bi-directionality for snapshots. Each snapshot change sends an OSC message. It also accepts range, * flag, “get”, “/?” and “dump”.
- OSC - Positions are not sent when dumping all properties through OSC on opening a session

- OSC - Send previous, current, and next snapshot name
- On Mac OS, clicking on the close (red) button may now ask to quit the application
- Source Header - Name requires now to double-click on the label to edit it
- feed signal generator before processing delay on the input block

Bugs

- 3D view - Barycenter is not updated when selecting several sources through the 3D view
- App - Reported latency is not accurate when Device In/Out are the same on macOS
- At session load, plugins transmit their automation through both OSC and LAP when LAP is engaged.
- Core - Changing IO Block Size may corrupt audio or crash
- Core - Changing the Engine's number of cores may kill audio playback
- Core - LFE of a source does not pass automatically to room LFE
- Delay comp mechanism - Reloading a ProTools session with input delay in SPAT Send doesn't propagate the delay value to the SPAT inputs until a value is changed
- Engine - M2 machines do not have the correct number of preferred core
- Essential - Reactivate a block does not work
- Fix some crashes with ASIO
- Fix some crashes with ioconfig
- Items page - select All and unselect all actions that are missing
- Nuendo - No automation recorded in either OSC and LAP
- Nuendo - When writing automation in touch, plugins write automation without touching the plugin.
- OSC - ADM-OSC does not support multiple LFE
- OSC - Issue with /project/count/source in adm/osc
- OSC - Multiselection values are not updated when the source is edited via OSC message
- OSC - On item creation, OSC messages are sent even if OSC is disabled
- OSC - On session load, items' count should be sent only once
- OSC - Rotation X,Y and Z buttons aren't refreshed on automation and OSC message reception
- OSC - Select message for sources use number and not remote index
- OSC - Selecting a source when receiving AED message blocked his azimuth and elevation
- OSC - Selection is not sent for source header parameter (mute, gain, solo)
- OSC - Some ADM-OSC messages asking for a dump are processed as invalid
- OSC - Source remote number is now unique on source creation
- OSC - The Multiselection values are not sent to OSC when another property than position is changed via snapshot or OSC sources.
- OSC - When remote Index is not default one, /source/*/[...] message does not send all sources
- OSC - Avoid sending several time the name when loading a session

- OSC - Send snapshot properties on session load
- OSC - Snapshot count is sent several times when loading a session
- PI - Reset Values doesn't send automation point to LAP PI
- Preferences - Shortcuts dialog is broken
- Protools - WIN - More than one return in LAP makes the DAW crash
- ReIndex in Spat Revo Plugins
- Room - When multiple sources are selected, moving some "filtered" parameter like distance in the property panel may generate bad position values
- SPAT Send - Crash Pyramix when closing session if LAP enabled
- SPAT Send - Index is not increased when duplicating the send.
- SPAT Send - When adding a plugin, it sends via OSC the default values, and reset the source in SPAT
- Setup - Changing the stereo panning law does not work
- Setup page - Connection issue with transcoders when input HOA preset is B-Format or Custom with SID sorting
- Snapshot - Removing active version history breaks some snapshot functions, such as recall
- Snapshot - Rooms setup parameters are recalled with snapshot when "Room recall" is enabled
- Snapshots - Multiselection values are not updated when source is edited via snapshots
- Snapshots - The snapshot list of the snapshot toolbar is black when the list is higher than the windows
- Snapshot - Previous and next snapshots are not updated when a snapshot is activated or deactivated
- Speaker editor - Mouse wheel does not work on Speaker configurations export dialog
- Speaker editor - Speakers orientation with delay is not the expected one in certain cases
- Speaker editor - Temporary room created when editing the speaker arrangement send OSC message on creation
- Speaker editor - Trying to create a speaker arrangement with a name already existing does not create one, nor display an error
- Sub sources of a multi-channel source can jump sometimes when automated from remote
- UI - On grid item, selecting the last visible row scrolls down to the next element
- UI - Setup page - UI issue when loading session and an input is connected to several sources
- UI - Spectral axis and spectral omni are not refreshed when multi selection and OSC messages or snapshots
- UI - The mouse wheel does not work on certain controls on the Preferences and the Setup Page
- UI - The speaker arrangement list does not display well on non-HDPi screens when too many speaker arrangements
- Win10 - VST3 - Nuendo && Pyramix - Crash when reloading a session with SPAT Return and LAP enabled
- Windows only - Crash at start when an OSC input is disabled

- macOS - Crash message can appear on some Intel machine at start, although the software starts

Known Issues

- App - A process still running after quitting SPAT Revolution when ASIO4ALL is used as audio driver on Windows
- App - Bad GUI initialization on Windows with intel graphics chipsets
- App - Core Audio headphones and loudspeakers cannot be used without an aggregate
- Display issues when OS display scaling are under the default resolution
- Nuendo - Room plugin cannot write automation
- OSC - Broadcast OSC messages are not received on macOS. Have to use unicast
- PI - AAX - Renaming the send track in protocols changes the corresponding source verb enable, early, cluster and tail
- S6L integration - SPAT Send track name is not retained properly and sent every time the UI opens (erasing your SPAT Source name).
- SPAT Plugins - Crash on reloading mixer with Pyramix versions earlier than 14 & Ovation versions earlier than 10
- SPAT Plugins - Tail reverb is not reported to the plugins in LAP when IO devices are “None” in SPAT Revolution
- SPAT Send - Renaming an instance cut the audio in Protocols/Windows10. Have to re-enable the LAP to fix
- SPAT plugins - The second OSC output status is not retained in AAX / AAX VENUE after a reboot or an AAX host rack reset

Build 22.10.0.50222

Bugs fixes

- SPAT Send - Fix: Selection is recorded and played back
- macOS - Fix SPAT crashes in some circumstances (native AVB on macOS, ...) depending of the OS version.
-

Known Issues

- App - A process still running after quitting SPAT Revolution when ASIO4ALL is used as audio driver on Windows
- App - Bad GUI initialization on Windows with intel graphics chipsets
- App - Core Audio headphones and loudspeakers cannot be used without an aggregate
- App - Reported latency is not accurate when Device In/Out are the same on macOS
- OSC - Broadcast OSC messages are not received on macOS. Have to use unicast
- OSC - Position are not sent when dumping all properties through OSC on opening a session
- OSC - Rotation X,Y and Z buttons aren't refreshed on automation and OSC message reception
- OSC - Selecting a source when receiving AED message blocked his azimuth and elevation
- PI - AAX - Renaming the send track in protocols changes the corresponding source verb enable, early, cluster and tail
- Room - When multiple sources are selected, moving some "filtered" parameter like distance in the property panel may generate bad position values
- S6L integration - SPAT Send track name is not retained properly and sent every time the UI opens (erasing your SPAT Source name).
- SPAT Plugins - Crash on reloading mixer with Pyramix versions earlier than 14 & Ovation versions earlier than 10
- SPAT Plugins - Tail reverb is not reported to the plugins in LAP when IO devices are "None" in SPAT Revolution
- SPAT Send - Renaming an instance cut the audio in Protools/Windows10. Have to re-enable the LAP to fix
- SPAT Send - Crash Pyramix when closing session if LAP enabled
- SPAT plugins - The second OSC output status is not retained in AAX / AAX VENUE after a reboot or an AAX host rack reset

- Setup page - Connection issue with transcoders when input HOA preset is B-Format or Custom with SID sorting

Build 22.10.0.50219

Improvement

- Speaker editor - Cartesian delay computation for up/down speakers

Bugs fixes

- 3D view - Issue with scale of source when an automatic scaling is applied on the speaker editor
- Grid - Double-clicking on boolean shows “true/false” instead of changing the state
- LAP - Distance Scaling is re-apply on source selection
- Room page - Collapsing a panel on the source or room panel does not work after rebuilding the GUI.
- Setup page - Issues with translation and items inspector
- Snapshot - Drop audio when creating, duplicating, moving up and down or removing a snapshot
- Snapshot - Issues with Relative recall and scaling
- Snapshot - Issues with snapshot list and version list when rebuilding the GUI
- Snapshot bar - Hiding the snapshot bar does not hide the background bar
- Speaker editor - Delays are wrongly applied when speakers have elevation and are not oriented toward the listener
- UI - When rebuilding the GUI, some 3D View parameters are not reapplied (scaling, output and number of view)

Known Issues

- App - A process still running after quitting SPAT Revolution when ASIO4ALL is used as audio driver on Windows
- App - Bad GUI initialization on Windows with intel graphics chipsets
- App - Core Audio headphones and loudspeakers cannot be used without an aggregate
- App - Reported latency is not accurate when Device In/Out are the same on macOS
- OSC - Broadcast OSC messages are not received on macOS. Have to use unicast

- OSC - Position are not sent when dumping all properties through OSC on opening a session
- OSC - Rotation X,Y and Z buttons aren't refreshed on automation and OSC message reception
- OSC - Selecting a source when receiving AED message blocked his azimuth and elevation
- PI - AAX - Renaming the send track in protocols changes the corresponding source verb enable, early, cluster and tail
- Room - When multiple sources are selected, moving some "filtered" parameter like distance in the property panel may generate bad position values
- S6L integration - SPAT Send track name is not retained properly and sent every time the UI opens (erasing your SPAT Source name).
- SPAT Plugins - Crash on reloading mixer with Pyramix versions earlier than 14 & Ovation versions earlier than 10
- SPAT Plugins - Tail reverb is not reported to the plugins in LAP when IO devices are "None" in SPAT Revolution
- SPAT Send - Renaming an instance cut the audio in Protocols/Windows10. Have to re-enable the LAP to fix
- SPAT Send - Crash Pyramix when closing session if LAP enabled
- SPAT plugins - The second OSC output status is not retained in AAX / AAX VENUE after a reboot or an AAX host rack reset
- Setup page - Connection issue with transcoders when input HOA preset is B-Format or Custom with SID sorting

Build 22.10.0.50213

Improvements

- German localisation update
- OSC - Add “SPAT Remote server” OSC preset
- OSC - Send Select OSC message on source selection and unselection
- Spanish localisation update
- Speaker editor - Add “up” and “down” orientation preset for speaker

Bugs

- Core - Changing settings WFS may stop the audio engine
- Empty OSC messages since 50208
- OSC - /room/1/name/? crash SPAT Revolution
- OSC - /source/1/dump message doesn't work on Windows10
- OSC - Global/project/... OSC command query & send are not working properly
- OSC - Some OSC In/Out Transformation do not working properly
- OSC - Some source config OSC messages are not working
- OSC - Unicode char or Korean char not supported in Table (Items/Speakers/Snapshot)
- Speaker editor - Delay/Gain compute can be wrong with triple lines of speakers
- Windows - Some UI refresh issues with the bottom of the window

Known Issues

- Important - When sources are SPAT Send and a room has a scaling, selecting the source applying the scaling. Avoid using scaling with LAP (use loopback solution or audio bridge on macOS).
- App - A process still running after quitting SPAT Revolution when ASIO4ALL is used as audio driver on Windows
- App - Bad GUI initialization on Windows with intel graphics chipsets
- App - Core Audio headphones and loudspeakers cannot be used without an aggregate
- App - Reported latency is not accurate when Device In/Out are the same on macOS

- OSC - Broadcast OSC messages are not received on macOS. Have to use unicast
- OSC - Position are not sent when dumping all properties through OSC on opening a session
- OSC - Rotation X,Y and Z buttons aren't refreshed on automation and OSC message reception
- OSC - Selecting a source when receiving AED message blocked his azimuth and elevation
- PI - AAX - Renaming the send track in protocols changes the corresponding source verb enable, early, cluster and tail
- Room - When multiple sources are selected, moving some "filtered" parameter like distance in the property panel may generate bad position values
- S6L integration - SPAT Send track name is not retained properly and sent every time the UI opens (erasing your SPAT Source name).
- SPAT Plugins - Crash on reloading mixer with Pyramix versions earlier than 14 & Ovation versions earlier than 10
- SPAT Plugins - Tail reverb is not reported to the plugins in LAP when IO devices are "None" in SPAT Revolution
- SPAT Send - Renaming an instance cut the audio in Protocols/Windows10. Have to re-enable the LAP to fix
- SPAT Send - Crash Pyramix when closing session if LAP enabled
- SPAT plugins - The second OSC output status is not retained in AAX / AAX VENUE after a reboot or an AAX host rack reset
- Setup page - Connection issue with transcoders when input HOA preset is B-Format or Custom with SID sorting

Build 22.02.0.50206

Bug fixed

- Any setup action in the setup page unmute ALL rooms
- Delay value on the Item list - Feets value appears wrong , giving same value as meters when your system is not in metric
- Grids - Hitting enter after renaming the last element of the grid does not close the edit box
- Linear2dB transformation on ADM preset - Out of range message Overloads SPAT for good
- Listener Z position , pitch roll, greyed out even when room config should permit it
- Menu doesn't close
- Mirror transformation is broken
- OSC - /adm/obj/1/name/? message replies with /source/1/name even if the port is set to ADM-OSC
- OSC - Dump at session creation appears incomplete
- OSC - ADM-OSC transform issue, scaling only on a single axis for all Incoming axis, after recalling snapshot
- Overload if changing to an arrangement with one using .1 (LFE) channel
- Preset 1.1 is broken: no LFE
- QLab4 - QLab5 AU Crash when instantiating
- Room duplication display a stereo speaker array in the 3D view, no matter of the original room config
- SPAT Revolution crash when the buffer size is set to 64 samples
- Select All sources becomes very long and CPU intensif
- Setting "Depth" parameter to zero generate an overload state that cannot be undo
- Snapshot - Update message dialog appears when updating from OSC messages
- Snapshot - Active version history is not correct when opening the session
- Snapshot - Propagate issue with snapshot history
- Snapshot Page - Snapshots created from OSC do not refresh on UI
- Snapshot - New shortcut Shift + Space (Mac and Windows)
- Snapshot list - double clicking on an empty part of the list makes SPAT crash
- Snapshot menu - Undefined name gives only one snapshot in the menu even if multiple
- Speaker automatic orientation: wrong orientation for some speakers in typical live setups
- Crash with Korean localisation

- Issues with Italian localisation
- macOS - Doesn't work in Sierra, High Sierra and Mojave

Known Issues

- App - A process still running after quitting SPAT Revolution when ASIO4ALL is used as audio driver on Windows
- App - Bad GUI initialization on Windows with intel graphics chipsets
- App - Core Audio headphones and loudspeakers cannot be used without an aggregate
- App - Reported latency is not accurate when Device In/Out are the same on macOS
- OSC - /source/1/dump message doesn't work on Windows10
- OSC - Broadcast OSC messages are not received on macOS. Have to use unicast
- OSC - Position are not sent when dumping all properties through OSC on opening a session
- OSC - Rotation X,Y and Z buttons aren't refreshed on automation and OSC message reception
- OSC - Selecting a source when receiving AED message blocked his azimuth and elevation
- PI - AAX - Renaming the send track in protocols changes the corresponding source verb enable, early, cluster and tail
- Room - When multiple sources are selected, moving some "filtered" parameter like distance in the property panel may generate bad position values
- S6L integration - SPAT Send track name is not retained properly and sent every time the UI opens (erasing your SPAT Source name).
- SPAT Plugins - Crash on reloading mixer with Pyramix versions earlier than 14 & Ovation versions earlier than 10
- SPAT Plugins - Tail reverb is not reported to the plugins in LAP when IO devices are "None" in SPAT Revolution
- SPAT Send - Renaming an instance cut the audio in Protocols/Windows10. Have to re-enable the LAP to fix
- SPAT Send - Crash Pyramix when closing session if LAP enabled
- SPAT Send - Crash when changing the insert slot while GUI is opened in Logic
- SPAT plugins - The second OSC output status is not retained in AAX / AAX VENUE after a reboot or an AAX host rack reset
- Setup page - Connection issue with transcoders when input HOA preset is B-Format or Custom with SID sorting

Build 22.02.0.50200

New Features

- New Snapshot page to ease snapshot overview and edition
- New parameters to enable or disable source's automation, OSC remote, or snapshots
- New Items page to ease items overview and edition
- Add delay setting on Input items, also manageable via a new "Input wizard delay"
- Add "SR Essential compatibility" mode and session check to SR Ultimate
- Speaker editor - Add "fix automatic orientation" and "Set chosen orientation" actions
- Speaker editor - New delay computation management dialog
- Setup page - Add manual activation/deactivation of items
- OSC - Add name to OSC slot
- OSC - Add an option to display the invalid messages for easier debugging
- SPAT Send - Add an "input delay" parameter in the plugin, used for the delay compensation mechanism in SR
- New Italian and Korean translation

Improvements

- App - New shortcut to create a snapshot: Ctrl+Alt+Space
- Speaker editor - Automatic scale added to the 3D view depending on max speaker distance
- 3D View - The background now follow the color theme
- 3D view - Remove the "source to listener" line when "presence info" display option is off
- DSP - CPU consumption optimization and parallel computing improvements: more robust and efficient
- DSP - Send to Analyzer feature improved: Audio processing performance with less jitter
- DSP - macOS audio workgroup implementation, allowing better threading optimization
- Preferences - Add pre or post speaker's "Mute/Solo" option for Speaker test signal
- Preferences - Update engine preferred number of cores for Apple M1 processors
- Room - Add solo for speaker
- Room - Generator on room output channels is now after room gain and mute
- SPAT Plug-ins - Allow the same index on different plug-ins
- SPAT Send - Update Distance default value according to Barycentre in SPAT Revolution

- SR Essential - Allow drag & drop and manual connection
- SR Essential - Allow to add blocks manually
- SR Essential - Enable / disable blocks depending on compatibility instead of deleting them
- Setup page - Fix refresh issue
- Speaker editor - Complete overhaul with multiselection, new UI for a safer and faster dialog
- Speaker editor - Add a warning when a factory preset is selected
- Speaker editor - Allow duplicating the speaker before computing delay and gains
- Speaker editor - Improved performances on panning type compatibility check
- Speaker editor - Propose to homogenize speakers' position after spherical alignment
- Speaker editor - Take speakers' orientation into account in delays computation
- UI - Update dialog buttons order

Bugs

- 3D View - Clamped sources may disappear after a while
- 3D View - Efficiency zone processing issues with linear setups non-aligned to an axis
- 3D View - The source spheres may blink when no signal
- 3D View - Virtual source barycenter is not recomputed after applying a transformation
- 3D View - Virtual source barycenter is not updated when resetting position or all
- 3D View - Wrong source real position displayed when the listener is moved and/or oriented
- 3D view - Efficiency zone display issues when moving the listener
- 3D view - Wrong clamping on frontal system when the source is over the depth efficiency zone and out of the "azimuth triangle"
- App - ALT modifier doesn't work on Windows - Shortcuts on Windows is so limited
- App - Crash when exiting while the speaker editor is open
- Speaker editor - Colinear frontal systems are no more allowed in DBAP
- DSP - Cached reverb tail may be heard when un-muting a source
- DSP - When reloading a session with at least two rooms and SPAT sends, there are issues with the listener position computing.
- App - Double clicking on "Input stream" on the status bar resets to default the hardware buffer and sampling rate if no Send/Return
- Invalidation issue on entering a graphical element when tooltips are deactivated
- LAP - Reset a parameter in SPAT Revolution does not write the value
- Meters - Large multi-channel meters are badly displayed
- On session opening, source's name are not sent on OSC dump
- Performance display histogram show a non existing peak on show.
- Plug-ins - Send and Return LAP does not work if no Hardware IO selected
- REAPER/WIN10 - VST - Automation reset source position

- Reset All fonction doesn't work on barycentric parameter
- Return plugin - Report Tail corresponding to the reverb tail to the DAW
- Room - Display issue with source list scrollbar
- Room - Sliders may have random and undesired behaviour
- Room parameter - Grid type, display output is not stored with the session
- SPAT Plugins - Send/Return parameter default values are not correct when using "Local Audio Path" in AudioUnit
- SPAT Return - Flush buffer on play (reset) in AudioUnit
- SPAT Send - Logic Pro - Index issues on reload under Apple M1 processor
- Setup - Bugs changing a speaker arrangement which is used on several room on session load
- Setup - Connect a Send input to a source should not reset all parameters
- Setup - Enabling signal generator on an input set it to "Channel-based, Mono"
- Setup Page - No more horizontal scrollbar after using the "RebuildGUI" function
- Setup Page - Scrollbar doesn't show up when location are different to English
- Setup page - Remove items does not refresh the scrollbar
- Snapshots - Distance scaling should not be included in snapshot recall
- Speaker Arrangement modifications are not propagate to all block using the same arrangement
- Speaker arrangement - Error reported when "computing" delay & gain on a co-linear system in VBAP
- Speaker arrangement - Transformations can be wrong for some speakers
- Speaker arrangement - Wrong speakers' automatic orientation when the setup is a square shape with no speaker in the corners
- Speaker editor - "Reset" button reset speakers orientation
- Speaker editor - Excel (XLS) import in speaker arrangement no longer work after file saved
- Speaker editor - Issue with scrollbar when adding speakers
- Speaker editor - Random crashes when using transformation
- Speaker editor - nor reset to zero nor propagate alignment values (delays, gains, and orientation) to other blocks with the same speaker arrangement
- The automatic orientation of lines of speakers is wrong if not aligned with a cartesian axis
- When a block is no more connected to an input but still to the output, it can generate noise
- Windows - Reverb factory preset cannot be recalled
- Windows - UI refresh issues
- [AU Logic] Send/Return/Room plugin: Automation from extern controller with touch/release works only if plugin set on polar mode
- [AU Logic] Send/Return/Room plugin: Automation from external controller without touch/release doesn't work
- [AU Logic] Send/Return/Room plugin: Automation of switches and lists isn't written
- [AU Logic] Send/Return/Room plugin: Automation on multiselection doesn't work

- [AU Logic] Send/Return/Room plugin: Renaming a track reset some parameters
- [AU] SPAT Send plugin properties no retained

Known Issues

- AAX PI - Win10 - Renaming an instance cut the send of the SPAT send. Re-enable the LAP to fix it.
- AU - Logic - Changing a send plugin from slot with GUI opened may crash
- Bad GUI initialization on Windows
- Broadcast OSC messages are not received on macOS. Have to use unicast
- Core Audio headphones and loudspeakers cannot be used without an aggregate
- HOA preset input: selecting B-Format or Custom with SID sorting may only connect the first channel of the input block
- LAP - Tail reverb is not reported to the plugins when IO devices are “None” in SPAT Revolution
- OSC - Rotation X,Y and Z and Scale buttons aren’t refreshed on automation and OSC message reception
- OSC - Selecting a source when receiving AED message blocked his azimuth and elevation
- PI - AAX - Renaming the send track in protocols changes the corresponding source verb enable, early, cluster and tail
- Position are not sent when dumping all properties through OSC on opening a session
- S6L integration, SPAT Send track name is not retained properly and sent every time the UI opens (erasing your SPAT Source name).
- The second OSC output status is not retained in AAX / AAX VENUE. Make sure to enable it after a reboot or an AAX host rack reset.
- VST - Pyramix - Crash when quitting session with SPAT Send and LAP enable
- VST - Pyramix/Ovation - Crash on reloading mixer with Send/Return with Pyramix versions earlier than 14 / Ovation versions earlier than 10.
- When multiple sources are selected, moving some “filtered” parameter like distance in the property panel may generate bad position values
- Win10 - /source/1/dump message doesn’t work
- Windows - A process still running after quitting SPAT Revolution when ASIO4ALL is used as audio driver
- macOS - Reported latency is not accurate when Device In/Out are the same

Build 22.02.0.50151

Improvements

- Performance - Optimize presence and clamping computation and rendering

Bugs fixes

- Speaker editor - Import from EASE is broken
- Setup page - The diagram scrollbar is not initialized when loading a session from the Home Page
- Crash with multi-canal sources in multiple rooms when disconnecting inputs, causing crash on quitting SPAT or closing session
- Room Gain and “Mute sources outside Efficiency Zone” do not affect LFEs

Known Issues

Most important

- **WARNING:** Generator on room output channel is after gain and mute. Use with precaution, change the level in the generator section of the preference page or on the MASTER module.
- ALT modifier doesn't work on Windows
- Position are not sent when dumping all properties through OSC on opening a session
- macOS - Reported latency is not accurate when Device In/Out is the same
- Issue with source slider which doesn't work all time
- Core Audio headphones and loudspeakers cannot be used without an aggregate
- when multiple sources are selected moving some “filtered” parameter like distance in the property panel may generate bad position values
- S6L integration, SPAT Send track name is not retained properly and sent every time the UI opens (erasing your SPAT Source name).
- REAPER/WIN10 - VST - Automation reset source position
- REAPER/WIN10 - VST - ReaVolution script can't create mono send

- PI - AAX - Renaming a send track in ProTools changes the corresponding source verb enable, early, cluster and tail
- Double-clicking on “Input stream” on the status bar resets to default the hardware buffer and sampling rate if no Send/Return
- Connect a Send input to a source reset all parameters

Important

- Win10 - /source/1/dump messages doesn't work
- OSC - Rotation X,Y and Z buttons aren't refreshed on automation and OSC message reception
- The second OSC output status is not retained in AAX / AAX VENUE. Make sure you enable it after a reboot or an AAX host rack reset.
- Broadcast OSC messages are not received on macOS. Workaround: use unicast

Build 22.02.0.50148

New features

- Sources - Add one gain per room - Parameter name: Room specific gain
- New parameter Room Reverb “Enable” to enable or disable all reverb components for all connected sources in the room
- New HybridVBAP - A new panning technique for 3D arrangements that mixes 2D and 3D VBAP depending on the elevation of the source
- AU plugin integration for Apple Logic Pro up to 10.7 and Figure 53 QLab
- Support of separated hardware input and output devices
- Wave Field Synthesis spatial reproduction technique algorithm available as new panning type - supporting collinear system as well as potentially any 2D/3D arrangements. (WFS add-on license required, Ultimate only)
- New Output Matrix: Allows on Master and Master Transcoder modules several connections per output on channel-based stream type
- New efficiency zone: where it makes sense to position the sources. Two behaviors for sources: clamp on the zone or mute outside
- Speaker orientation on speaker editor, automatically set according to the arrangement shape. This orientation is only taken into account on WFS panning type.
- New sources solo clear button
- New elevation clamping - Source-object is projected on speaker extremum planes for 3D channel-based speaker arrangements
- New option “Source fit speaker elevation” in the protection section of Room output panel: render a 3D creation on 2D setups will be more coherent
- Official support for software input (Local Audio Path via plugin) and hardware output in the same session.
- SPAT Revolution Essential - Now supports up to 16 channel arrangements

Improvements

Spatial Processing

- Reverb parameter - Rename “Reverb Enabled” by “Reverb Tail”
- Hide “Source over Listener” on 2D arrangements

- Improved clamping on 2D and 3D speaker arrangements
- Add special case for azimuth which should interpolate to nearest angle instead of linear range [-180, 180]
- Maximum speaker distance is now the default width protection zone
- Hide Spread factor for WFS and binaural

Speaker Management

- Support of 4 independent LFE with its own send
- Preferences page - Add option to enable/disable the compute action on LFE in Speaker arrangement editor/Room output list
- Elevation is now reset to zero when the setup is detected as 2D

OSC

- Send OSC message when renaming a source and a room
- Add missing /room/1/xy message
- Improve OSC robustness - bad messages preventing and filtering
- Support of ADM-OSC query command (/get, /?)
- Add enable/disable and command log per socket
- Add option to dump properties on OSC socket changes (port, IP and enable)
- Show network port name in tooltips
- Reviewed snapshots workflow and add OSC messages for recall Sources/Rooms/Masters options (/snapshot/options/recall/source, /snapshot/options/recall/room, /snapshot/options/recall/master, /snapshot/options/recall)

Setup

- Save on the project the “Input Type” field of input blocks
- Essential - Setup Wizard - Improve outputs behavior when changing the stream type
- SPAT Essential - Improving handling of panning type on Session compatibility dialog
- SPAT Revolution Essential - Automatically add binaural monitoring when opening sessions made with Spat Revolution Ultimate
- Allow multiple input connections to Input transcoders and sources to permit aggregation from multiple input
- New timecode options sources choice - Absolute as default
- GUI - Align binaural monitoring block with master block on the setup page
- Setup Wizard - Patch input on unused hardware input

3D View

- Speakers and Nebula follow listener position and orientation
- Room reverb graphic display - XY scales improved

Workflow

- Overall AU PI integration robustness
- SPAT Revolution reopens on Home page at startup if no session is automatically recalled
- Room Reverb Preset - prevent from overriding a factory preset

Core optimization

- Huge jitter reduction, especially with small buffer values (e.g. 256 amp @ 96 kHz)
- Up to 20% improvements on single or multi-core configuration

Various

- Add “strict” keyword in source parameter filter to strictly define which parameter to display. e.g. “strict: gain distance”
- Add source parameter filter presets on source parameter panel
- Add Main Menu access from the Main UI Header. Therefore, we can access main menu even in full screen
- Hide system menu in fullscreen
- Improve loading time when starting SPAT revolution when there is no internet connection

Bugs fixes

Spatial Processing

- Add distance scaling support for source “Scale” and “Radius” (if different from default)
- Fix presence vector not drawn over the listener head in certain cases
- Create or edit an arrangement through the setup wizard now change the width of the protection zone and the distance scaling
- Source’s pitch displayed could be wrong
- Multichannel sources rotate the source of 90° when distance set to 0
- Doppler effect is not working anymore

OSC

- Preferences/OSC connections - Fix refresh issue on OSC sockets when IP and port already used
- Reverb density is not recalled by snapshots
- OSC - Fix possible issue with /global OSC command
- Fix OSC messages consistency in regard to packed or unpacked queries
- Preferences/OSC connections - Add missing distance transformation for ADM-OSC presets
- Preferences/OSC connections - Fix possible error on OSC Transform windows
- “/source/[index]/name/get”, “/room/[index]/name/get” and “/master/[index]/name/get” don’t work anymore

3D View

- Fix some drawing issues on clamping, presence factor and radius
- Speaker Alpha setting is now applied to all rooms

Binaural

- Binaural - No direct sound with SphericalHeadModel and SnowManModel
- Add Warning on session opening with an HRTF not available in the computer
- Hide “head scale” for non-binaural panning types
- Fix potential crashes with binaural rooms

Setup

- Fix custom speaker config duplication when importing from a session
- HOA Transcoder - Fix issues with some configurations
- HOA Transcoder - Transcoder do not correctly connect to input channels
- HOA Transcoder - Bad initialization of Yaw and Normalization
- Setup Wizard - Ultimate - Binaural Monitoring not created
- Deleting a room may corrupt different sources parameters
- Binaural monitor block connected on the same master block than an output block controls the master block gain

Speaker Editor

- Setup config lists differ on the speaker editor when opening it through the setup page or the room output panel
- Speaker Transform - Coordinates don't always refresh after transformation
- Selecting or editing an arrangement through the speaker editor output panel does not propagate configuration to connected blocks
- SPAT Revolution Essential - Opening speaker editor through output panel does not show the channels limitation
- Navigation keys add some unwanted chars on speaker names

Workflow

- Better handling of automation and multiple inputs connected to one source
- Removed warning: "Current preset has been modified" when loading a preset

Various

- Snapshot may not reload all properties depending on the language
- Changing language sets all rooms background to black
- Fix import legacy preferences
- Home page - Last saved session is not displayed
- Fix some crashes on exit on macOS
- "Check essential compatibility" menu action is not working
- Crash on Windows with ASIO driver when switching of driver
- Spat freezes when trying to save a file that is not accessible

UI

- Reorder of Rooms doesn't update the Room order in the top bar
- Long room name doesn't display, refresh and resize properly on top bar
- Setup page - Fix alignment issue when changing the Preferences/Global language
- Refresh issues on input wizard, transform panel, on setup page and source list when a scroll bar is present
- Source properties panel - Filter frequencies display issue & improved gain text
- When back to the room, the source parameter panel still shows up while no source selected
- UI Issue with retractable panels

Plugins

- Ableton Live - Automation is broken
- Non-interpolable values like “Position mode” can’t be automated in Audio Unit
- AU parameter automation curve display is not correct
- Crash with Nuendo and SPAT plugins on macOS

Known Issues

Most important

- **WARNING:** Generator on room output channel is after gain and mute. Use with precaution, change the level in the generator section of the preference page or on the MASTER module.
- ALT modifier doesn’t work on Windows
- Position are not sent when dumping all properties through OSC on opening a session
- macOS - Reported latency is not accurate when Device In/Out is the same
- Issue with source slider which doesn’t work all time
- Core Audio headphones and loudspeakers cannot be used without an aggregate
- when multiple sources are selected moving some “filtered” parameter like distance in the property panel may generate bad position values
- S6L integration, SPAT Send track name is not retained properly and sent every time the UI opens (erasing your SPAT Source name).
- REAPER/WIN10 - VST - Automation reset source position
- REAPER/WIN10 - VST - ReaVolution script can’t create mono send
- PI - AAX - Renaming a send track in ProTools changes the corresponding source verb enable, early, cluster and tail
- Double-clicking on “Input stream” on the status bar resets to default the hardware buffer and sampling rate if no Send/Return
- Connect a Send input to a source reset all parameters

Important

- Win10 - /source/1/dump messages doesn’t work
- OSC - Rotation X,Y and Z buttons aren’t refreshed on automation and OSC message reception
- The second OSC output status is not retained in AAX / AAX VENUE. Make sure you enable it after a reboot or an AAX host rack reset.
- Broadcast OSC messages are not received on macOS. Workaround: use unicast

Build 21.4.0.50030

Improvements

- Add special cases for azimuth which should interpolate to the nearest angle instead of linear range [-180, 180].
- Add new Dolby speaker arrangements
- Improve session compatibility
- Update German and Spanish translations

Bug fixes

- Crash possible when opening a session other than from the home page
- Empty Session compatibility when opening a session with master/monitor/output with Sloane 15/27/43
- Fix possible crash with panning validation on Speaker Config Editor
- Fix possible error on Speaker Config Editor
- Fix possible error with Setup wizard validation

Build 21.4.0.50020

This build contains SPAT Revolution and SPAT plug-ins suite.

SPAT Plug-ins suite

New feature

- Position Mode parameter: Polar, Cartesian or None automation recording and reading mode.

Bug Fixes

- Record AED OSC messages and snapshot interpolation via Local Audio Path automation is not working.

SPAT Revolution new features

- New software licenses support for SPAT Revolution Essential (1 activation) and SPAT Revolution Ultimate Bundle (2 activations)
- Ability now to disable the behavior of source/object moving over listener head when entering the spherical protection (radius) zone instead
- New per room spherical protection zone (width parameter)
- Spherical protection zone following the listener position
- New distance scaling per room parameter for source-object automation, OSC and snapshots
- Session compatibility dialog (partial session read) for SPAT Essential and SPAT Ultimate compatibility
- Home Page with session history and resources shortcuts
- Read/write Audio Pipe (LAP) Automation using scaling
- Read/write OSC position messages using scaling
- Store/recall snapshot using scaling
- New stereo panning law options (7)

- Perceptually motivated amplitude panning (PMAP) for Accurate Phantom Image Localization (constant-power amplitude-panning).
- Constant Power and Constant Amplitude
- Square root or Sinus/Cosinus

Improvements

User Interface

- Addition of Real source-object position infos option on 3D view
- Addition of Presence Infos option on 3D View
- 3D view - Source-object is projected on speaker plane for 2D channel-based speaker arrangements
- Sources, Room and Reverb parameters - Hide/show for panels
- Updated Reverb panel unifying related parameters
- Snapshot Rename Popup - Select automatically the edit on window popup

Core behaviors

- Radius protection zone is now automatically set with the channel-based arrangement based on speaker distance average.
- Dialog box when changing speaker arrangement to offer auto scaling room parameters for source-object
- New update Setup Wizard with mode for Essential license based sessions
- Radius per source is now a trim factor expressed in meters for session compatibility and to integrate with new protection zone

Bug fixes

- OSC -1 index doesn't work with Polar / AED. (Requires additional validation)
- Record AED OSC messages and snapshot interpolation via Local Audio Path automation is not working
- Aggregate input on process block does not work + add aggregate input on sources
- Listener head pivot (roll) point should be at the Center
- Fix source pitch not updated when a listener position change
- Listener Yaw not correctly working if the listener is not at the center (origin)
- Multi-sources should keep alignment when elevation is done in polar
- RTTrPM - Tracking scaling not work

- Source presence factor + Listener 6DOF don't work on all rooms and sources for sources connected to multiple rooms.
- UI - HRTF name displayed on the setup block only on the first selection.
- OSC automation of azimuth has an offset of -90
- Refresh issue on Windows
- Reset to default on source parameters may not propagate the setting correctly to the sub source in case of multi-channel sources
- Strange “distance” behavior with multi-channel sources
- Setup Wizard - Crash on deleting a line then press tab
- Sometimes sources do not sound at the right place after reloading a session
- SPAT Revolution does not exit correctly in some cases
- Possible crash on exit
- If a session is closed while the user is in a room, SPAT Revolution main window can turn black
- [FrameRate] GUI problem with low frame rate

Build 20.12.0.49943

Improvement

- Add IO overload Detection + auto mute + reset button
- SPAT Send (software) input sync error detection optimization
- Audio processing optimizations

Build 20.12.0.49930

Bug Fixes

- Solved freezing issues on Windows computers.
- Drop-down menu stuck
- Error when deleting user config in Speaker Arrangement Editor
- Change language sets all rooms background to black (if they were white before)
- When a list is selected in a Dialog, typing escape closes the window instead of the list

Build 20.12.0.49890

SPAT Revolution new features

- New MultiCore parallel processing algorithm.
 - A new preference option to select various engine processing profile.
- Stability improvements (hardening) with all latest operating systems (OS).
- SPAT Plug-ins (Local Audio LAP - AudioPipe) hardening and improvement against host applications and OS.
- Catalina, Big Sur and Apple notarization official support.
- New built in error reporting tools for SPAT Revolution.
 - In SPAT preferences, you can enable or disable sending anonymously of software data error to FLUX::. This application monitoring tools helps track developer errors for better diagnostic, fixes and performance optimization.
- New V20.12 release (new versioning system), now including 1 year of support and upgrade with perpetual license purchase.
- Added 23 reverb presets imported from IRCAM Verb and the SPAT v3 plug-in.
- New Drag and Move + Drag and Connect module feature.
 - Drag any modules to move before or after any module of same type.
 - Drag a drop module into module to connect output to input.
- Support for the IRCAM ADMix/SPAT and Meyer SpaceMap Go OSC Grammar on input.
 - You can now play ADM Master (Audio definition model) files containing audio and object metadata with the [ADMix player](#) playback and receive metadata as OSC data in SPAT
- New Apple and Windows menu and shortcuts
 - SPAT Revolution now includes the standard Apple and Windows menu bar for new and added software action.
 - New shortcut options
- New input naming pages for bulk renaming
- Multi-Selecting input blocks and pressing Alt + N

- Setup wizards;
 - Available by pressing Alt + N or in the Setup menu.
- New Import and Export SPAT preferences.
 - System to System, to back up your preferences.
- Basic snapshot system implementation;
 - snapshots can now be generated, recalled (with interpolation - crossfade option) , deleted or renamed.
 - Gives the ability to create static images instead of using full automation.
 - Basic menu is provided to manage snapshots and available to OSC messages (Lemur and Qlab template available).
- Import speaker arrangements;
 - Import speaker arrangements from simulation software of Adamson, CODA Audio, Nexo, d&b audiotechnik, Ease or simply Excel.
- Speaker arrangement panning tips and validation guidelines.
- New Panning type (LBAP, DualBandVBP)
- New Binaural Options - Nearfield, Spherical Head, Snowman Model
- New A-Format transcoding (Sennheiser Ambeo, Oktava MK4012, Rode Nt SF1)
- New HOA transcoding method (Regularized Pseudo-Inverse, Improved AllRAD, CSAD)

OSC - Many OSC optimizations in the core

- The OSC Matrix (OSC Slots) has been updated and contains various changes.
- Now supporting third party OSC devices without Auto-bundle feature (Digico Consoles now can have bidirectional control)
- Make OSC dictionary available with snapshot
- ADM-OSC
- ADMPlayer compatibility (currently no scale)
- Meyer Sound - Spacemap Go - OSC Grammar support (input only)

Major optimizations

- Open GL fix for Mac Mini (Anti-Alias)
- SPAT Plug-ins (LAP - AudioPipe) hardening and improvement against host App.
- Now supporting third party OSC devices without Auto-bundle feature
- ADM-OSC Format support.
- Retina display enhanced support

- New HOA Decoding Methods
- New Binaural Options - Nearfield, Spherical Head, Snowman Model - (Not using HRTF for some???)
- Per speaker channel mute option in room
- Shift drag to move 3D scene
- Install templates with PI format for SPAT.

Other improvements

- BackGround - Set Background Offset Step to centimeters
- Display Performances (CPU) measurements help option
- Opening a new session doesn't prompt a "save current session" dialog
- Use calling thread (Audio) as thread 0 on parallel processing to avoid using other threads on small configuration.
- Color Indicator update for Mute - Solo
- Draw source and speaker text in 3D View
- Update to last Python version and some other SDKs
- Display units in the speaker/channel editor
- "Output list" for HOA Room should not display x/y/z/Azimuth/etc
- Import speaker arrangement - Check the height of the arrangement and lower all the arrangement.
- Check for update based on current installed version number
- Display "new version" / "update information" at startup
- Add number of channel property to Send/Return plug-in to allow ReaVolution to change to the plug-in format
- Automatically define speaker arrangement on Transcoder when possible
- Add Duplicate room without same sources or with duplicated sources
- Handle "dead" PyControl in UI
- Ircam's exceptions handling in Transcoder
- Dump all properties through OSC output after loading a session
- Pl: remove Independent duration + rename as "interpolation time"
- Pl: update page table
- Rebuild GUI without reloading session

Bug Fixes

- Creating Snapshot freezes on Mac OS X.

- Rename snapshot dialog not correctly initialized ; especially on retina display
- Panning Tips - Display of Coincident speakers was broken.
- Potential error when recalling snapshots.
- Recurring crash in setup while making connections with drag and drop
- Strange issues with input OSC transforms
- Transcoders - Wrong Normalization for AmbiX.
- Yaw/Pitch/Roll - Number of digits in sliders
- Crash on connecting HOA order 7 input to source multiple time.
- Crash when “Max Number of cores” is one
- Crash when build setup without any sources and with Audio Device enabled

Build 20.12.0.49880

This build contains SPAT Revolution and SPAT plug-ins suite.

SPAT Plug-ins suite

Bug fixes

- Fix stability issues with LAP
- Room PI - Fix automation in VST
- Room PI - Fix No automation parameter available in room PI AAX
- Room PI - Fix the non-linear behavior of the listener Yaw Automation
- Send PI - Improve stability on Avid S6L
- Automation - Fix : reset values doesn't send automation point to LAP PI
- VST - Fix latency compensation issues
- AU - Fix automation
- Ableton Live - Fix automation parameter not listed
- Logic Pro X - Fix automation issues from SPAT
- Fix DXD Block-size mismatch
- AAX DSP Version no longer supported

SPAT Revolution bug fixes

Core

- Fix crashes with some ASIO Driver
- Fix listener pitch and roll inverted
- Fix source presence, orientation and distance when moving listener position in the room
- Fix crashes in the setup page
- Fix crashes in the speaker configuration
- Fix crashes tweaking room stream type
- Fix crashes choosing certain speaker arrangements
- Fix session loading time
- Fix issues with same room names

- Fix crashes with HOA transcoders
- Fix Yaw angle in transcoders
- Fix OSC Mute command issues
- Fix crashes deleting empty speaker arrangements
- Fix input transcoding A-Format to CB gain normalization issues
- Fix speaker arrangements duplication issues when importing from session
- Fix source reset all doesn't reset rotation XYZ
- Fix OSC stability and performances
- Fix synchronization error detection

UI

- Fix retina/non-retina issues
- Fix label cropped with other languages than English
- Fix text of linear sliders display
- Fix name of the sources refresh
- Fix parameters on the right hidden on the setup page after scrolling down through a larger parameters list
- Fix renaming a room doesn't reflect the new name in the tab
- Fix double-click on some slider doesn't reset to default value
- Fix incorrect and missing hints
- Fix sources and color duplication
- Fix room background image proportions are not correct for non-square images
- Fix source aperture display glitches
- Fix scrolling bar blocked in the speaker arrangement editor when switching from large arrangement to light arrangement
- Fix selection on deleting a speaker on the speaker arrangement editor
- Fix UI issues with OpenGL mac mini.

Various

- Fix dialogs do not always close or validate with Escape and Return keys
- Fix presets import
- Fix some crashes with Terminal actions
- Fix default position with 5.1 sources
- Fix opening session with non-ASCII characters

Known Issues

- KU100 HRTF not included in the default HRTF list for Essential

- 3D View - Output meters can display only 50 channels
- Shortcuts with Alt doesn't work on Windows 10.
- Only the first LFE receives LFE send.
- [HOA] Crash on transcoding B-Format 2D to HOA3D
- OSC OSC message “/master/[index]/dump” do not work
- OSC No OSC control for reverb modal density
- [PI] LAP with AAX 96K can face synchronization issues.
- [PI] SPAT Send Solo parameter is not working in LAP or OSC
- [PI] Enable Reverb Automation not working
- [PI-VST] Closing Nuendo and Ableton Live session with SPAT PI can crash applications.
- [PI-VST] Loading a project or rebuild the mixer with SPAT Sends/Return can crash Merging Technologies Ovation / Pyramix