



# Gig Performer® 5 User Guide

v5.0.40  
February 2025  
<https://www.gigperformer.com/>

Nemanja Pudar

Copyright © 2016-2025 Deskew Technologies, LLC

All rights reserved.

Gig Performer and Own The Stage are registered trademarks of Deskew Technologies, LLC



<b>1.</b>	<b>Welcome to Gig Performer</b>	<b>9</b>
1.1	Thank you note .....	9
1.2	What is Gig Performer? .....	9
1.3	Who can use Gig Performer? .....	11
1.4	Gig Performer standard features .....	12
<b>2.</b>	<b>What is new in Gig Performer 5?</b>	<b>15</b>
2.1	GP Relayer .....	15
2.2	MIDI Chord Maker .....	16
2.3	Enhanced Streaming Audio File Player .....	16
2.4	Envelope Follower .....	17
2.5	Audio and MIDI activity display .....	18
2.6	MIDI Rechannelizer .....	19
2.7	MIDI Transposer .....	20
2.8	Improved Automation .....	21
2.9	Plugin enhancements .....	21
2.10	Other enhancements .....	23
2.11	GPScript - Breaking Change .....	24
<b>3.</b>	<b>Getting started</b>	<b>26</b>
3.1	Before you start .....	26
3.2	Basic setup for keyboardists .....	28
3.3	Basic setup for guitarists .....	31
3.4	Basic setup for vocalists, drummers, other musicians and FOH engineers .....	33
3.5	Built-in templates .....	34
3.6	How to update Gig Performer .....	36
<b>4.</b>	<b>Gig Performer user interface and concepts</b>	<b>39</b>
4.1	The main Gig Performer window .....	39
4.2	Rackspaces and Variations .....	42
4.2.1	Global Rackspace .....	43
4.2.2	Manage Rackspaces/Variations contextual menu .....	45
4.2.3	Rackspace Properties .....	48
4.2.4	Variation Properties .....	49
4.3	Wiring view .....	50
4.3.1	Wiring view features .....	52

4.3.2	Wiring view contextual menu .....	55
<b>4.4</b>	<b>Panels view .....</b>	<b>57</b>
<b>4.5</b>	<b>Setlist view .....</b>	<b>58</b>
4.5.1	Manage setlists contextual menu .....	61
<b>4.6</b>	<b>Widgets .....</b>	<b>64</b>
4.6.1	Widget contextual menu .....	66
<b>4.7</b>	<b>Plugins .....</b>	<b>68</b>
4.7.1	Plugin Manager .....	72
4.7.2	Plugin contextual menu .....	77
4.7.3	Selected blocks contextual menu .....	82
4.7.4	Plugin Editor More menu .....	85
<b>4.8</b>	<b>Global MIDI Monitor .....</b>	<b>89</b>
<b>4.9</b>	<b>Rig Manager .....</b>	<b>90</b>
<b>4.10</b>	<b>ChordPro .....</b>	<b>96</b>
4.10.1	PDFToGPChordPro .....	100
<b>5.</b>	<b>Inserting and Wiring your plugins .....</b>	<b>107</b>
5.1	How to insert a plugin and connect it .....	107
5.2	Blue and orange colored connections and pins .....	113
<b>6.</b>	<b>Panels and Widgets .....</b>	<b>117</b>
<b>6.1</b>	<b>What is a panel? .....</b>	<b>117</b>
6.1.1	Panel contextual menu .....	119
<b>6.2</b>	<b>Widgets .....</b>	<b>120</b>
6.2.1	How to add widgets to a panel? .....	120
6.2.2	Widget alignment and distribution .....	123
6.2.3	Widget z-ordering .....	126
6.2.4	Widget Properties inspector .....	127
6.2.5	How to connect a widget to a plugin parameter? .....	139
6.2.6	How to connect a widget to a MIDI device? .....	143
6.2.7	How to add a widget to a Widget Link group? .....	145
6.2.8	How to add a widget to a Radio Group? .....	147
6.2.9	How to scale widget values with curves? .....	148
6.2.10	How to control widgets from the Global rackspace in a regular rackspace? .....	154
<b>7.</b>	<b>Songs and Setlists .....</b>	<b>157</b>
<b>7.1</b>	<b>Rackspaces/variations and Song Parts relationship .....</b>	<b>157</b>
<b>7.2</b>	<b>How to create a song? .....</b>	<b>158</b>
7.2.1	Song Properties .....	161

7.2.2	Song Part Properties .....	162
<b>7.3</b>	<b>How to create a setlist? .....</b>	<b>165</b>
<b>7.4</b>	<b>How to capture variation edits? .....</b>	<b>168</b>
<b>8.</b>	<b>Rig Manager .....</b>	<b>171</b>
8.1	How to create an alias for a MIDI device and a MIDI control? .....	171
8.2	How to remap MIDI controls for a new MIDI device? .....	178
8.3	How to use aliases? .....	181
<b>9.</b>	<b>The Tuner .....</b>	<b>183</b>
<b>10.</b>	<b>Built-in Plugins .....</b>	<b>185</b>
10.1	Audio In .....	187
10.2	Audio Out .....	188
10.3	Audio Mixers .....	190
10.3.1	Audio Mixer applications .....	193
10.4	Gain Controls .....	198
10.4.1	Gain and Balance Control .....	199
10.4.2	Gain Control (mono) .....	200
10.4.3	Gain and Balance Control (mono to stereo) .....	201
10.5	Global Processing .....	202
10.5.1	To Global Rackspace .....	202
10.5.2	From Global Rackspace .....	203
10.6	Media Players .....	204
10.6.1	Audio File Player .....	205
10.6.1.1	MIDI metronome .....	210
10.6.2	MIDI File Player .....	211
10.6.3	Streaming Audio File Player .....	216
10.6.3.1	Action Management .....	223
10.7	MIDI Inputs .....	234
10.8	MIDI Outputs .....	243
10.9	MIDI Processing and Monitoring .....	248
10.9.1	Envelope Follower .....	248
10.9.2	MIDI Channel Constrainer .....	259
10.9.3	MIDI Chord Maker .....	260
10.9.4	MIDI Filter .....	264
10.9.5	MIDI Monitor .....	267
10.9.6	MIDI Rechannelizer .....	268
10.9.7	MIDI Transposer .....	269

10.9.8	Scriptlet .....	270
<b>10.10</b>	<b>System .....</b>	<b>273</b>
10.10.1	Auto Sampler Generator .....	273
10.10.2	Auto Sampler Recorder .....	278
10.10.3	Comment .....	280
10.10.4	GP Relay .....	281
10.10.5	System Actions .....	283
<b>10.11</b>	<b>From Rackspace .....</b>	<b>288</b>
<b>10.12</b>	<b>To Rackspace .....</b>	<b>289</b>
<b>11.</b>	<b>Recording your performance .....</b>	<b>291</b>
11.1	Recorder settings .....	291
11.2	How to use the Recorder? .....	293
<b>12.</b>	<b>Time controls .....</b>	<b>296</b>
12.1	Tempo and Time signature .....	296
12.2	Metronome .....	297
12.2.1	Audio metronome .....	298
12.3	Ableton Link support .....	300
12.4	Rackspace or Song tempo overrides .....	301
<b>13.</b>	<b>OSC and Gig Performer .....</b>	<b>303</b>
13.1	How to set up OSC .....	303
13.2	OSC Resources .....	306
<b>14.</b>	<b>Multi-instance support .....</b>	<b>307</b>
<b>15.</b>	<b>Latency measurement tool .....</b>	<b>309</b>
<b>16.</b>	<b>Options and preferences .....</b>	<b>311</b>
16.1	General .....	312
16.2	Display .....	314
16.3	Audio I/O .....	318
16.4	Global MIDI .....	322
16.5	Setlists .....	326
16.6	Recording .....	328
16.7	MIDI Ports .....	329
16.8	OSC .....	330
16.9	Scripting .....	333

16.10	Extensions .....	335
16.11	Locations .....	336
16.12	Tap Tempo .....	338
<b>17.</b>	<b>Keyboard shortcuts</b>	<b>341</b>
<b>18.</b>	<b>GPScript</b>	<b>345</b>
18.1	How to control multiple parameters with a single widget? .....	345
18.2	Local GP Port .....	351
<b>19.</b>	<b>Important workflow tools and tips</b>	<b>354</b>
19.1	Predictive loading .....	354
19.2	Quick Plugin, Favorite or Preset Finder .....	356
19.3	Quick Rackspace/Variation finder .....	357
19.4	Save one or more plugins as favorites .....	358
19.5	Drag and drop support .....	362
19.6	MIDI Assignment List window .....	363
19.7	Widget Mappings List window .....	364
19.8	Plugins In Use .....	365
19.9	Probabilistic Sound Designer .....	366
19.10	MIDI Helper Tool .....	367
19.11	Remap MIDI messages without using the MIDI Filter plugin .....	367
19.12	Additional tips .....	368
<b>20.</b>	<b>Extensions</b>	<b>374</b>
20.1	What are extensions? .....	374
20.2	Enabling/disabling extensions .....	375
<b>21.</b>	<b>Optimizing your computer for a glitch-free audio performance</b>	<b>377</b>
21.1	Latency explained .....	377
21.2	Optimizing your Windows PC for maximum performance and stability .....	378
21.3	Optimizing your MacOS .....	379
<b>22.</b>	<b>Join our community</b>	<b>380</b>
<b>23.</b>	<b>A word from the Gig Performer creators</b>	<b>382</b>
<b>24.</b>	<b>Appendix</b>	<b>384</b>

<b>24.1</b>	<b>License agreement .....</b>	<b>384</b>
<b>24.2</b>	<b>Basic terminology .....</b>	<b>387</b>
<b>24.3</b>	<b>File extensions used in Gig Performer .....</b>	<b>392</b>
<b>24.4</b>	<b>File locations in Gig Performer .....</b>	<b>393</b>
<b>24.5</b>	<b>Cross-Platform Usage .....</b>	<b>394</b>
<b>24.6</b>	<b>Troubleshooting .....</b>	<b>395</b>
<b>24.7</b>	<b>OSC messages .....</b>	<b>397</b>



## 1 Welcome to Gig Performer

Welcome to Gig Performer! Deskew Technologies developed Gig Performer, the world's most versatile audio plugin host application for live performance. With it, you can control your entire rig and instantly access all your instrument and effects plugins—no DAW software is necessary.

### 1.1 Thank you note

Thank you very much for choosing Gig Performer. This software is the result of the finest blend of programming skills and serious touring experience. Gig Performer allows performing musicians to be creative and express themselves on stage using a computer. We put great effort into making Gig Performer easy to use and ensuring you enjoy all its features. This manual will enable you to understand Gig Performer's advanced features better, opening up unlimited possibilities.

We are sure that by allowing you to focus on the music and not the technology, you will enjoy using Gig Performer for all your performance needs.

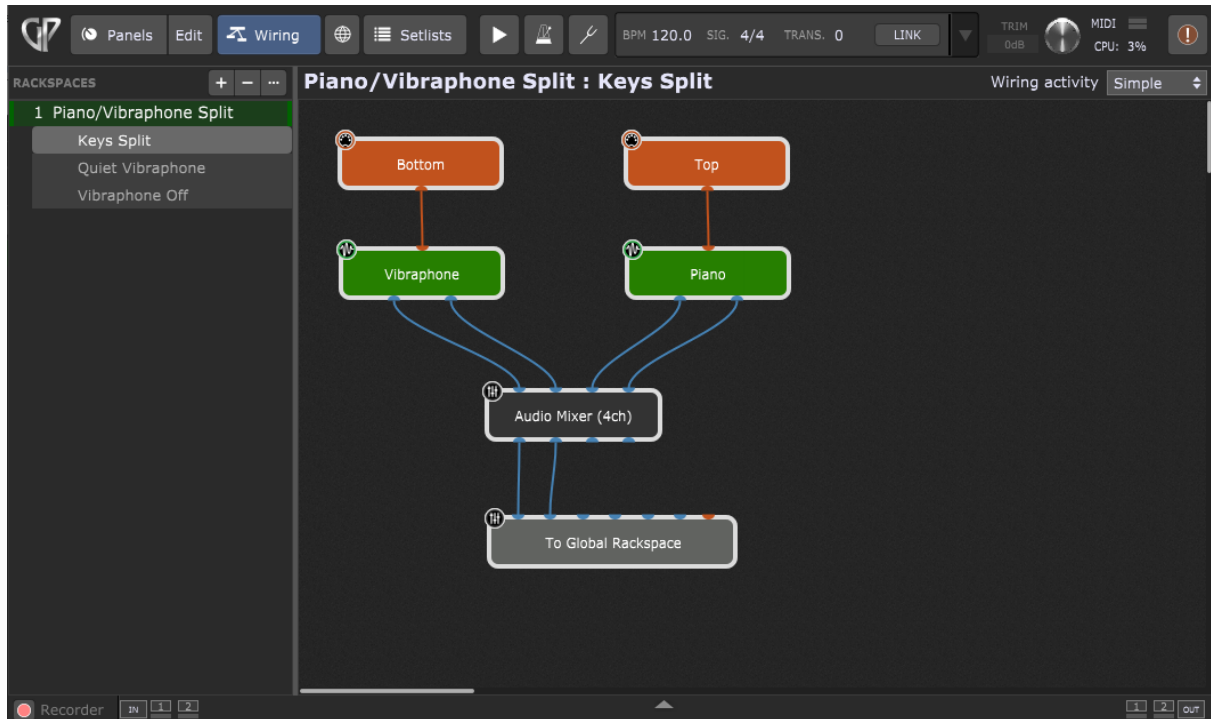
The Deskew Technologies Team.

### 1.2 What is Gig Performer?

Gig Performer is an audio plugin host that provides the most flexible experience for your stage or session performance. As we stated in the previous section, the secret behind its robustness lies in the very finest blend of programming and musical experience: Gig Performer was [developed](#) by people who are both software developers and serious touring musicians. Advanced optimization techniques and heavy stress tests address the requirement for Gig Performer to run reliably on stage and use as few CPU cycles as possible.

Many [well-known](#) music professionals worldwide rely on Gig Performer as their audio plugin host of choice for live performances.

Gig Performer's innovative approach and focus on usability are paired with an intuitive rack design called **rackspace**, which transforms your plugins into the most powerful yet user-friendly live music software system.



You can connect your plugins in any way you wish without creating channel strips, buses, sends or inserts. Read [this blog article](#) to learn why a graphical interface is superior to the channel strip model.

The impressive list of cutting-edge features makes Gig Performer stand out [compared](#) to other audio plugin hosts. For example, the [Predictive loading™](#) feature enables you to intelligently load parts of your show instead of loading your entire show so that you can switch instantly to the next rackspace or Song in your setlist. This feature significantly reduces both your [RAM](#) and CPU usage so you can perform confidently even with a [very old laptop](#).

The extensive OSC implementation in Gig Performer allows you to control your show from a [mobile phone](#), [iPad](#), [Android tablet](#), or [Web browser](#).

As a software product created by musicians, for musicians, new updates and features are often implemented in Gig Performer. Gig Performer is also an **ultimate enhancer** of your whole setup. Read [this blog article](#) to learn how.

You can [download](#) a free, fully functional 14-day trial of this plugin host that truly addresses the needs of live performers.

## 1.3 Who can use Gig Performer?

Every artist who performs on any instrument live or in the studio, as well as front of house sound engineers, can benefit from Gig Performer, as it is an [ultimate enhancer](#) of every aspect of your rig. For example, whether you are a keyboard player, a guitarist, a vocalist, a drummer, or a live sound engineer, Gig Performer has much to offer:

- **Keyboardists** can control all their software synths, samplers, and sequencers - as well as effects - directly from the sliders, knobs and switches on their instruments.
- Using their MIDI pedalboards, **guitarists** can freely switch between multiple amp/cabinet models and digital effects in real time without audio glitching. Check out the blog article [Top 10 reasons you should move to a Gig Performer-based guitar rig](#).
- **Singers** can easily add compression, EQ and other desired effects to their vocals.
- **Drummers** can mix and match their favorite drums and cymbals from different drum plugins to create their perfect drum kits. Drum kits can be switched on the fly to access hundreds of different sounds during your gig instantly.
- **Live sound engineers** can use Gig Performer to create a standalone mixing system or to intercept audio coming from the FOH or monitor system, add desired effects processing, and then send the results directly back to the mixing board. Gig Performer can simultaneously record multichannel audio from the mixing board for later processing and mixing in your favorite DAW.

You can even run multiple Gig Performer instances simultaneously, allowing different band members to access different plugin setups independently as long as the computer is fast enough.

Gig Performer can be used for many other use cases; to learn more, make sure to check out this blog article: [Gig Performer is the best companion for all your musical efforts and more](#) and to visit our [Gig Performer in Action](#) forum and see photos and stories of many great setups with Gig Performer.

Gig Performer's intuitive interface makes it easy to use and remarkably fast to get started. However, many advanced features lie beneath the surface, which we will cover in this User Guide.

## 1.4 Gig Performer standard features

This is an incomplete list of standard Gig Performer features. Be sure to check <http://www.gigperformer.com/features-and-benefits.html> for a complete list of current features.

- [Cross-platform](#) audio plugin host - runs on both macOS and Windows platforms so you can create, share, and modify your files on any computer and such files will run on either platform as long as the needed plugins are available.
- Supports VST2 and VST3 (macOS and Windows) and AU (macOS only) plugins.
- Includes [built-in plugins](#) - MIDI File Player, Streaming Audio File Player, Audio Mixers, MIDI Out to OSC converter plugin, MIDI Channel Constrainer plugin, and many more.
- Minimal CPU usage.
- Rock solid performance, extremely stable, leading to less stress on stage.
- Flexible automation through Timeline actions and Song Part Actions.
- Easily play multiple instruments and controllers, layer sounds, and add effects.
- Intuitive user interface - simply connect any plugin to any other plugin in any configuration; no confusing buses, sends or inserts.
- Build fully customizable [rackspaces](#) consisting of a collection of interconnected plugins along with one or more front panels containing knobs, sliders and other widgets.
- Instantaneous glitch-free switching between sounds, even while holding notes for seamless transition from one song to the next or from one part of a song to another.
- Use keyboards, MIDI pedals, and control surfaces to alter plugin parameters in real-time by connecting them to knobs, sliders, buttons, labels, meters, and other widgets.
- Learn mode for easy assignment of MIDI controls to widgets and widgets to plugin parameters.
- [Build setlists](#) that automatically change to the right sound for each part of each song in the list. Capture on-the-spot tweaks right into song parts. A gig becomes as simple as stepping through its setlist.

- [ChordPro](#) support - create and edit ChordPro song documents and set each to automatically display when its song is selected in the setlist.
- Automatic mixing of signals when multiple outputs are connected to a single input.
- Deactivate any plugin at any time to remove a sound or bypass an effect.
- Widget groups for making cross-faders, grouped level controls, and other ganged behaviors (see [Linked Widget group](#) and [Radio Group](#)).
- The [Rig Manager](#) simplifies switching a Gig Performer rig to use a different set of physical controllers - great for backline touring.
- LEDs show input/output levels on each channel as well as audio and MIDI activity/
- Built-in full-screen [chromatic tuner](#).
- [Audio metronome](#) using built-in or user-supplied sounds.
- Create [multiple instances](#) for different members of the band which run independently or for when you want to control some plugins while leaving others alone.
- [Audio and MIDI recorder](#) - An audio recorder captures audio from input and/or output channels to WAV files for post-production editing, processing, and mixing in a DAW. A MIDI Recorder captures MIDI input from all devices, placing each on its own track of a Type 1 Standard MIDI File that can be imported for post-production in a DAW.
- Works with any operating system-compliant audio interface.
- Predictive Loading reduces RAM and CPU demand in systems with limited resources.
- Automatic detection of new MIDI devices—connect and reconnect on the fly.
- Full MIDI remapping, transposition, layering, and keyboard splitting.
- Send program change and other messages to external MIDI devices automatically.
- VST Bus Layout support.
- [OSC support](#) for remote control via tablets and smartphones.
- Compatible (via OSC) with Max/MSP and Bidule for the development of complex interactive control algorithms.

- [GPScript](#) proprietary programming language for complex or custom needs.
- Gig file Templates.
- The [Global rackspace](#).
- Zoomable Wiring view.
- Non-linear scaling curves that allow advanced widget scaling and keyboard velocity scaling.
- The Probabilistic Sound Designer.
- An external API for third-party developers.
- Support for Cakewalk-style instrument definition files.
- Select and copy/paste/duplicate multiple blocks simultaneously.
- Export selections to a file and easily drag them into rackspaces.
- Save and load a preset for a plugin.
- Large numeric keypad for easier numeric editing on a touchscreen.
- Startup hints.

Please navigate through this manual to find more information about these bullet items.

Refer to the next chapter to find what's new in Gig Performer.

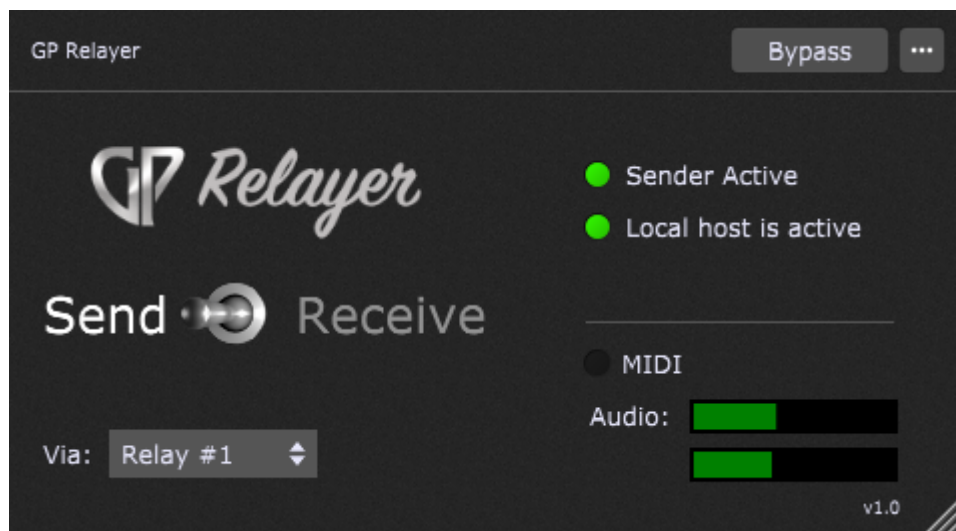
## 2 What is new in Gig Performer 5?

Gig Performer 5 introduces many exciting new features and improvements to existing ones. Gig Performer 5 can now send and receive MIDI and audio to and from your DAW. We have added an Envelope Follower plugin that you can use for such things as auto wah or automatic ducking when speaking. Our new MIDI Chord Maker plugin not only lets you map notes to chords but also lets you block notes. See below for more details on all the new plugins. We have added time stretching and pitch shifting to the Streaming Audio File Player. To help you understand more complex wiring setups, you can now arrange for wires to flash like meters to show you exactly from where (and to where) signals are going. Keep reading to find out more.

Some of the new additions are listed below. For the complete list of changes, please refer to our [Release Notes](#) page.

### 2.1 GP Relayer

GP Relayer is a new plugin in Gig Performer 5 that allows you to transport audio and MIDI effortlessly between Gig Performer instances, to other DAWs for easy recording, and from DAWs back to Gig Performer to utilize their proprietary sounds and effects.

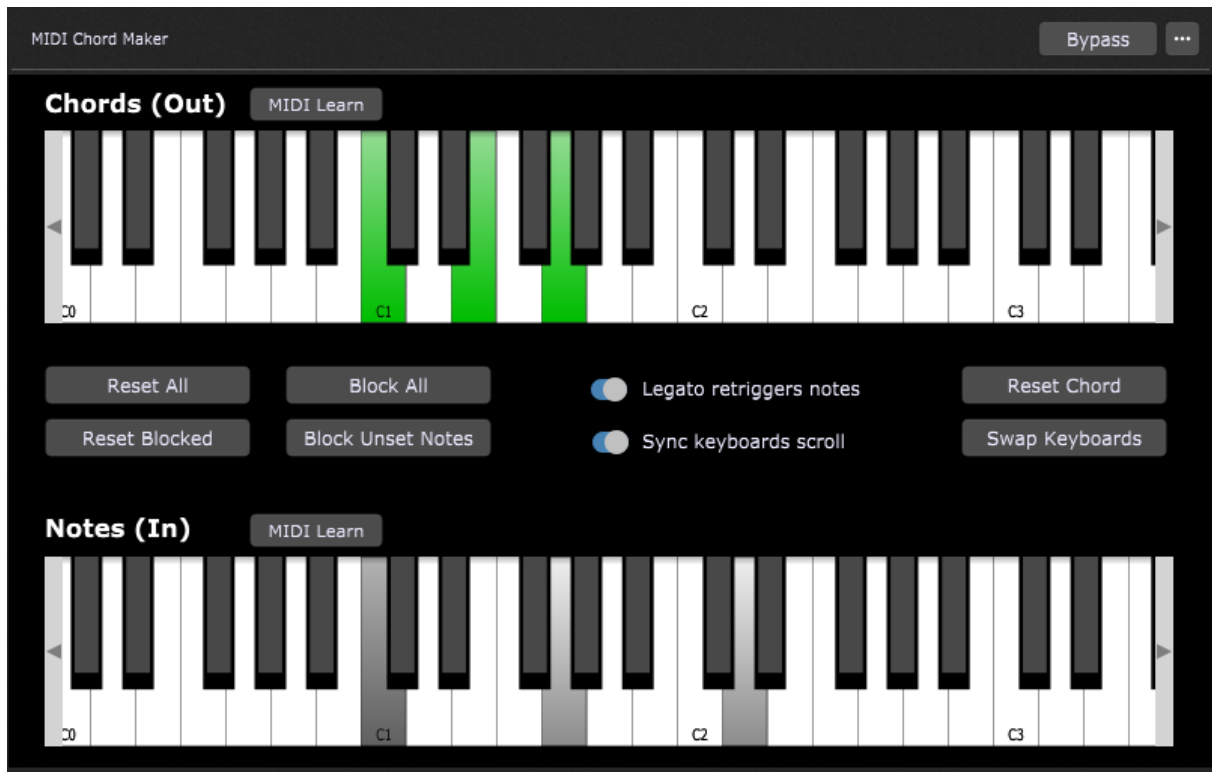


If you want to record Gig Performer's outputs directly to your DAW, you no longer need any third-party software.

GP Relayer allows up to 10 simultaneous relay connections. Refer to [this chapter](#) to learn more. Watch [this YouTube video](#) to see this feature in action.

## 2.2 MIDI Chord Maker

The MIDI Chord Maker is an advanced note mapper that not only allows you to trigger chords and block unwanted notes effortlessly but also features MIDI learn capabilities to speed up the workflow for mapping notes to chords.

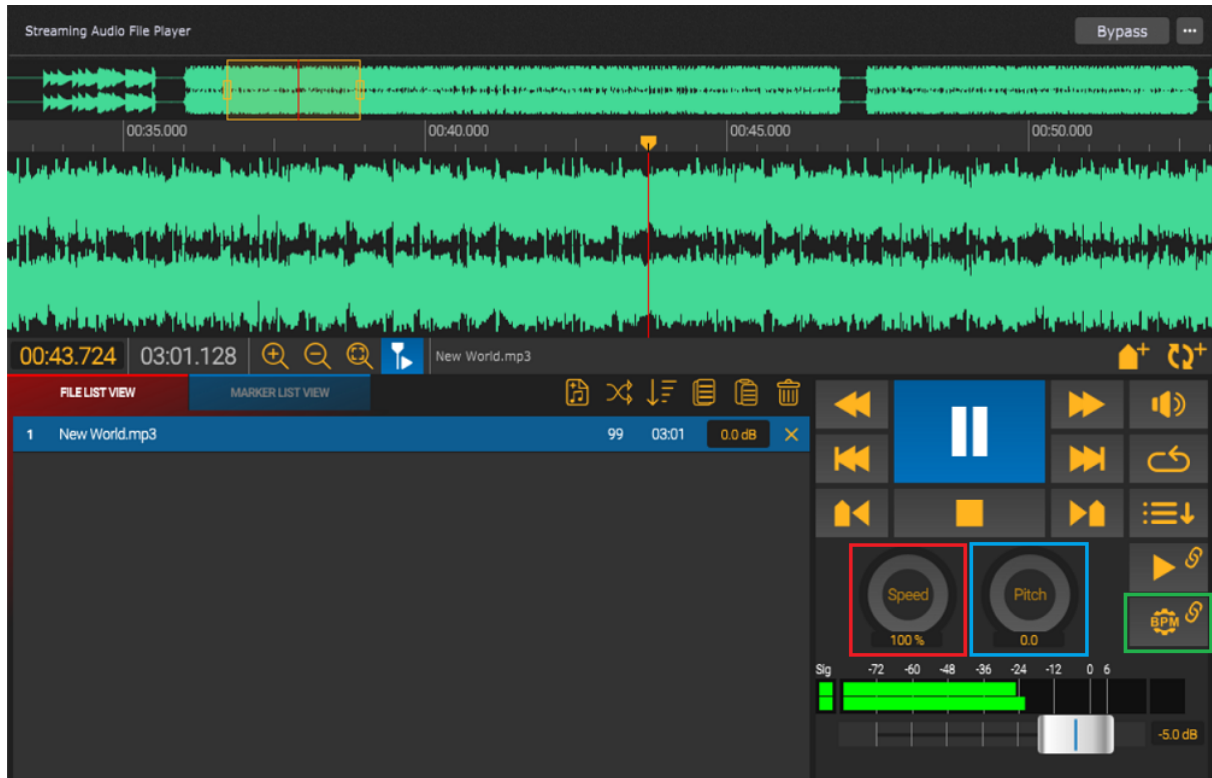


With the MIDI Chord Maker, activating multiple notes is straightforward and intuitive; click [this chapter](#) to learn more. To see this feature in action, check out [this YouTube video](#).

## 2.3 Enhanced Streaming Audio File Player

The Streaming Audio Player plugin now includes support for **time stretching** and **pitch shifting**.





The speed and the pitch of individual audio files can be easily adjusted by moving the knobs, indicated by the red and blue rectangles respectively. You can also sync the BPM of the track to the Global BPM value in Gig Performer (indicated by the green rectangle); the speed of the track will be automatically adjusted to match Gig Performer's tempo.

Click on [this chapter](#) to learn more and check out [this YouTube video](#) to see these features in action.

## 2.4 Envelope Follower

The Envelope Follower is a plugin that analyzes an incoming audio signal and produces MIDI CC messages with values proportional to the magnitude of that signal.

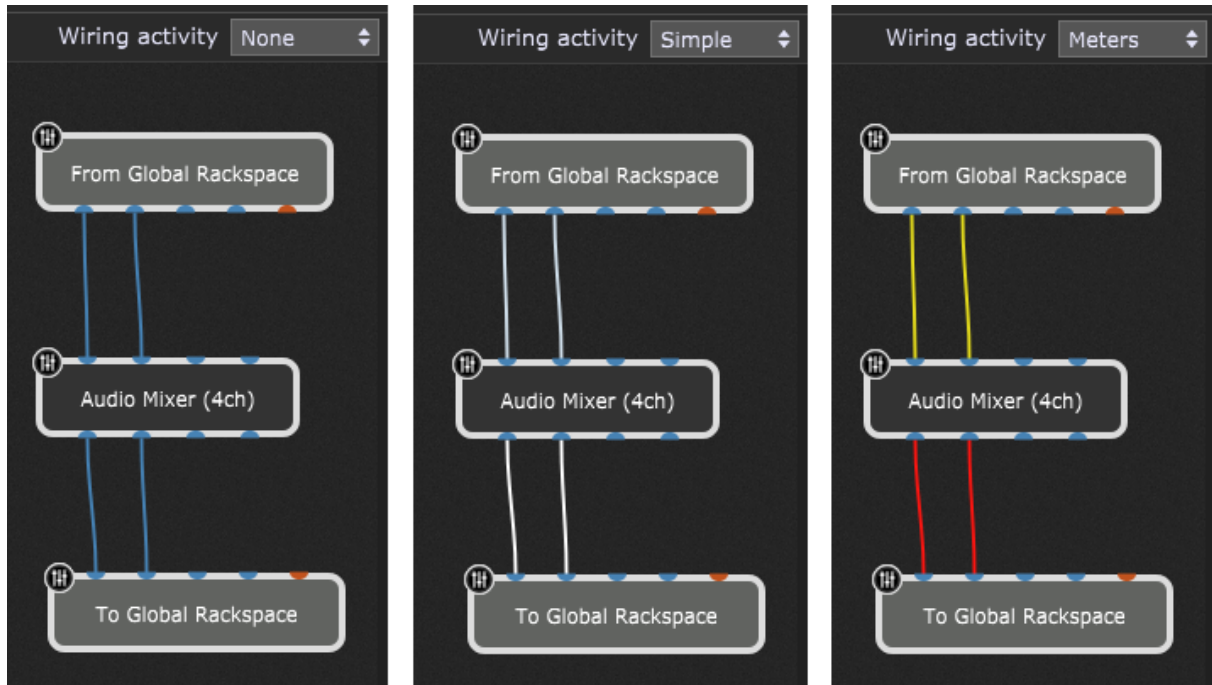
Uses for electric guitar are, for example, signal dependent Wah-wah or signal dependent high cut-off filtering, which would produce almost violin like effects. Another valuable use is for ducking audio while speaking. Read [this blog article](#) to see examples.



Click on [this chapter](#) to learn more about this plugin.

## 2.5 Audio and MIDI activity display

New to Gig Performer v5 is the Wiring activity feature that allows you to easily view the audio levels and MIDI activity in the Wiring view.



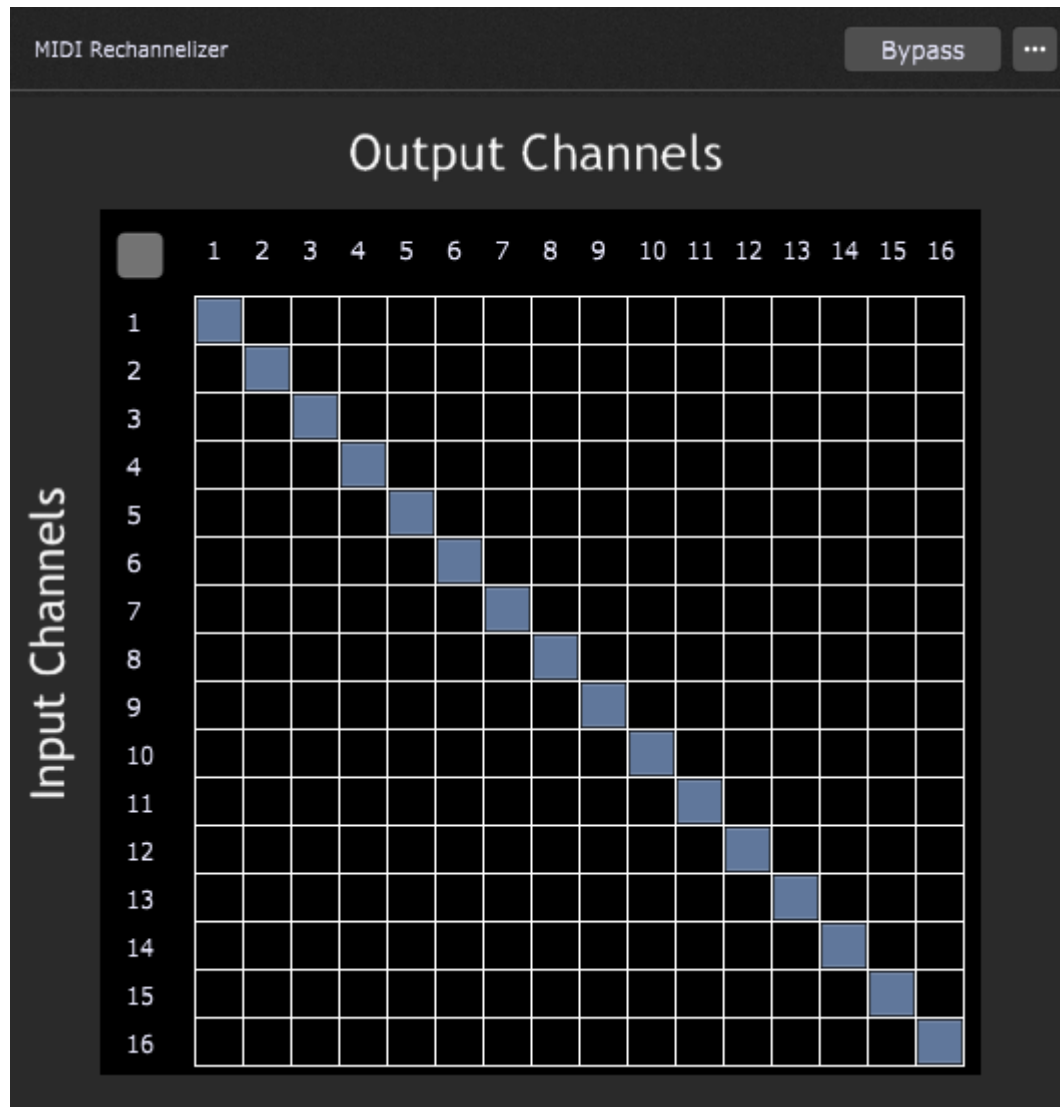
There are three modes of the Wiring activity:

- **None** - Turns off the feature.
- **Simple** - The color of audio wires changes from dark blue to lighter nuances or blue and MIDI wires light up to show MIDI activity.
- **Meters** - The color of audio wires changes from green to yellow to red and MIDI wires light up to show MIDI activity.

This feature allows you to easily understand your audio and MIDI and discover where your audio might be too high or troubleshoot your setup if you are not receiving audio or MIDI at some point.

## 2.6 MIDI Rechannelizer

The MIDI Rechannelizer allows you to simplify MIDI channel routing with a grid-based system, which is perfect for sending MIDI messages to external gear.

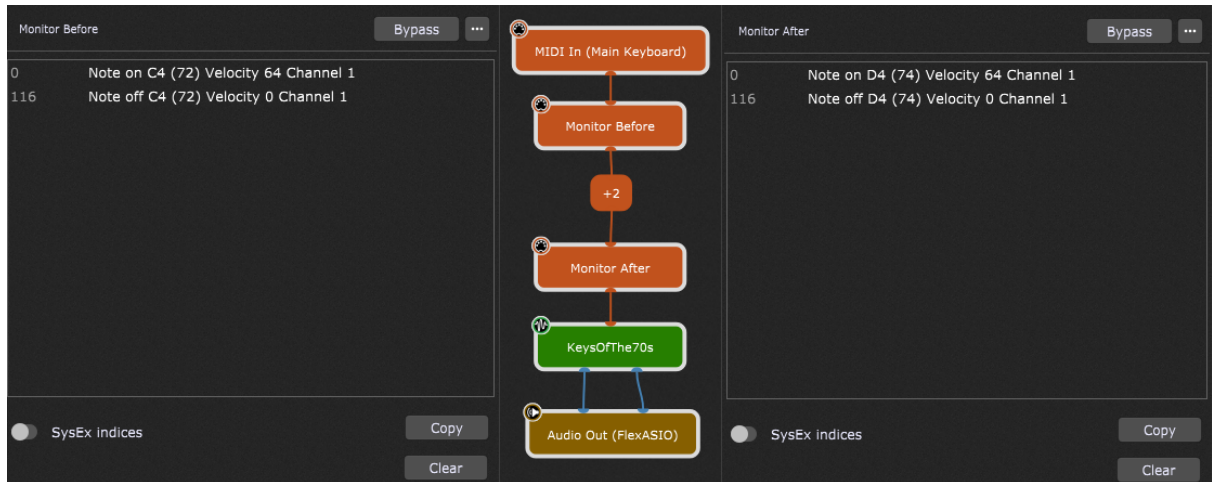


An intuitive grid-based interface simplifies the redirection of MIDI signals to and from any MIDI channel or multiple MIDI channels simultaneously. Ideal for both external gear integration and internal channel reconfiguration, the MIDI Rechannelizer enhances your ability to manipulate and streamline MIDI flows.

Learn more about this plugin in [this chapter](#) and see it in action in [this YouTube video](#).

## 2.7 MIDI Transposer

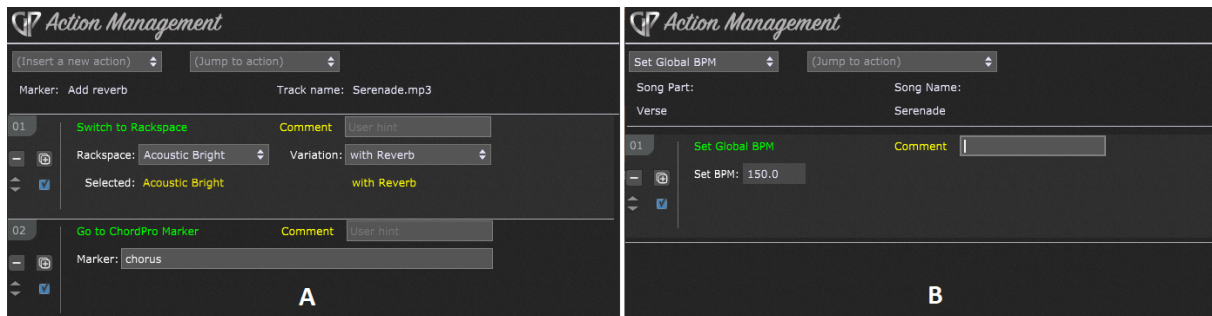
The **MIDI Transposer** is a simple plugin that allows you to transpose incoming note and poly aftertouch messages.



[Click here](#) to learn more.

## 2.8 Improved Automation

Gig Performer 5 brings new automation capabilities, with the ability to trigger actions from song parts.



The familiar action automation, previously available only through the Streaming Audio File Player markers **(A)** is now expanded and can be triggered from any song part **(B)**.

A new action "Metronome Click" is added to the list of available actions and allows you send a single metronome click. Refer to the [Action Management](#) chapter to learn more.

## 2.9 Plugin enhancements

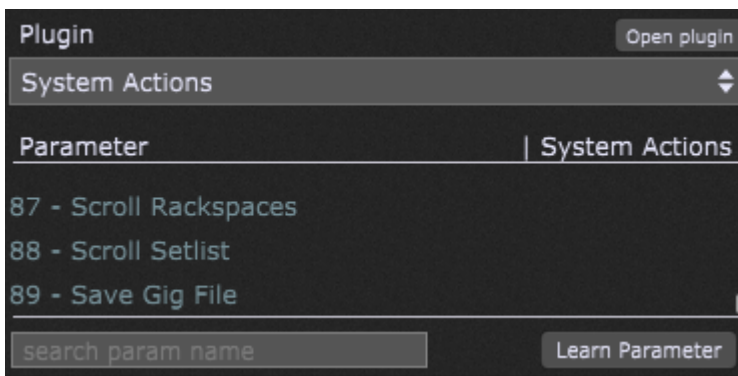
In this chapter are listed enhancements to Gig Performer plugins.

First, all **Global blocks** now support sending and receiving MIDI, as well as audio.



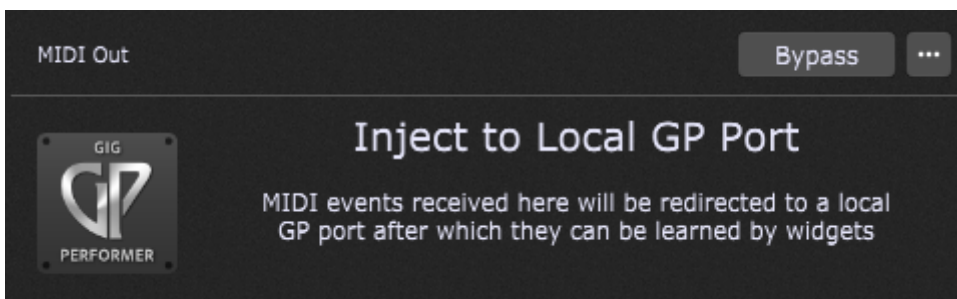
The **System Actions** block now includes new functionality:

- The Save Gig File parameter allows you to quickly save your gig file.
- You can now also scroll the Panels and Setlist views.



Note: To improve readability, spaces are added to the parameter names in the System Actions plugin.

The MIDI In (Local GP Port) plugin now has its MIDI Out counterpart – **The MIDI Out (Local GP Port)**.



This block is designed to receive messages from plugins such as the [MIDI File Player](#), [Envelope Follower](#), or other third-party plugins such as arpeggiators so that they can be mapped to widgets.

## 2.10 Other enhancements

Here is a short list of other notable enhancements. For the complete list of changes, please refer to our [Release Notes](#) page.

- The Global Rackspace button is now on the main panel, making it easier to access from any view (previously it was located above the rackspace area).
- The MIDI activity indicators in the main Gig Performer window are now a bit brighter.
- The Plugin Manager now includes the Widget Refresh column that allows you to explicitly refresh mapped widgets after loading a GP User Preset (rarely needed, for plugins that don't notify a host for their parameter changes).
- The categories for the internal plugins menu groups are now alphabetical.
- There is now a menu option to clear the recent files list.
- The Song script and Gig script editor positions are now properly remembered and restored.
- It is no longer necessary to hold the **Shift** key down to copy widgets from global rackspace to local rackspace. If the global widgets have parameters defined, they will be automatically configured.
- MIDI In blocks no longer send out Note Off information when changing variations if the transpose value hasn't changed.
- A simple Decent Sampler template is now included with the installer – it is in Documents/Gig Performer/Decent Sampler Templates on both platforms.
- The Auto Sampler editor plugin now uses the correct name for the Decent Sampler preset.
- The OSC subsystem now support `/handle/LoadGPPreset` so that you can load a GPUUserPreset into any plugin via OSC.

## GPScript improvements

- Enhanced the template for demonstrating parameter variables.
- Added functions to scroll rackspace and songs and to save the gig file.
- A widget can now be appended to an array using the ← operator.
- Added SaveGPPreset system function.
- Added system functions to send widgets behind or in front of another widget.
- Added system function to set the bounds of one widget to that of another widget (which makes it easy to position one widget on top of another).
- Added functions for converting between linear and dB values.

## SDK improvements

- Added functions to get the song key and song tempo.
- Removed restriction of having to be in Setlist mode to query properties about songs.
- Hardening to getSongKey and getSongTempo functions.

## 2.11 GPScript - Breaking Change

The return value of a function can no longer be discarded and the result must either be tested or saved to a variable. For example:

```
Function foo(i : integer) returns boolean
```

You can no longer simply write

```
foo(42)
```

Instead you must either declare a boolean value into which the result is stored or test the value.

Examples:

```
bool someResult;
```



```
someResult = foo(42)
```

or

```
if foo(42)  
  then ....  
End
```

### 3 Getting started

This chapter will provide useful information before you start actively using Gig Performer.

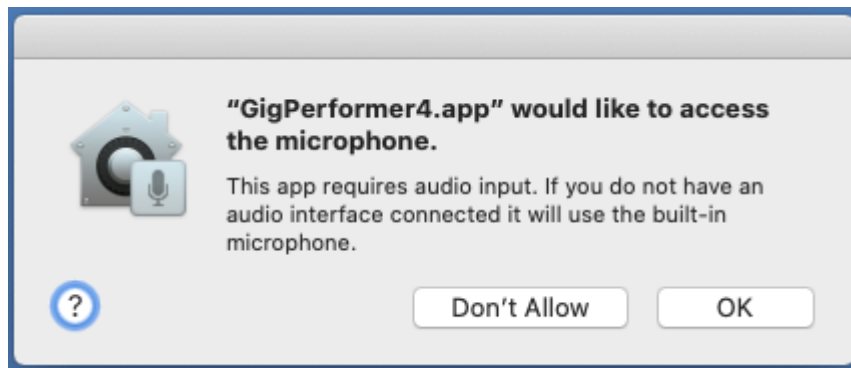
Further, check out our:

- Special blog article dedicated to new users [Getting started with Gig Performer](#).
- Getting Started YouTube video [here](#).

#### 3.1 Before you start

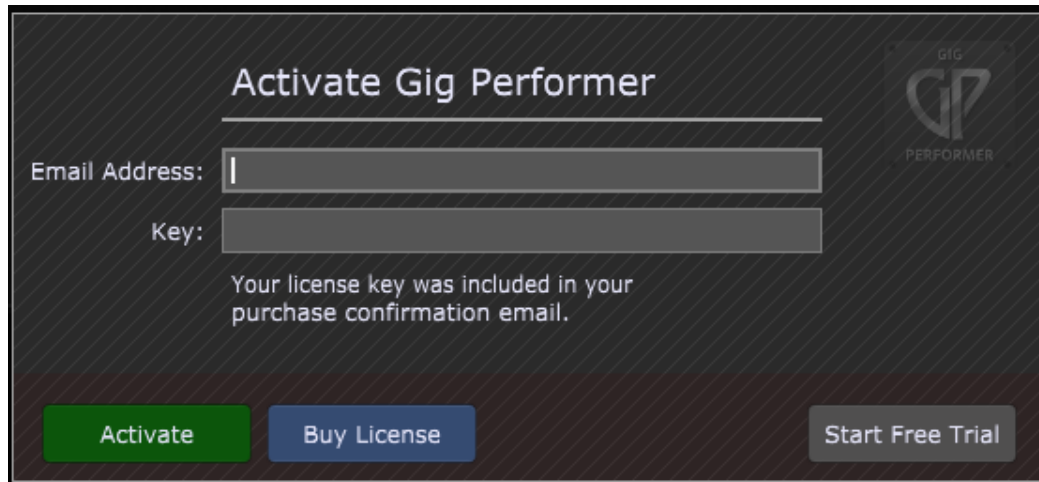
Before launching Gig Performer, **ensure that all audio and MIDI interfaces connected to your system are powered on**. If no external audio interface is connected, Gig Performer will try to use your computer's built-in input and output if available. If you have more than one interface connected, you can choose among them after the application launches. Note that you can use different interfaces for input and output if desired (refer to the [Audio I/O](#) section for more information).

**Note:** if you are running Gig Performer on macOS 10.14 or above, you will be prompted to allow access to the "microphone":



You must allow this as otherwise Gig Performer will be unable to receive the audio from your guitar, microphone or other audio device that you plug into any of the inputs of your audio interface.

When Gig Performer is first launched, the following dialog will appear:



If you have purchased a license, enter your email address, and type in or paste the activation key (your key is emailed to you when you buy a license; make sure your anti-spam blocker didn't prevent the confirmation email with your license key from being received). Then click on the **Activate** button to activate your copy of Gig Performer.

**Note:** you need an Internet connection to activate Gig Performer. After you activate Gig Performer, you can work offline.

**Important note:** some operating system updates can cause the computer to "look" like it is a new or different computer, thereby requiring applications and some plugins to be reactivated. We recommend that you turn off automatic operating system updates. [Check out this blog article](#) to learn why.

If you have not yet purchased a license, click on the **Start Free Trial** button to access a fully operational trial version of Gig Performer for 14 days (the dialog will also show you how many days remain for this trial).

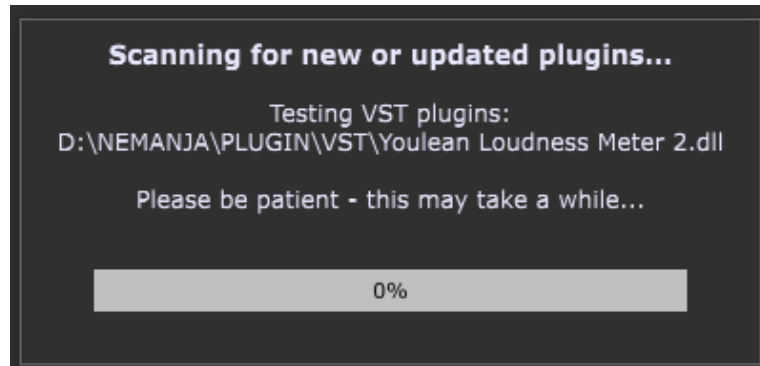
You can purchase a license key anytime by clicking the **Buy License** button.

**Note:** when Gig Performer is in trial mode:

- there will be random momentary dropouts in the audio,
- the screen will flash red,
- All relevant options in the [Auto Sampler Generator](#) are locked and cannot be changed.

Before using Gig Performer, please make sure you agree to the license agreement, a copy of which is included (see Appendix -> [License agreement](#)).

Immediately after launching, Gig Performer will scan your system looking for plugins and the following dialog will appear:



When running Gig Performer on macOS systems, it will scan for VST, VST3 (Virtual Studio Technology) and AU (Audio Unit) plugins; when running on Windows systems, it will look for all VST and VST3 plugins. Note that only 64-bit plugins are supported.

**Note:** If you are a Windows user, you may have to tell Gig Performer what folders to search, since there is no single location where all plugins are guaranteed to be installed.

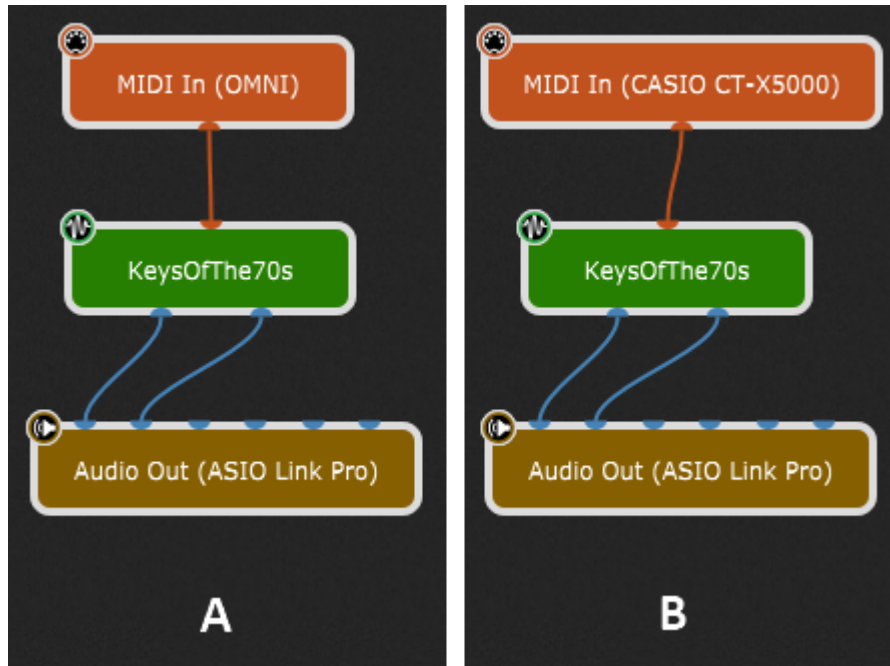
To do so, launch Gig Performer, go to the Windows menu, then select [Plugin Manager](#). In the resultant window, click **Manage**, then choose the **Set VST folders for scanning** option. We suggest that you begin by having Gig Performer search in the folder C:\Program Files\VstPlugins. If additional plugins are stored in other folder locations (as some plugin and DAW manufacturers automatically create), the Plugin Manager allows you to add other folders to scan easily.

This scanning process can take some time the first time you use Gig Performer but will be nearly instantaneous on subsequent launches unless you have installed or updated any plugins since the last usage (if you like, you can disable the automatic scan completely in the [General options](#) dialog).

**Note:** Gig Performer comes with several [specialized plugins](#) to handle basic MIDI and audio operations. Some third-party plugin vendors have kindly provided us with versions of some of their commercial plugins which are used in some of [our templates](#) to make it easier for new users to work with Gig Performer if they do not have their own plugins yet.

## 3.2 Basic setup for keyboardists

To get started as a keyboardist, first insert the desired synth plugin and connect the existing [MIDI In \(OMNI\)](#) block to it (check [this chapter](#) to learn more):



Incoming MIDI messages from any of your keyboards will now be sent to the synth plugin. Complete the signal chain by connecting the synth plugin block to your audio interface outputs. This very basic setup is displayed in image **(A)**. At this point, you should be able to play your connected keyboard(s) and hear the result.

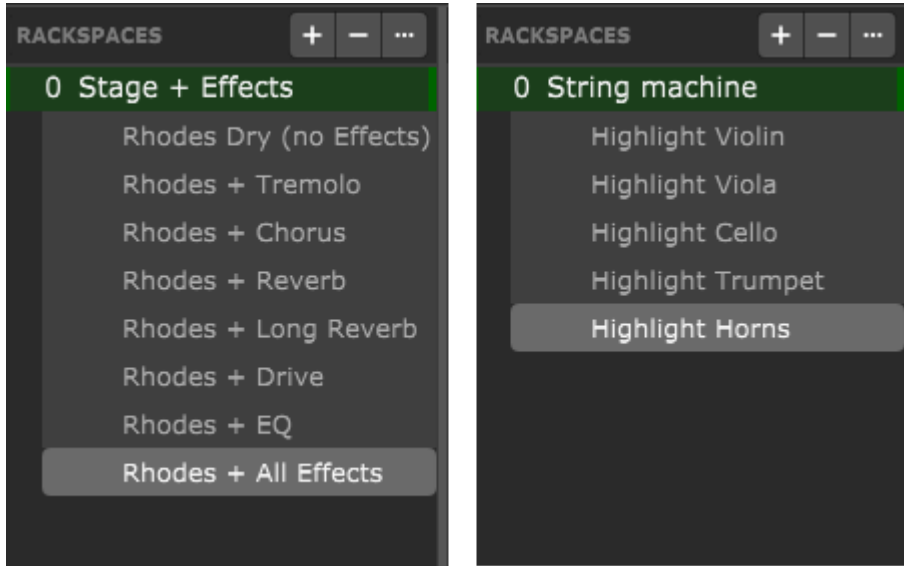
If you use more than one keyboard and want each one to control different plugins, you must replace the MIDI In (OMNI) block with MIDI In blocks associated with each keyboard (MIDI device). To do this, you can either insert a new MIDI In block or [right-click](#) on the MIDI In (OMNI) block and replace it with the desired one, as indicated with **(B)**.

When you have finished adding your plugins, switch to the [Panels view](#), and enter Edit mode. Add panels and widgets for the plugin parameters you mostly use or want to control either from the panel or from your keyboard or other MIDI control surface. Here is an example:

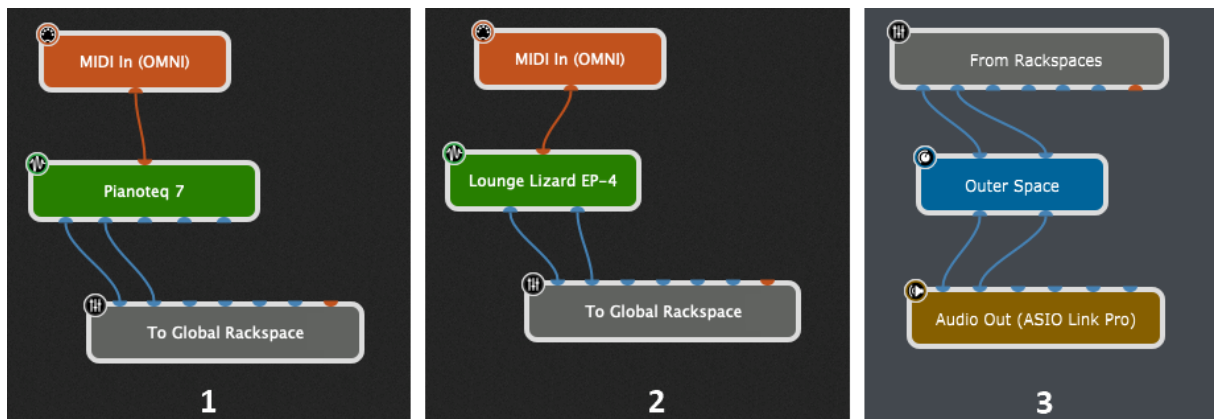


Refer to the [Widgets](#) chapter to learn more. Make sure also to check our [built-in templates](#) and Community-driven shared rackspaces (click Help -> Community -> Community Shared Gigs, Rackspaces and More) to access nicely designed, community-created rackspace and gig files that you can use directly or as templates for your setups.

Depending on the features of your plugins (i.e. are they multi-timbral) you can organize your rackspaces and variations in many different ways, for example:



Gig Performer includes the Global rackspace where you can place your common effects, such as reverb, delay, or other plugins that you might want to be present all the time (i.e. an Audio File Player or a MIDI File Player). The screenshot below shows the Wiring views of two rackspaces and the Global rackspace. One rackspace contains an acoustic piano **(1)** and the other contains an electric piano **(2)**. A delay is added to both of these sounds in the Global rackspace **(3)**:



So rather than inserting a delay plugin in both rackspace, you can simply insert it in the Global rackspace.

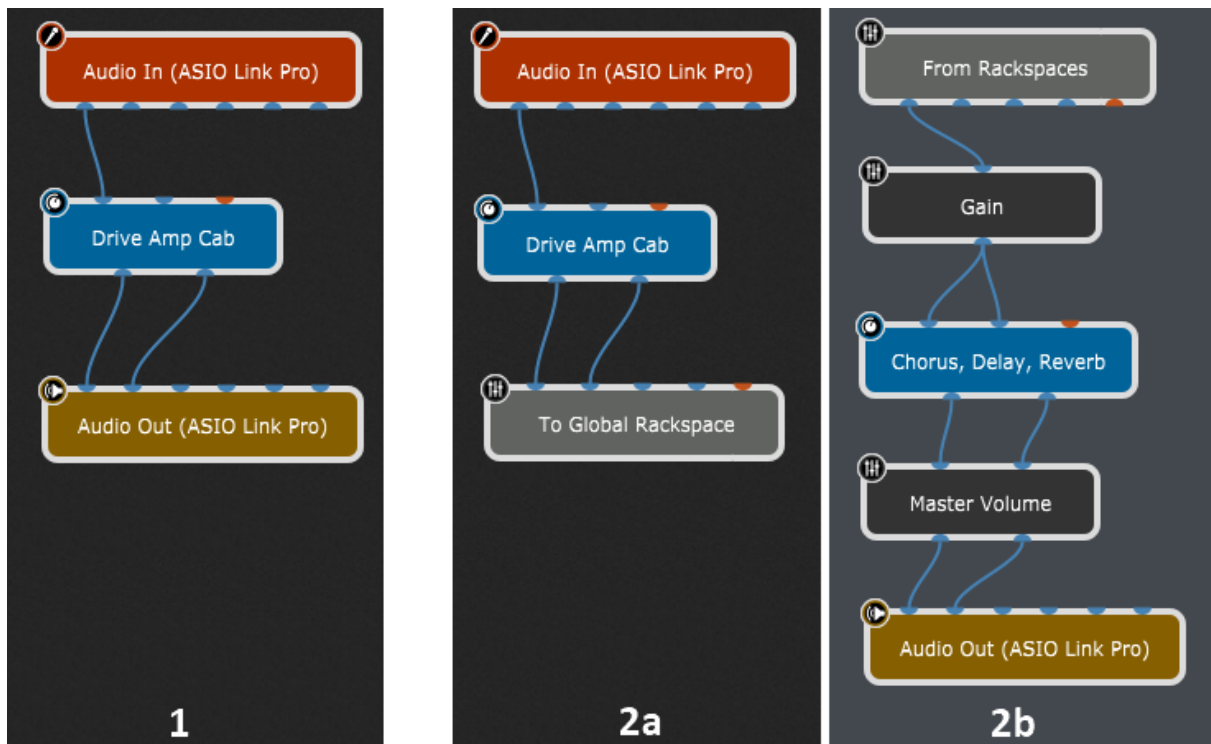
Check the [Global Rackspace](#) chapter to learn more about the Global rackspace.

As a keyboardist you'll appreciate that each rackspace supports a mechanism that allows your sounds to stay on until you release all the keys that you were holding down (or the sustain pedal if it was down), even if you switch to another rackspace with different sounds for the next part of your set. This mechanism is called Patch Persist (check the *MIDI Patch Persist* section in the [Rackspace properties](#)).

When this feature is disabled, when you hold a note down and then switch to the other rackspace, the note you're holding down stops immediately.

### 3.3 Basic setup for guitarists

To get started as a guitarist, first connect your audio interface inputs to your effect and amp modeling plugins (check [this chapter](#) to learn more):



Complete the signal chain by connecting these plugins to your audio interface outputs. The very basic setup is displayed in image [1]. At this point, you should be able to play your guitar and hear the result.

You can mix and match plugins together just as if they were physical devices. There are no complicated AUX buses, sends or inserts. Simply connect things as if you were using wires.

If you want to parallel process your signal and, for example, run it through two different amps you can simply connect your input signal to two amp plugins, then sum these together with our [Audio Mixer](#) plugin.

Gig Performer includes the Global rackspace where you can place common effects, such as Chorus, Delay and Reverb. Image **[2a]** displays a rackspace that routes the audio signal to the Global rackspace, and **[2b]** displays a connection where the audio signal is coming from the Global rackspace, does some further processing and then completes the signal chain by sending the signal to your audio interface outputs.

Check the [Global Rackspace](#) chapter to learn more about the Global rackspace.

When you finish adding your plugins, switch to the [Panels view](#), and enter Edit mode. Add panels and widgets for plugin parameters you mostly use or want to control via MIDI or [OSC](#). Here is an example:



Refer to the [Widgets](#) chapter to learn more. Make sure also to check our [built-in templates](#) and Community-driven shared rackspaces (click Help -> Community -> Community Shared Gigs, Rackspaces and More) to access nicely designed, community-created rackspace and gig files that you can use directly or as templates for your setups.

We recommend that you organize your rackspaces based on the common sounds you use, like Clean, Crunchy, Distorted, etc. This is not a rule, however, and you're free to organize things the way it works for you and your playing style.



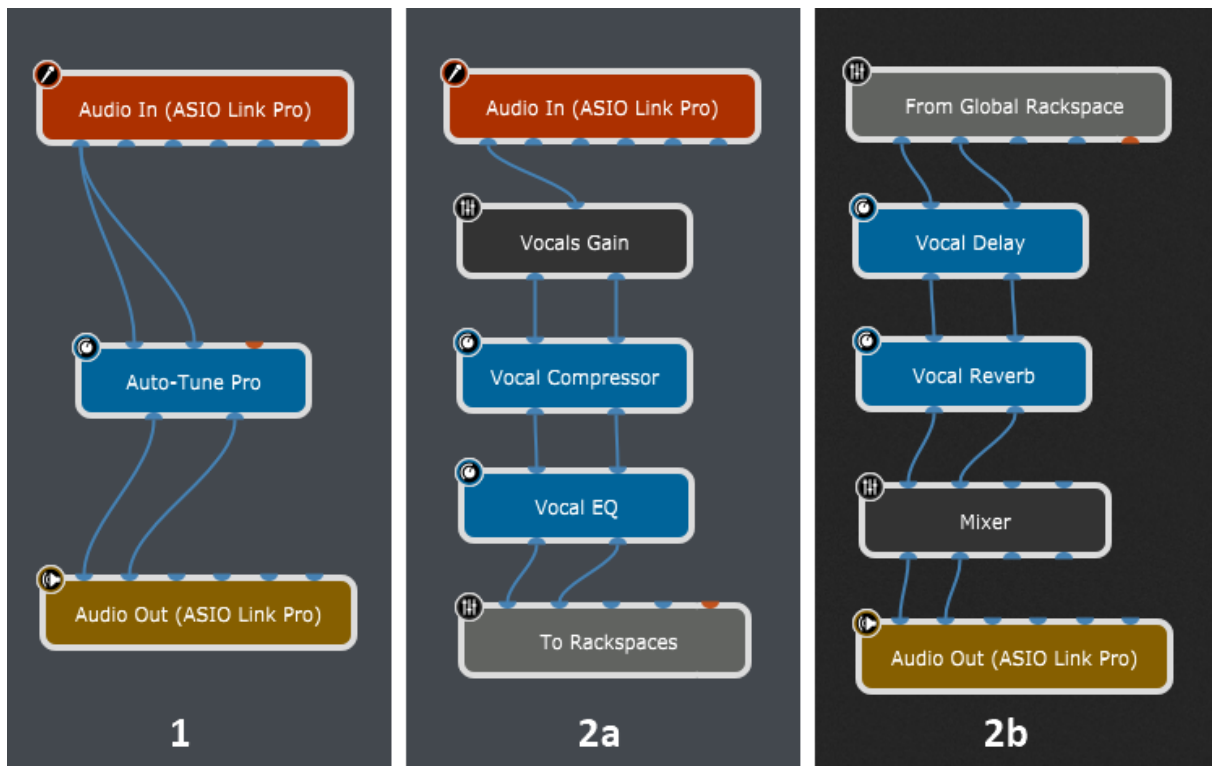
As a guitarist you'll appreciate that each rackspace has fine-grain control over its "silencing periods". This means you control how long each rackspace has to silence its input/output as you switch to another rackspace. This provides an incredibly smooth transition between your sounds (check the *Tail Length* section in the [Rackspace properties](#)).

Check out the blog article [Top 10 reasons why you should move to a Gig Performer based guitar rig](#).

### 3.4 Basic setup for vocalists, drummers, other musicians and FOH engineers

The last two chapters cover basic setups for keyboardists and guitarists, but Gig Performer can be used by any artist who performs on any instrument live or in the studio using their specific plugin setups.

For example, vocalists can easily add compression, EQ and other desired effects to their vocals.



Vocalists might want to take advantage of the [Global Rackspace](#) and add their effects there if they don't change them often, as shown in **(1)**. Of course, feel free to make

arrangements any way you like; for example, you can use the Global rackspace for plugins that you want to be always present in your setup (like shown in **2a**) and take advantage of [rackspace variations](#) to apply different amounts of reverb and delay (like shown in **2b**). Check out the YouTube video [How to create a vocal effects chain in Gig Performer](#) to see an example with free plugins.

Similarly, drummers can mix and match their favorite drums and cymbals from different drum plugins and create drum kits. Drummers can organize drum kits as rackspaces and switch them instantly.

Front-of-house sound engineers will appreciate Gig Performer's [OSC implementation](#). Sophisticated plugin chains can be reviewed and adjusted remotely. Your entire FOH setup can consist of a rackmount mixer and Gig Performer, all of which can be controlled remotely from a tablet, such as an iPad.

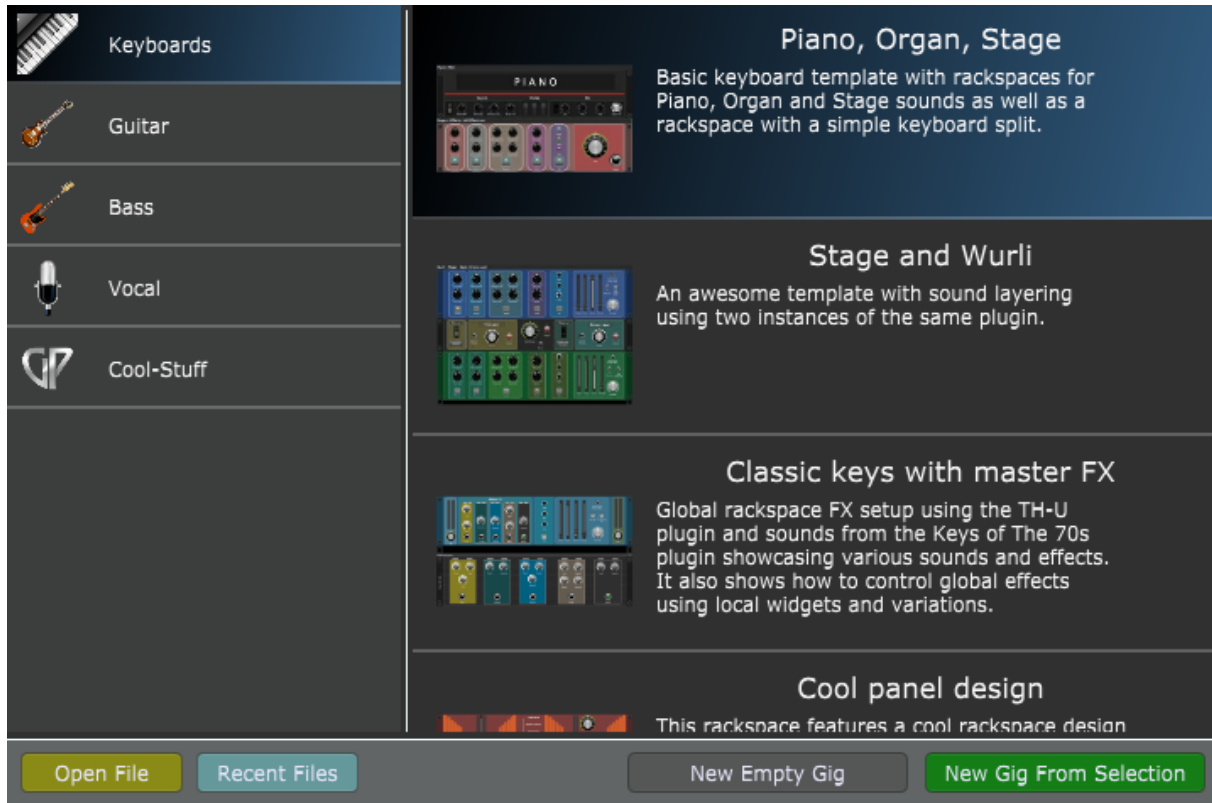
Using Gig Performer [instances](#), every band member can use different plugin setups on a single computer (provided that a computer is powerful enough to host these setups).

Check out our Community-driven shared rackspaces (click Help -> Community -> Community Shared Gigs, Rackspaces and More) to access nicely designed, community-created rackspace and gig files that you can use directly or as templates for your setups.

### 3.5 Built-in templates

A template is simply a predefined gig file that is nicely laid out so users can easily reuse it. You can use the supplied plugins or replace them with other user-provided plugins and then reassign the widgets. A template can contain one or more rackspaces with variations.

When you first start Gig Performer the New Gig dialog will appear:



**Note:** the contents of this dialog may differ from the screenshot above, as new templates may be included in future versions of Gig Performer.

This dialog will also appear every time you start Gig Performer if you turn off the *Reload the last gig on startup* toggle button in the [General Options](#). You can invoke this dialog any time by clicking on File -> New -> New Gig from Template...

To create a gig from the template, simply select a category, such as Keyboards (as indicated in the screenshot above) and double-click the desired template (alternatively, select the desired template and click on the *New Gig From Selection* button).

Depending on your selection, the fully functional gig file will open with nicely laid out panels with widgets:



You can use templates to quickly try out Gig Performer features or get ideas for your own gig design.

**Note:** the screenshot above is used for demonstration purposes only and might be changed in the future as templates might change.

You can also browse the Community-driven compilation of gig files and rackspaces; click *Help -> Community -> Community Shared Gigs, Rackspaces and More* to access this section of our [Community](#).

### 3.6 How to update Gig Performer

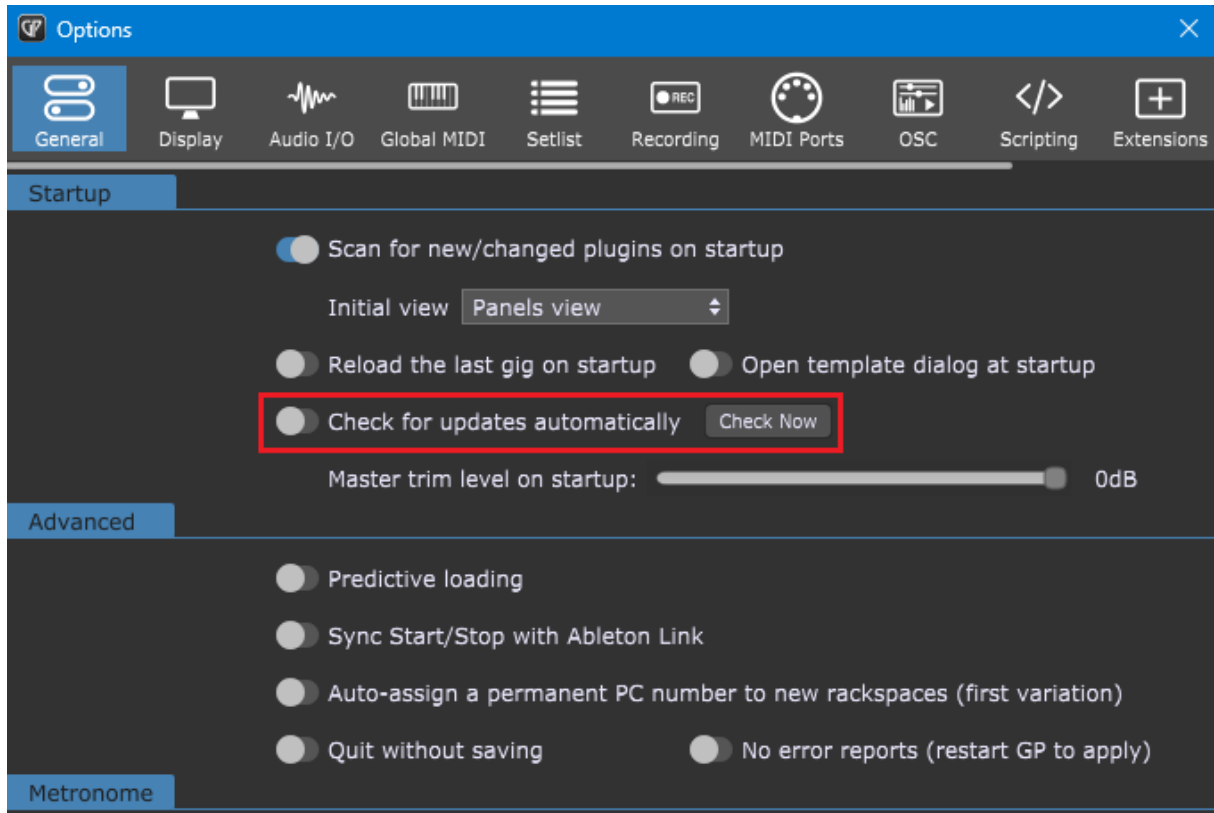
First, we strongly recommend that you don't update anything within at least **two weeks of an upcoming show**. That way, if a problem is discovered, there's time to deal with it without stress.

If you're about to go on tour, dealing with potential issues like this are not how you want to spend your time. Presumably your touring setup, including your existing version of Gig Performer, is working fine so leave everything alone and wait until your show or tour is over.

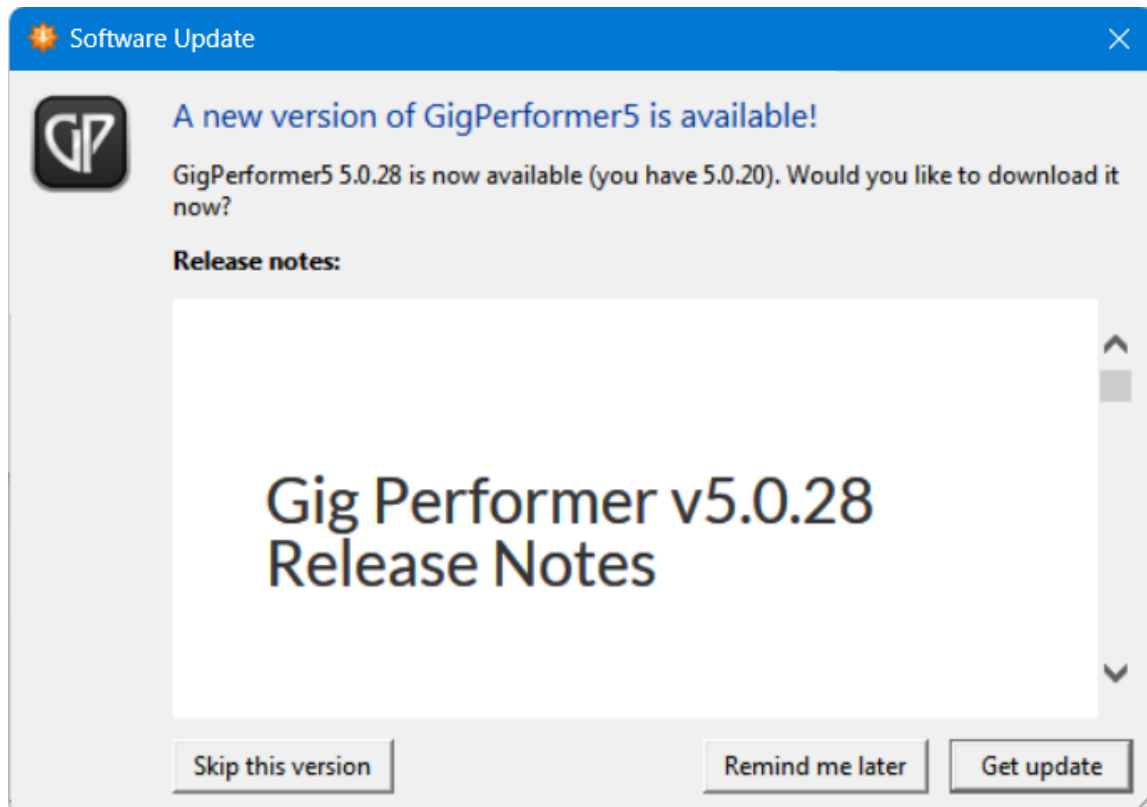
Here are two blog articles with the detailed explanation:

- [Be careful with updates](#)
- [Best practices for a rock solid software based setup](#)

To check whether there are new updates available for Gig Performer, go to [General Options](#) and then click the **Check Now** button.



If there is an update available, a new dialog will pop up:



Click the **Get update** button to go to the download page.

When you download the installer for your platform, install Gig Performer as usual.

Alternatively, you can go to the Gig Performer [Downloads page](#), download the installer (as if you were a trial user) and install Gig Performer. Your license will be automatically recognized.

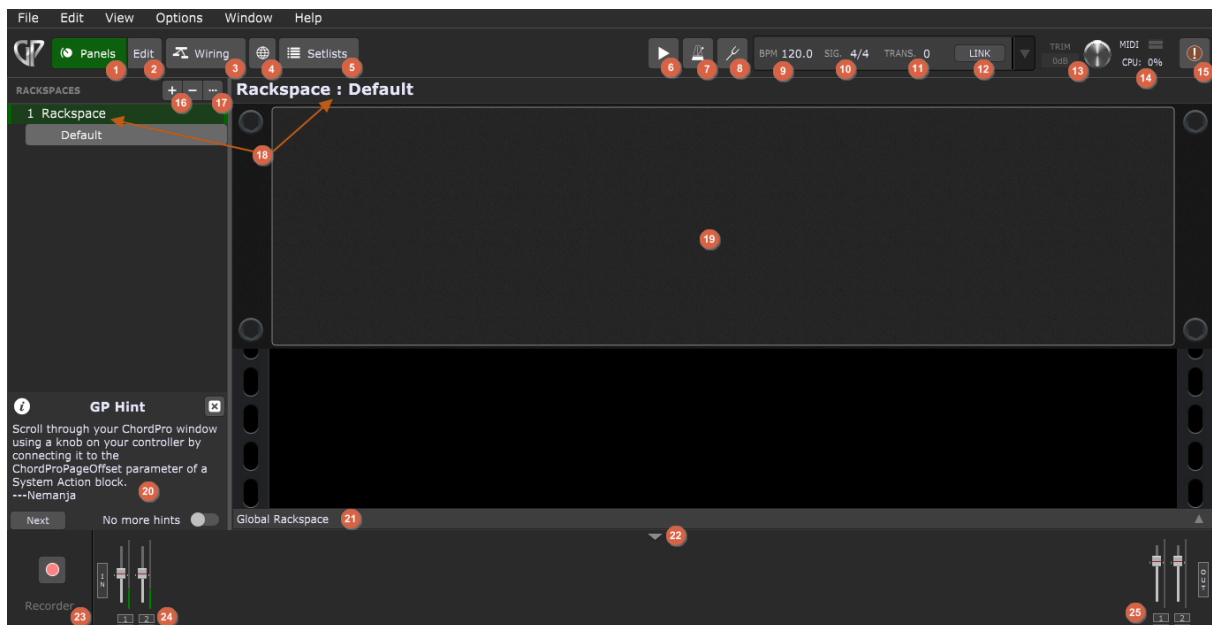
Note: if you use an older version of Gig Performer, please go to the [Store page](#) to check your update price.

## 4 Gig Performer user interface and concepts

In this chapter you will learn more about the Gig Performer user interface, different views, and important concepts such as rackspace, variations, panels, widgets, the Rig Manager and other key components.

### 4.1 The main Gig Performer window

When you start Gig Performer and the **New Gig** window shows up, you can create a new empty gig, or choose a template to use as your starting point. The following screenshot shows the main Gig Performer window after a new empty gig has been chosen:



1. **Panels view** - click here to show the [Panels view](#).
2. **Edit button** - in the Panels view, click on this button to enter Edit mode, where you can edit panels and widgets.
3. **Wiring view** - click here to show the [Wiring view](#).
4. **Globe icon** - click on this icon to show or hide the [Global Rackspace](#).
5. **Setlists** - click here to show the [Setlist view](#).
6. **Play/Stop button** (also referred to as the **Global playhead**) - click to start playback, click again to stop it. When playing, all plugins that support playback control (for

example, sample, loop, or beat playback plugins) play back. When disengaged ("stop"), they are all put in a stopped state.

7. **Audio metronome on/off** - click to turn on the audio metronome, click again to turn it off. See [Metronome](#) for more information.
8. **Tuner icon** - click here to toggle the [Tuner view](#) (the icon turns red).
9. **Tempo display** - shows the current tempo being used by all connected plugins (for example, sample/loop/beat-playback plugins, as well as those generating echoes, delays and reverbs). You can of course change the tempo any time. The tempo can also be set automatically by song and song part tempos. For more information see [Tempo and Time Signature](#). The minimum tempo value is 20 BPM and the maximum tempo value is 640 BPM.
10. **Time signature display** - shows the current time signature being used by all connected plugins (for example, sample/loop/beat-playback plugins, as well as those generating echoes, delays, and reverbs). You can change this value and enter any time signature, up to 32/64.
11. **Transpose** - enter the Global transpose value that will be applied to all [MIDI In](#) blocks by clicking on the field and entering a number of semitones directly, or by clicking on TRANSPOSE and using the increment (+) and decrement (-) buttons to set the number of semitones. Note: individual blocks can optionally ignore the global transpose value, something that is useful if a MIDI In block is being used for percussion or sound effects.
12. **Link button** - click this button to activate or deactivate [Ableton Link](#).
13. **Global output volume (Trim) knob** - use this to temporarily change the overall volume of your gig. An increase in volume is highlighted in red and a decrease is highlighted in blue. Double-click on the knob to reset its value to 0dB. The Trim knob can be remotely controlled; map it to a MIDI controller in the Global MIDI Assignments section of the [Global MIDI options](#) dialog. The initial value when Gig Performer first starts can be set in the [General options](#).
14. This section contains:
  - a. **MIDI activity indicator** - flickering here indicates the presence of MIDI data. The top bar lights green when incoming MIDI messages are received, while the bottom bar flickers when MIDI messages are being transmitted (via a MIDI Out block). If the "Sync tempo with external MIDI clock" option is selected for the current rackspace (see [Rackspace Properties](#)), a thin yellow bar above the two



flickers to indicate the presence of incoming MIDI Clock data and also serves as a visual tempo indicator. Open the [Global MIDI Monitor](#) window to see detailed information about the MIDI messages being received.

- b. **CPU usage indicator** - shows the percentage amount of CPU overhead currently being used by Gig Performer for audio processing. Gig Performer itself is responsible for only a small percentage; most of the CPU overhead is taken up by the plugins currently loaded in memory. It is normal for this value to change in real-time as incoming MIDI data is received and as widgets are moved.
15. **Panic button** - click this to issue "turn all notes off" messages on all MIDI channels, along with a Sustain OFF message and a Center PitchBend message, thus stopping stuck notes. Double-click this button to also reset the audio engine.
16. **New rackspace/variation ("+")** and **Delete selected rackspace/variation ("-")** buttons - click "+" to add a new rackspace or variation; click "-" to delete the selected rackspace or variation.
17. **More... button** - click on the button with the ellipsis (three dots) to drop down a [menu of operations](#) related to rackspaces, variations and widgets.
18. **Browser** (indicated by the left arrow) - in Panels and Wiring views, the browser shows a list of all rackspaces and variations that have been created. The current rackspace is outlined in green. The right arrow indicates the display of the current rackspace and variation. Click on a rackspace in the list to select it. Double-click on a rackspace in the list to open the [Rackspace Properties](#) dialog. Drag rackspaces up and down in the list to reorder them. The browser looks different in Setlist view, though the rackspace/variations list is still displayed. For more details, see [Setlist view](#).
19. **Panel surface** - in the Panels and Setlist views, the purpose of this surface is to display widgets previously dragged onto this surface in the Edit mode.
20. **GP Hint** - the hints database contains various tips and tricks about Gig Performer, many submitted by [Community](#) members. By default, a new hint will appear every time you start Gig Performer though of course you can turn this off. Show hints at any time by clicking on the Help menu -> Show a Gig Performer Hint.
21. **Global Rackspace toggle** - click on this toggle button to show/hide the [Global rackspace](#) controls.
22. **Collapse/Expand bottom controls toggle** - click on this arrow to collapse or expand the bottom controls, i.e. faders for all audio input and output channels.

23. **Recorder button** - click this to open the Recording Options dialog. For more details, see [Recording your performance](#).
24. **Interface input levels** - a set of meters and sliders that allow you to view and adjust the levels of your audio interface's various inputs. Flickering here indicates the presence of input signal. Double-click a fader to set it to 0 dB.
25. **Interface output levels** - a set of meters and sliders that allow you to view and adjust the levels of your audio interface's various outputs. Double-click a fader to set it to 0 dB.

**Note:** Gig Performer displays up to 16 input or output interface channels at a time. If your interface has more than 16 I/O connections, the remaining inputs and outputs are accessed by clicking the Next or Previous button to scroll the display.

## 4.2 Rackspaces and Variations

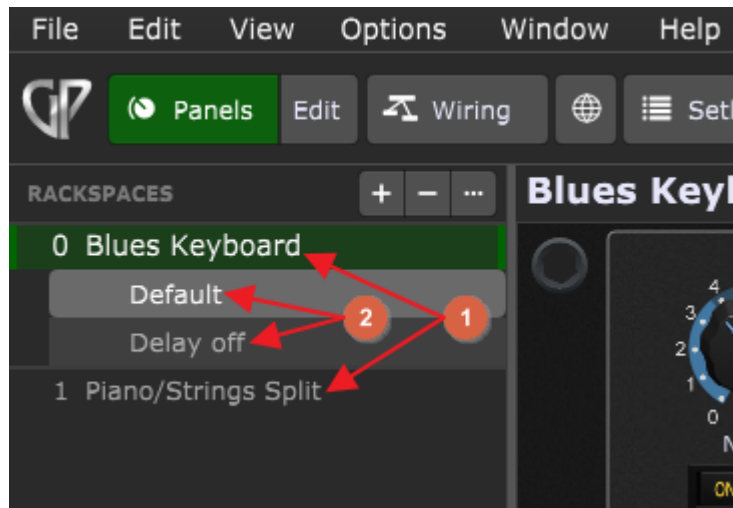
A **rackspace** is a collection of interconnected [plugins](#), along with one or more panels containing [widgets](#) to control plugin parameters in real time. A rackspace can be pretty much anything you like: it can represent a single sound or all the elements you need for an entire song - or just one part of a song. You can instantly switch from one rackspace to another, either from your laptop, with up/down pedals, or via MIDI program change messages. Sounds will change without any glitching and sufficiently quickly that you can switch rackspaces even in the middle of a bar.

A **variation** is a rackspace preset containing the same plugins and interconnections, but with different widget settings. For example, you can create a rackspace to add phasing to a piano sound, and then add several variations, each with different degrees of phase intensity. Since variations actually store widget settings, every parameter you want stored in a variation must be assigned to a widget on a rack panel. As with rackspaces, you can switch from one variation to another instantaneously and with no interruption of sound either from your laptop, with up/down pedals, or via MIDI program change messages.

Variations allow you to leverage the plugins in a single rackspace to create many different sounds without having to duplicate the rackspace for each sound. It's also a great way to save on memory. For example, you could have multiple virtual instrument plugins feeding a mixer plugin, and then mute different plugins, or have a different mix of them, in each variation. Or you could connect multiple effects (i.e. phasing, flanging, tremolo, reverb, or echo) and have different ones enabled or disabled in each variation. Rather than creating a rackspace for an instrument with phasing, and another for the same instrument with

echo, the same thing is accomplished by two variations of a single rackspace containing both phaser and echo plugins.

In the picture below are depicted rackspaces **(1)** and variations **(2)**:



**Note:** see also the Global rackspace, a special rackspace that allows audio plugins to be shared and used by regular rackspaces. Find out more in the [Global Rackspace](#) chapter.

Assuming that your keyboard is connected to Gig Performer and you are able to play sounds, you can switch from one rackspace to another by sending a standard MIDI Program Change message from your MIDI keyboard to Gig Performer. By default, each rackspace/variation is associated with a program change message, starting at 0. For other ways to select specific rackspaces and variations and for more details, see this in-depth blog article [How to change sounds and control plugins from your MIDI controller](#).

**Technical note:** Audio processing is disabled in non-active rackspaces; therefore if you insert a plugin in, e.g. three different rackspaces, it will not use up three times as many CPU cycles.

Visit [this blog article](#) to learn more about rackspaces and variations and also [this blog article](#) to learn more about program changes and how you can manage them.

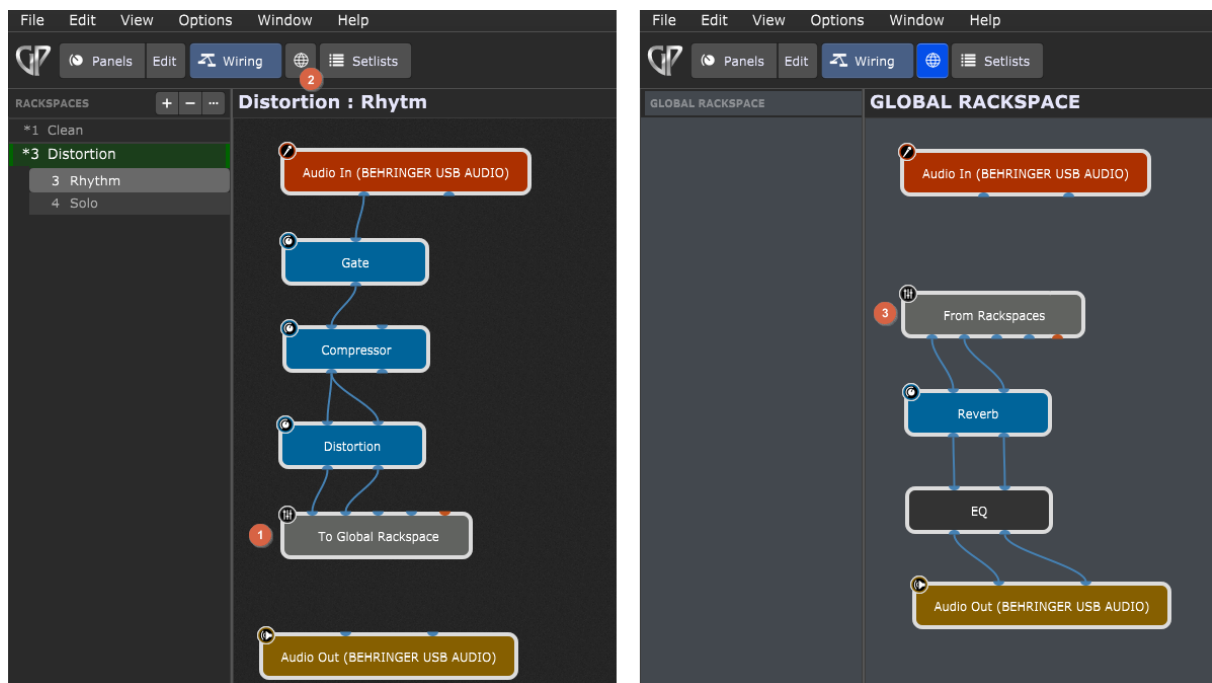
#### 4.2.1 Global Rackspace

The Global rackspace is a special rackspace that allows audio plugins to be shared and used by normal rackspaces (from now on, these will just be called "rackspaces"). The Global rackspace is always active. For example, you can create global effects - e.g. a

common reverb, delay, or chorus, instead of creating them in individual rackspaces. Audio and MIDI from the currently active rackspace can then be sent to the Global rackspace for processing. You can of course also insert regular synth plugins along with MIDI In blocks to the Global rackspace. This would allow you to, for example, have a piano or organ that is always available.

The Global rackspace is bidirectional. That means that while audio and MIDI can be sent from rackspaces to the Global rackspace (to support common effects as described above), audio and MIDI can also be sent from the Global rackspace back to whatever rackspace is currently active. This is particularly valuable for guitarists or singers who may want to always add a certain amount of "pre-processing" to their input, for example.

Below is an example of a rackspace and the Global rackspace working together:



The signal flow is easy to understand; note the special block called [To Global Rackspace](#) (1) in the left screenshot, where signals are directed to the Global rackspace instead of being sent to the Audio Out block. The Global rackspace can be accessed by clicking on the globe icon (2).

In the right screenshot, the Global rackspace wiring layout can be seen. Note the special block [From Rackspace](#) (3), used to receive audio (and MIDI) sent from rackspaces to be further processed serially through the Reverb and EQ blocks before being sent to the Audio Out.

When the Clean rackspace is active (in which no distortion or compressor plugins are in use), the signal flow is still ultimately directed to the Global rackspace and further processed by Reverb and EQ and so there is no need to include effects such as reverb or delay in every rackspace.

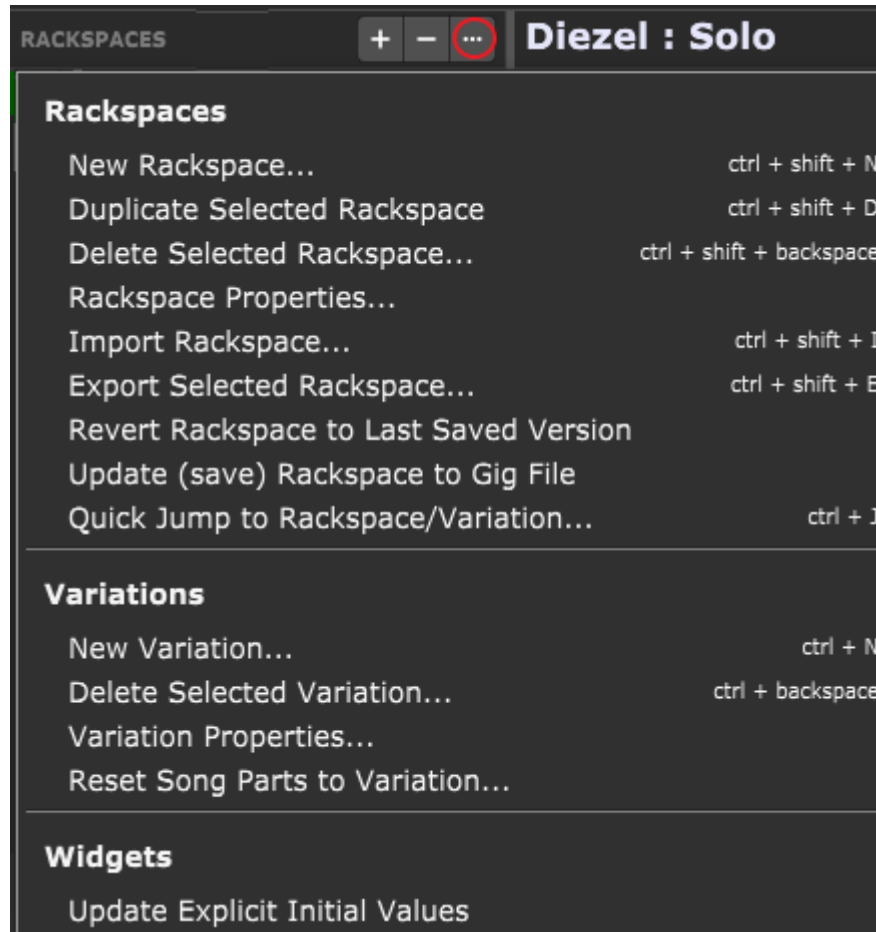
To conclude, while the Global rackspace can be used for many purposes, the three primary uses are:

- **Master effects processing** - allows you to send a signal from any rackspace to the Global rackspace where you can add effects, EQ, and other plugins for further signal processing. In previous Gig Performer editions, you must insert and connect all these effects in every rackspace if you want to use them; in Gig Performer, you can insert these effects into the Global rackspace and send signals from the active rackspace to the Global rackspace and thereby saving a lot of plugin duplication that was necessary in older versions (Gig Performer 3 and older).
- **Input pre-processing** - the Global Rackspace receives audio and MIDI input from your instrument, does some common pre-processing and sends the result to the active rackspace for specific effects needed for individual songs.
- **Permanent instrument patch** - allows you to add an instrument patch (e.g. a standard piano patch) that is always available from any rackspace. You can also create (say) a permanent vocal setup in the Global rackspace and just use rackspaces for your guitar, piano or other instrument layouts.

**Tip:** check [The Main Gig Performer window](#) chapter, point 21, to learn how to show/hide the Global rackspace controls when regular rackspaces are displayed.

## 4.2.2 Manage Rackspaces/Variations contextual menu

Right-clicking on any item in the rackspace list or clicking on the "three dots" button (indicated with the red circle) in the rackspace list in Panels or Wiring View opens the following menu:



This menu is split into three sections: Rackspaces, Variations, and Widgets.

### Rackspaces

**New Rackspace...** - creates a new rackspace. **Note:** the advantage of putting different sounds into different rackspaces (as opposed to creating one complex rackspace with multiple signal chains and complicated switching) is that you can go from one to the other, glitch-free, with the use of simple MIDI program change messages transmitted via a MIDI controller, such as up-and-down switches on a pedalboard - or even via a smartphone or tablet running an [OSC](#) app. What's more, Gig Performer's [Tail length](#) function allows you to set the maximum time before a sound is cut off, up to 10 seconds when you switch rackspaces.

**Duplicate Selected Rackspace** - duplicates the selected rackspace.

**Delete Selected Rackspace...** - deletes the selected rackspace.

**Rackspace Properties...** - opens the [Rackspace Properties](#) dialog.

**Import Rackspace...** - opens a dialog that allows you to select the desired rackspace file and import it.

**Export Selected Rackspace...** - exports the selected rackspace to the **.rackspace** file format.

**Revert Rackspace to Last Saved Version** - reverts the current rackspace to the last version saved. **Note:** click OK in the confirmation "Are you sure?" dialog only if you are sure you want to do this because all changes made since the file was last saved will be permanently lost. If you want to save your work first, click on the Cancel button, then go to File -> Save or File -> Save As...

**Update (save) Rackspace to Gig File** - saves the selected rackspace to the currently open .gig file; make sure to save a gig first for this feature to work. **Note:** this option saves only the currently selected rackspace, and changes made to all other rackspace will not be saved; if you want to save/update all rackspace, click on File -> Save instead. Also note that you cannot add and save new rackspace this way - the rackspace must exist and must be saved to the gig file first (you cannot add a new rackspace and then select this menu item to update it because it is not in the gig file yet).

**Quick Jump to Rackspace/Variation** - opens the [Quick Rackspace/Variation finder](#).

## Variations

**New variation...** - creates a new variation. **Note:** creating a new variation always duplicates the currently selected one.

**Delete Selected Variation...** - deletes the selected variation.

**Variation Properties...** - opens the [Variation Properties](#) dialog.

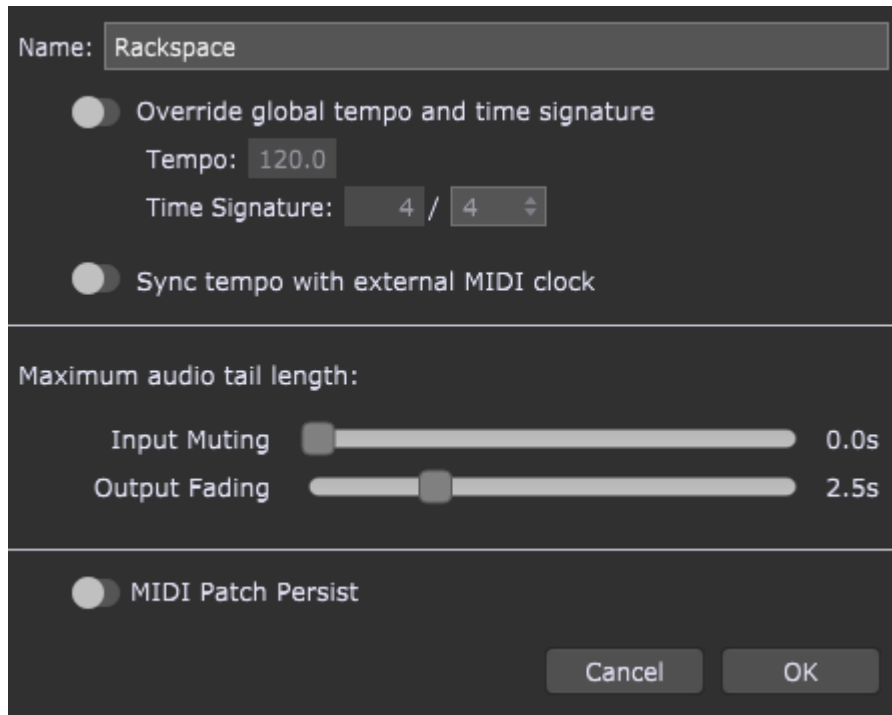
**Reset Song Parts to Variation...** - writes the current variation settings to every song part that uses the current variation. Use this command to update a sound and ensure the newest version is used everywhere the sound occurs.

## Widgets

**Update Explicit Initial Values** - clicking on this menu item updates widget initial values to the current ones when a gig is loaded. Initial values settings are defined in the [Widget Properties inspector](#).

### 4.2.3 Rackspace Properties

Double-clicking on a rackspace name (or choosing the Show Rackspace Properties... menu item from the [contextual menu](#)) opens a dialog that displays rackspace properties:



**Name** - to rename the rackspace, type a new name here.

**Override global tempo and time signature** - turn on this toggle button if you want the rackspace to have its own tempo value and time signature, then enter the desired BPM (beats per minute) and time signature. If this toggle button is turned off, the rackspace will use the [global tempo and time signature](#) information.

**Sync tempo with external MIDI clock** - turn on this toggle button if you want the rackspace to adjust its tempo from incoming MIDI clock messages (24 ticks per quarter note) sent from an external device, thus enabling echo/delay or arpeggiator plugins capable of receiving MIDI clock to be automatically synchronized with your external device. Visit [this blog article](#) to learn how to synchronize Gig Performer's tempo to an external MIDI clock or Link-enabled products.

**Maximum audio tail length** - sets the maximum time before a sound is cut off when you switch rackspaces. The maximum length is 10 seconds. Often known as an audio "tail", this feature is useful when using effects such as reverb or echo. In Gig Performer you can select Input muting and Output fading:



**Input muting** - move this scroll bar to select the desired number of seconds (up to 10s) in which input to this rackspace will be smoothly silenced.

**Output fading** - move this scroll bar to select the desired number of seconds (up to 10s) in which the audio output of this rackspace will be smoothly faded out.

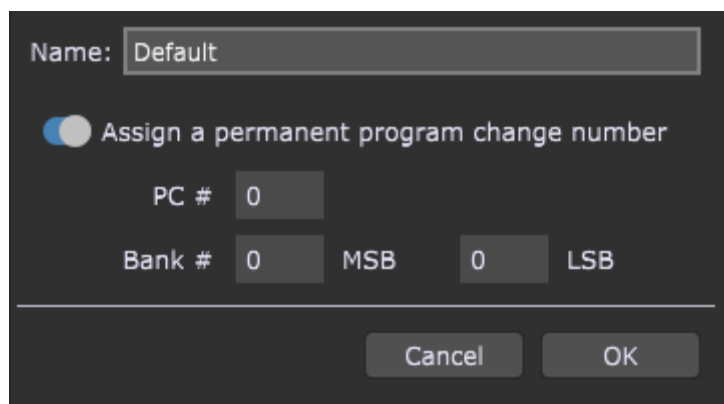
**MIDI Patch Persist** - if this feature is turned on, whenever you switch to another rackspace, notes that you are holding (either directly on your keyboard or via your sustain pedal) will continue sounding until you release the keys or the pedal. This feature works with any number of rackspaces, up to the limits of your system's available RAM and CPU resources, so you could conceivably keep the sound of five or more rackspaces going simultaneously.

If this feature is turned off, whenever you switch from one rackspace to another, the sound of the previous rackspace is immediately silenced even if you are holding notes down or have pressed the sustain pedal.

NB: While MIDI Patch Persist will work with multiple plugins when switching between two (possibly complex) rackspaces, plugins in the Global rackspace, however, cannot participate in the patch persist mechanism.

#### 4.2.4 Variation Properties

Double-clicking on a variation name (or choosing the Show Variation Properties... menu item from the [contextual menu](#)) opens a dialog that displays variation properties:



**Name** - to rename the variation, type a new name in this text box.

**Assign a permanent program change number** - when this toggle button is turned on, you can specify the program change number and, optionally, the bank select numbers (you

set MSB and LSB values separately) used to switch to this variation. **Note:** if a program change number has been assigned, that number will appear to the left of the variation name. If there is only one variation in the rackspace, the program change number will also appear to the left of the rackspace name; if there is more than one variation, an asterisk will appear there instead.

If this toggle button is turned off, Gig Performer will assign program change numbers in ascending order. For example, the first variation in the first rackspace will be called up when program change 0 is received, the first variation in the next rackspace will be called up when program change 1 is received, etc.

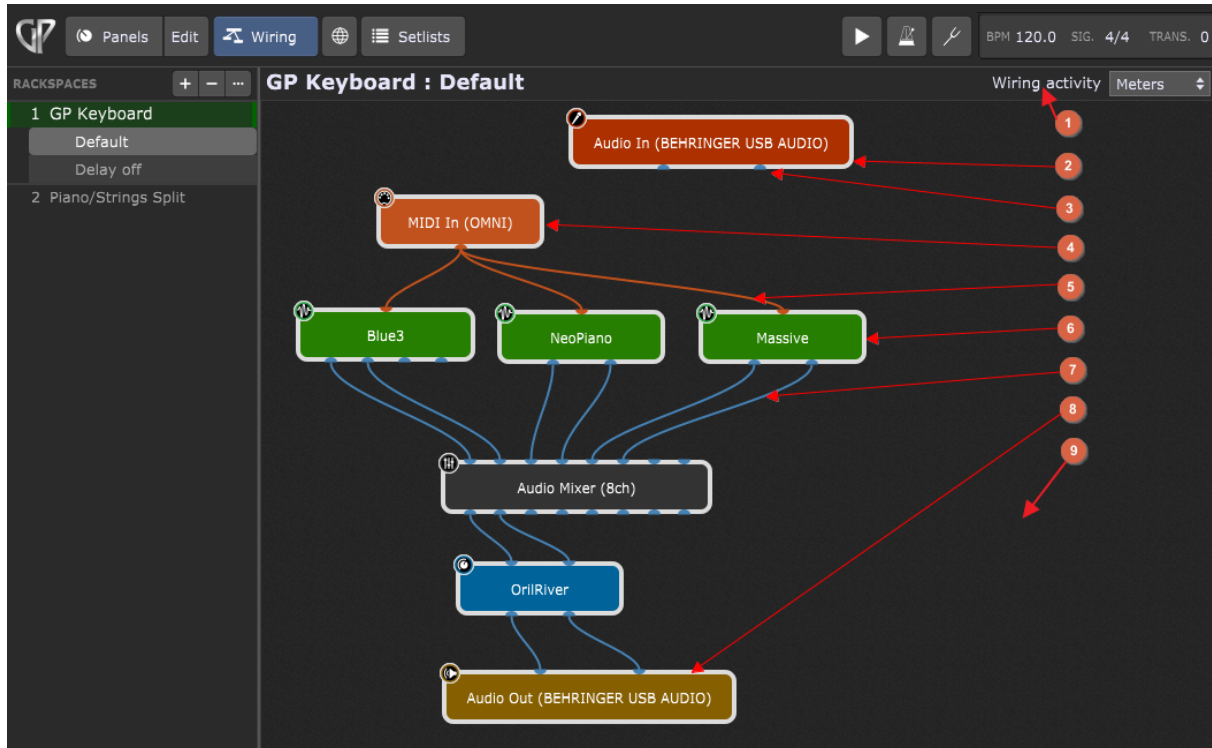
Make sure that you also review these options when it comes to assigning variations and program change numbers:

*Auto-assign a permanent PC number to new rackspaces (first variation), in the [General](#) tab.*

*Use zero-based PC numbers, in the [Global MIDI](#) tab.*

### 4.3 Wiring view

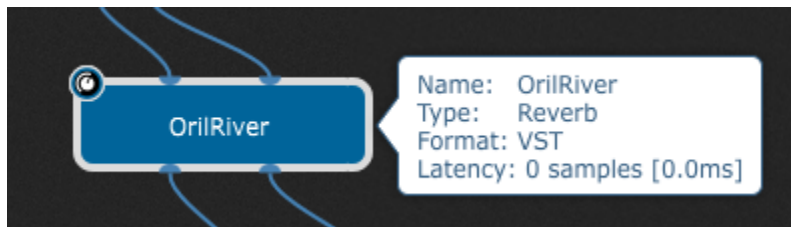
The Wiring view enables you to insert your plugins and connect them with virtual wires. Look at this example of a plugins layout in the screenshot below:



1. **Wiring Activity** - allows you to easily review the audio levels and MIDI activity in the Wiring view.
2. **Audio interface input block** - Gig Performer automatically shows inputs for audio interface device(s) known to the operating system.
3. **Plugin port** - plugins are interconnected by virtual wires, where an output port (or pin) of one plugin is connected to an input port of another plugin. Blue ports indicate audio ports, and orange ports indicate MIDI ports. Note that audio ports cannot be connected to MIDI ports (and vice versa).
4. **MIDI In (OMNI) block** - Gig Performer inserts the built-in MIDI In (OMNI) plugin by default. Other MIDI devices can be added in the same fashion as any other plugin. MIDI plugins are colored orange. Note that you can connect the MIDI output ports to multiple MIDI plugin input ports (the same applies to audio ports). Refer to the chapter [MIDI Inputs](#) to learn about these built-in plugins.
5. **MIDI connections** - shown as orange wires between from MIDI Out ports to MIDI In ports.

6. **Audio plugin** - virtual instruments are colored green, audio processors are colored blue, and some Gig Performer integrated plugins, such as Audio Mixer (8ch), are colored black. Also note the mini-icon located at the upper-left corner of a plugin, which visually indicates the plugin type.
7. **Audio connections** - shown as blue wires from audio output ports to audio input ports.
8. **Audio interface output block** - Gig Performer automatically shows outputs for audio interface device(s) known to the operating system.
9. **Wiring view design surface** - this large area of the screen is used to display your plugins and their interconnections. Drag a plugin to move or reposition it. Double-clicking a plugin opens its plugin editor. Refer to the [Wiring view features](#) to learn more about the features, such as zooming, scrolling, and selecting multiple plugins.

Hovering the cursor over a plugin brings up a tooltip with basic information about that plugin:



You can easily check how much latency your plugins introduce to your setup (in this example, there are 0 samples). The tooltip can be disabled by turning off the *Show information on plugin hover* option in the [Display Options](#).

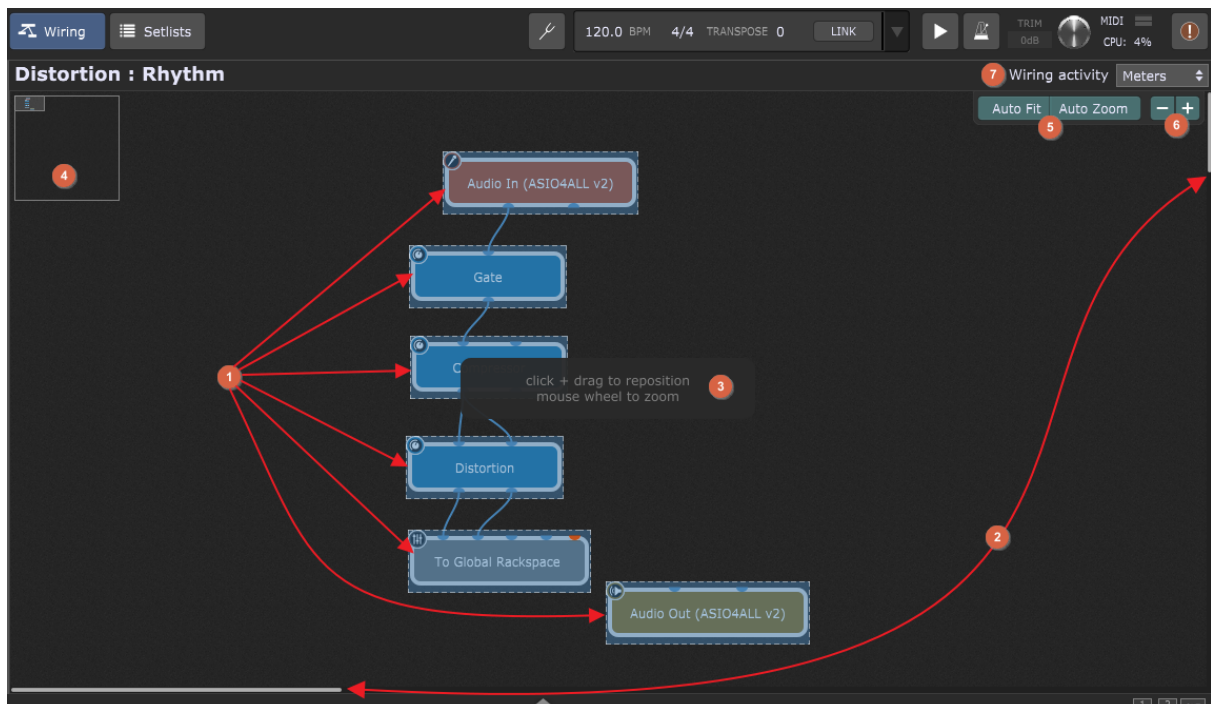
Note: You can control the Wiring view activity in the [Display Options](#) window. You can use Simple or Meters mode, which will allow you to have a great overview and troubleshoot your audio and MIDI signal flow.

#### 4.3.1 Wiring view features

The [Wiring view](#) provides you with many features to connect and organize your plugins efficiently.

You can select multiple blocks by dragging over them with your mouse. You can select all blocks (explicitly excluding the Audio I/O blocks) by pressing **Ctrl+A** (**Cmd+A** on Mac). If

you want to select specific blocks, hold the **Shift** key down and then click on each individual block **(1)**:



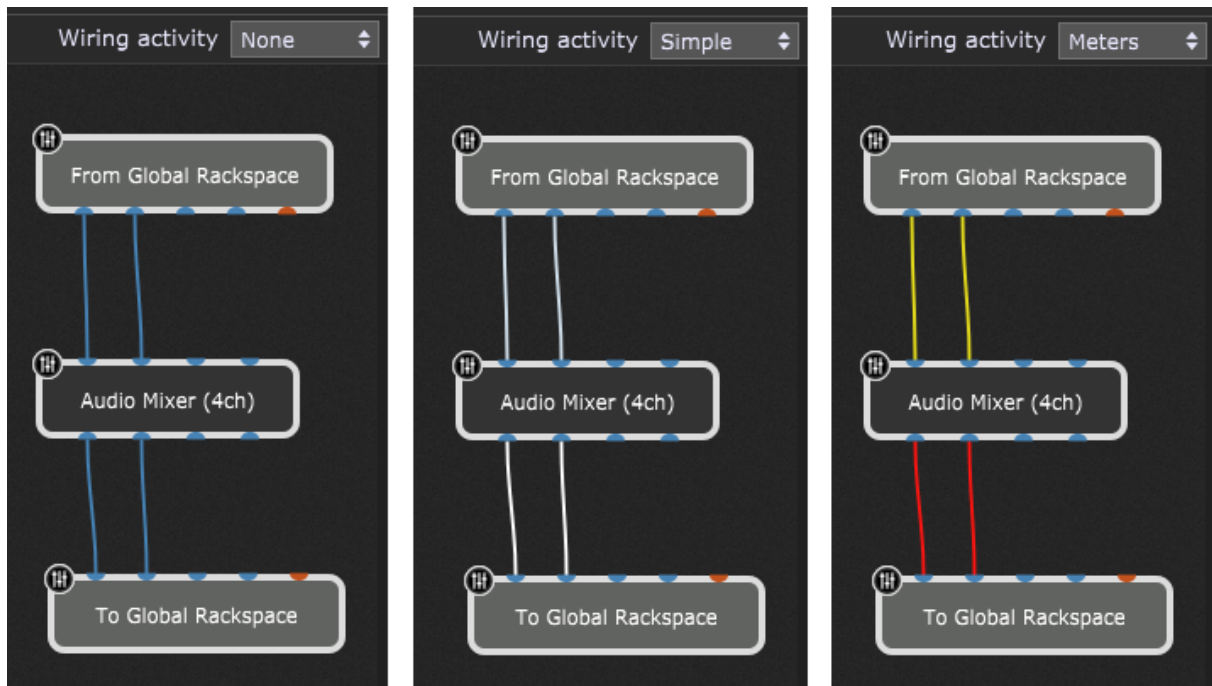
You can easily drag, copy and reposition multiple blocks together.

The design surface is virtual, giving you much more room to lay out your plugins without cluttering or placing them too close together. You can zoom in or out to focus on a particular area of the wiring view. Use the arrow keys on your PC keyboard or the horizontal and vertical scroll bars **(2)** to move around. When you press and hold the **Alt** key (**Option** key on a Mac), a tooltip appears **(3)**, notifying you that you can click on the design surface and drag it to navigate through it. While doing this, note the minimap in the left upper corner **(4)** that provides a quick overview of the whole design surface and the position of your layout.

You can use your mouse wheel to zoom in or zoom out the design surface in the Wiring view. If you also hold down the **Alt** key (**Option** key on a Mac), then **Auto Fit** and **Auto Zoom** options will appear in the upper-right corner **(5)**; clicking the Auto Fit button places all your plugins within the current screen and fits them to the current zoom level. Clicking the Auto Zoom button zooms all your plugins to fit the current screen. **Zoom Out** and **Zoom In** buttons are also available **(6)**, both of which allow you to set the desired zoom level.

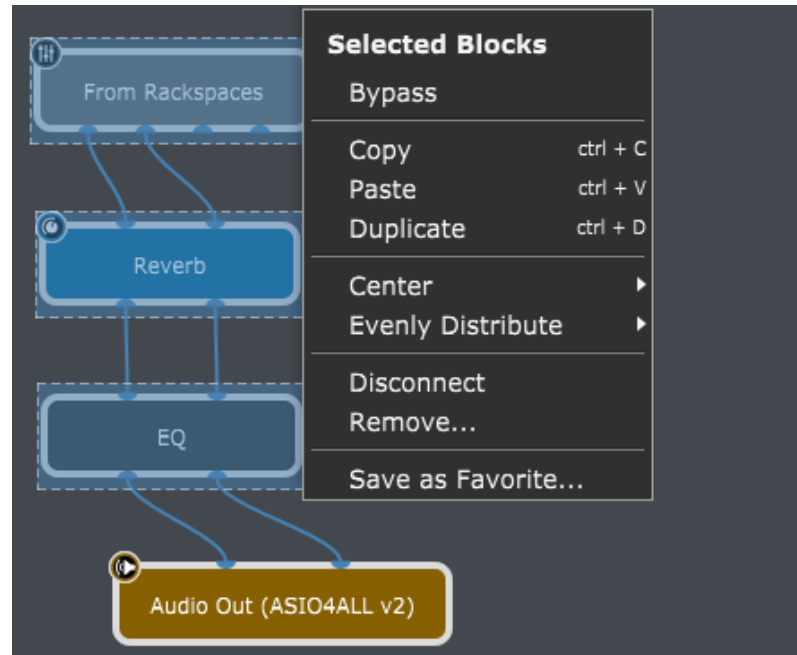
The Wiring Activity mode **(7)** allows you to easily review the audio levels and MIDI activity in the Wiring view. There are three modes of the Wiring activity:

- **None** - Turns off the feature.
- **Simple** - The color of audio wires changes from dark blue to lighter nuances or blue, and MIDI wires light up to show MIDI activity.
- **Meters** - The color of audio wires changes from green to red and MIDI wires light up to show MIDI activity.



This feature allows you to easily understand your audio and MIDI, discover where your audio might be too high, or troubleshoot your setup.

You can also save multiple interconnected blocks as favorites for quick reuse; select the blocks you want, right-click on your selection and select **Save as Favorite...** (refer to the [Selected Block contextual menu](#) section for more information):

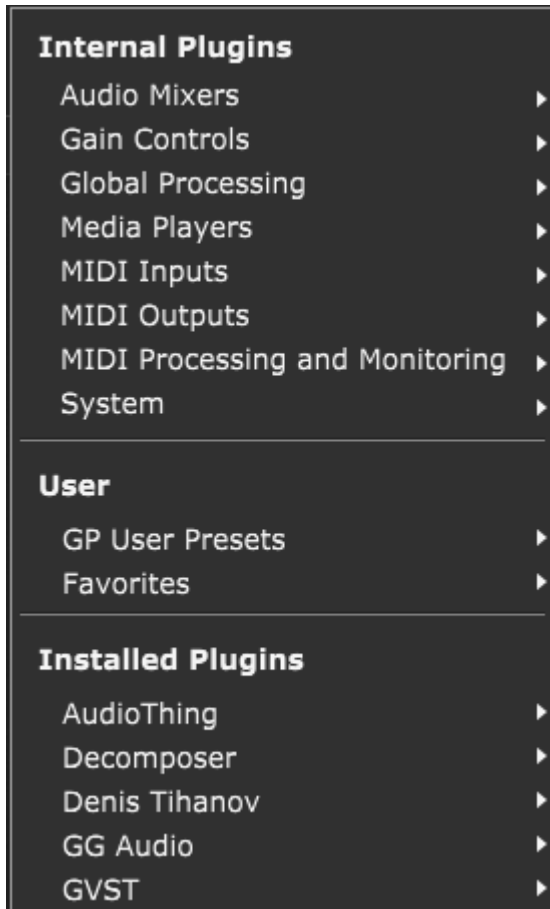


Favorites can be dragged in from the file system, and doing so will automatically insert the required plugins. Favorites also show up in the Wiring view [contextual menu](#) and the [Quick Plugin, Favorite or Preset Finder](#) dialog.

When you select multiple plugins, you can Center or Evenly distribute them; refer to the [Selected Block contextual menu](#) section for more information.

#### 4.3.2 Wiring view contextual menu

When you right-click anywhere in the background in the Wiring view, the following menu appears:



This menu is split into three main sections:

1. **Internal plugins** - allows you to insert one of the many built-in plugins such as Gain Controls or Media Players. Refer to the [Built-in plugins](#) chapter for more information.
2. **User** - allows you to insert plugins with user-saved Presets and Favorites (exported plugin selections will appear under Favorites). Refer to [this chapter](#) for more information.
3. **Installed Plugins** - all your installed plugins are available through these menus, organized alphabetically by the manufacturer of each plugin. Navigate through these menus to insert the desired plugin (click on [this chapter](#) to learn how to insert and connect plugins).



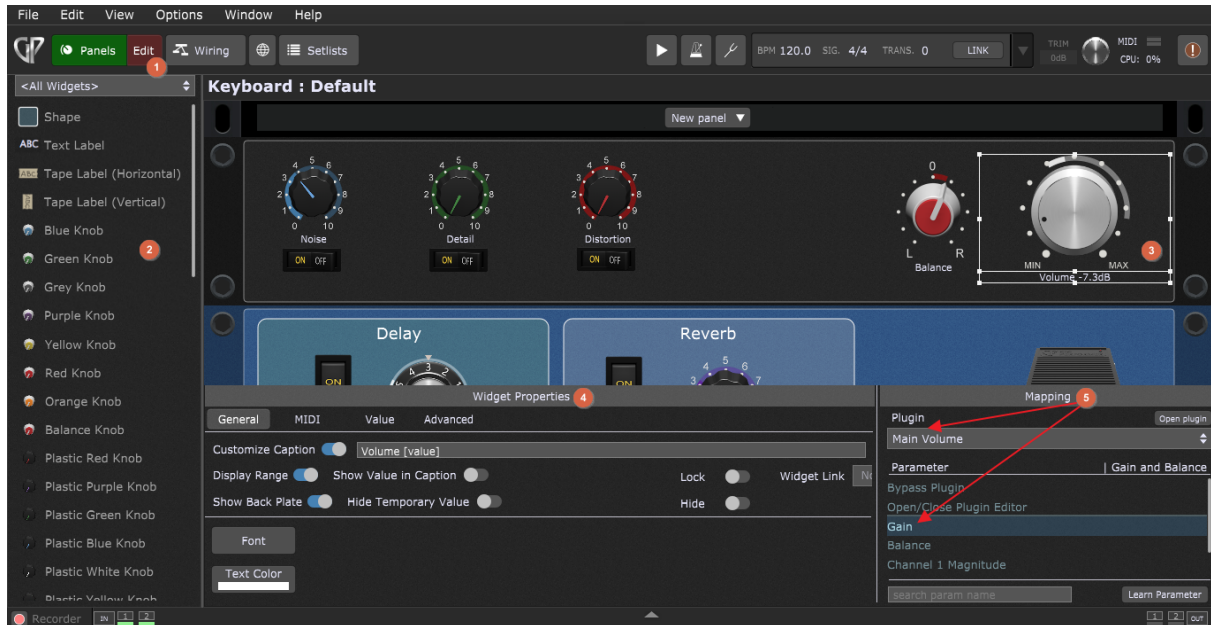
## 4.4 Panels view

The Panels view represents the front of the rackspace with their panels and widgets:



This view allows you to manage panels and widgets, assign plugin parameters, and assign hardware MIDI controls to widgets. Check the [Panels and Widgets](#) chapter to learn more. Make sure also to check out [the main Gig Performer window](#) section to familiarize yourself with various elements and their names in Gig Performer.

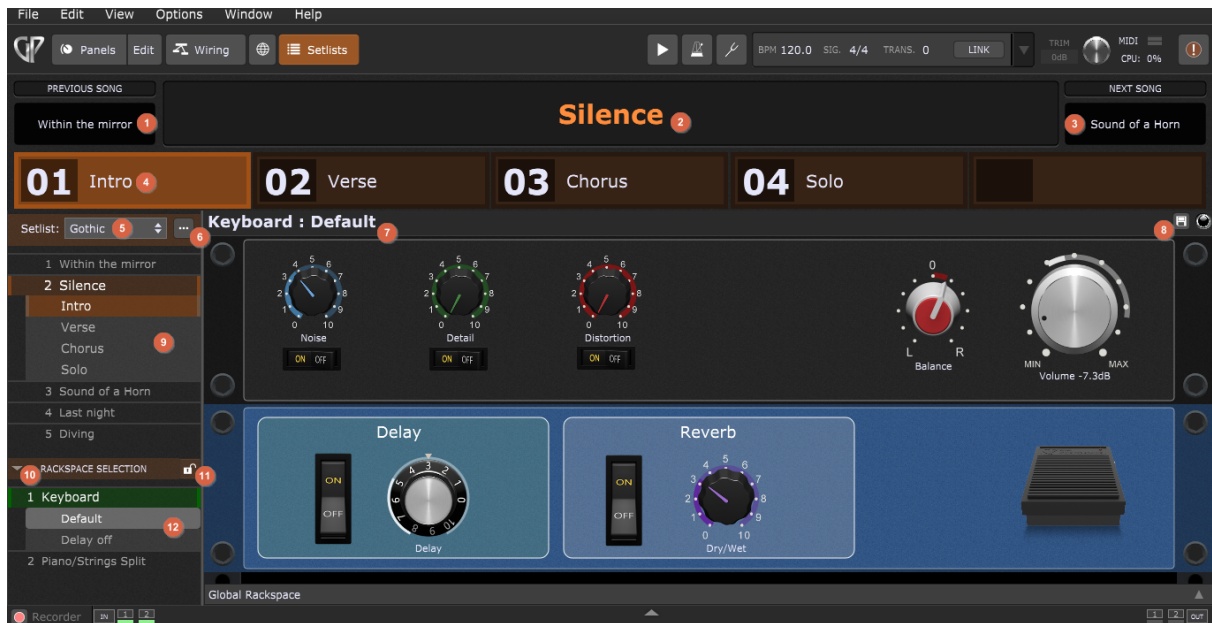
To see how widgets are connected to plugins and their parameters, click on the **Edit** button:



The Panels view in the Edit mode **(1)** enables you to [add new widgets](#) **(2)** by dragging them onto a panel and [connecting them with your plugins](#). Look at the example of a widgets layout in the screenshot above; the metal knob widget is selected **(3)**, and you can see its properties **(4)** and mapping **(5)**. Refer to the [Widget Properties inspector](#) chapter for more information. In the Mapping section, you can see that the selected widget is assigned to the Gain parameter of the Main Volume plugin.

## 4.5 Setlist view

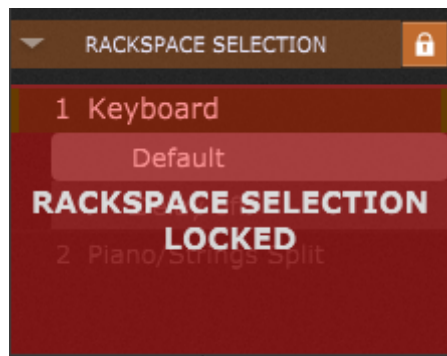
A setlist is a collection of songs. Songs consist of an arrangement of song parts. The Setlist view is shown in the screenshot below:



1. **Previous Song button** - click this button to move to the previous (earlier) song in the setlist.
2. **Song name** - displays the name of the current song. Double-clicking the song name opens the Quick Song Finder window.
3. **Next Song button** - click this button to move to the next (later) song in the setlist.
4. **Song part tile bar** - click on any of these large, highly visible buttons to instantly switch between song parts. Tiles can be configured to match the layout of a MIDI foot pedal controller. A song part is linked to a rackspace variation and can store edits to the variation it uses (refer to the [Songs and Setlists](#) chapter for more information).
5. **Setlists drop-down menu** - select a setlist to play from this menu. The All Songs setlist is always present in the list.
6. **More... button** - clicking this opens the [contextual menu](#), which contains numerous commands for managing setlists, songs and song parts.
7. **Rackspace/Variation display** - displays active rackspace and variation.
8. **Capture variation button** - click on this button to save edits you have made to a variation into the currently selected song part only. The changes are not saved to the underlying variation in the rackspace browser. This is a quick way of customizing

a variation and saving it. Check the [How to capture variation edits](#) chapter to learn more.

9. **Setlist editor** - allows you to reorder songs and song parts by dragging. Many important functions for the Setlist editor are found in the More... contextual menu. A song can be loaded by selecting in a setlist, in the Quick Song Finder or using Previous/Next buttons that can be mapped to incoming MIDI messages.
10. **Rackspace Show/Hide button** - click here to show or hide the rackspace list.
11. **Lock Song button** - this is a safety feature to prevent accidental changes. When this button is clicked, changes to rackspaces assignments in songs are not allowed. Attempting to change a rackspace produces an error screen:



12. **Rackspace browser** - rackspaces and variations can be selected here to map to song parts on the fly. There is no rackspace management in the Setlists view - for that, you must enter Panels or Wiring view. **Note:** Panels and Wiring views access only stored versions of variations; edits performed in the Setlist view are not available in those views. However, there is a method to push changes made in the Setlists view back to the underlying variations; see the *Push Song Part Changes to Rackspace* option in [this chapter](#).

When selected, a song can send a MIDI program change message out to an external MIDI device. An incoming MIDI program change message can select a song part and send out a sequence of MIDI messages upon selection. Check the [Song Properties](#) and [Song Part Properties](#) chapters to learn more.

Each song can have its own key, transpose value, tempo and time signature; the same also applies to song parts. A song can be linked to a [ChordPro](#) document for lyrics and chords.

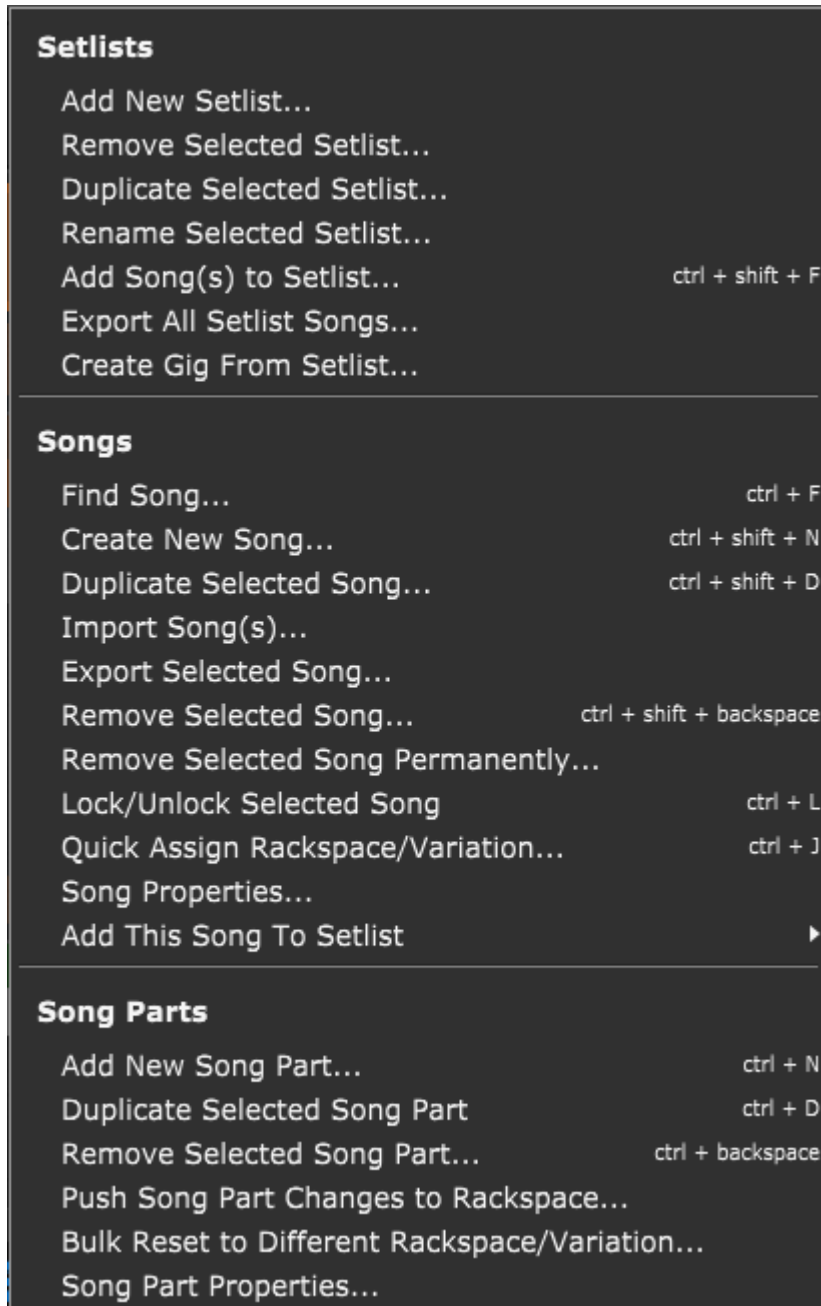
Assuming that your keyboard is connected to Gig Performer and you can play sounds, you can switch from one song to another by sending a standard MIDI Program Change

message from your MIDI keyboard to Gig Performer. By default, each song/song part is associated with a program change message, starting at 0. For other ways to select specific songs and song parts and for more details, see this in-depth blog article [How to change sounds and control plugins from your MIDI controller](#).

Visit [this blog article](#) to learn more about program changes and how you can manage them.

## 4.5.1 Manage setlists contextual menu

Click on the **More...** button or right-click on any song or a song part to open the Manage setlists contextual menu. This contextual menu contains various commands for managing setlists, songs and song parts:



This menu is split into three organizational units - Setlists, Songs, and Song Parts:

### Setlists

- **Add New Setlist...** - opens a setlist name dialog and adds a new setlist in the browser (the new setlist becomes the current setlist).
- **Remove Selected Setlist...** - removes the selected setlist and sets the previous setlist as the current setlist.

- **Duplicate Selected Setlist...** - opens a setlist name dialog and creates a copy of the selected setlist in the setlist editor (this duplicate becomes the current setlist). This menu item is grayed out when the *All Songs list* is active.
- **Rename Selected Setlist...** - opens a setlist name dialog and allows renaming of the selected setlist. The renamed setlist is automatically sorted in alphabetical order.
- **Add Song(s) to Setlist...** - opens a new dialog with the list of all songs where you can easily add a song (or songs) to the setlist.
- **Export All Setlist Songs...** - exports all songs from a setlist to a user-defined folder.
- **Create Gig From Setlist...** - allows you to export a setlist to a stand-alone gig file.

### Songs

- **Find song...** - choose this command to open the *Quick song finder* window displaying all the songs currently in the setlist. To add a song from the list in the window to the current setlist, select the song you want to add and either press the **Enter** key or double-click it. Press the **Escape** key or click the **Close** button when you have added all the songs you want in this setlist.
- **Create New Song...** - opens the [Song Properties](#) dialog and allows you to create a new song.
- **Duplicate Selected Song...** - opens a *Song name* dialog and makes a copy of the selected song.
- **Import Song(s)...** - opens a new dialog and allows you to select one or more **.gpsong** files to import. Refer to the [File extensions used in Gig Performer](#) chapter for more information on different file types in Gig Performer.
- **Export Selected Song...** - opens a new dialog and allows you to save a song in the **.gpsong** file.
- **Remove Selected Song...** - removes the selected song from the current setlist.
- **Remove Selected Song Permanently...** - removes the selected song from all setlists.
- **Lock/Unlock Selected Song** - choosing this menu item blocks changes to rackspace assignments in a selected song. This protects the selected song from accidental modification. Clicking on a rackspace in the browser when the song is locked produces an error message. Use **Shift** + click to apply to all songs.

- **Quick Assign Rackspace/Variation...** - opens the *Quick Rackspace/Variation Finder* window, allowing you to switch to any rackspace and variation immediately.
- **Song Properties...** - opens the [Song Properties](#) dialog.
- **Add This Song To Setlist** - adds a song to a setlist; select the song in the browser, then choose this menu item, picking a target setlist from the drop-down menu.

### Song parts

- **Add New Song Part** - opens a *Song part name* dialog and creates a new song part in the song currently selected in the browser.
- **Duplicate Selected Song Part** - makes a copy of the selected song part.
- **Remove Selected Song Part** - removes the selected song part from the current song.
- **Push Song Part Changes to Rackspace** - selecting this menu item overwrites the current variation with any changes made to widgets and then stored to a song part using the *Capture Variation Edits* button. After changes have been written back to the original variation, the red indicator next to the button goes out, and the *Compare* and *Revert* buttons disappear. **Note:** If you edit values on the fly and like the changes, this is how you save those changes. If you want to keep the original variation, duplicate it before clicking the *Capture Variation Edits* button.
- **Bulk Reset to Different Rackspace/Variation...** - this menu item enables replacing a rackspace variation everywhere it occurs with a different variation. This makes it fast and easy to change e.g. all of your piano sounds in one gesture when you get a new, more glorious piano instrument or sound. Choose the rackspace you want to use as a replacement, and after accepting the change in the confirmation dialog, all instances of the old variation will be replaced with the new variation.
- **Song Part Properties...** - opens the [Song Part Properties](#) dialog.

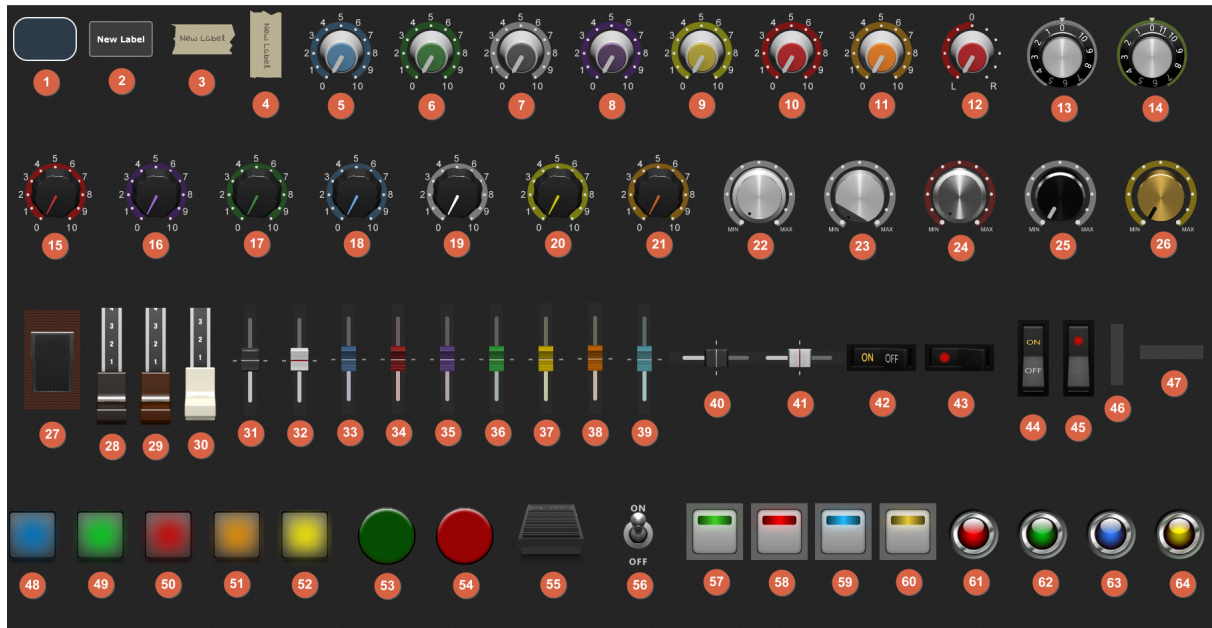
## 4.6 Widgets

A widget is an object (such as a knob, slider, button, or meter) used to control and display plugin parameters. Widgets can be controlled with your laptop's mouse or trackpad, or remotely from a MIDI controller. With OSC, widgets can also be controlled



from iOS / Android tablets and smartphones using the appropriate apps and other applications, such as MaxMSP or Ableton Live, that support OSC.

The picture below depicts all widgets available in Gig Performer:



1. Shape, 2. Text Label, 3. Tape Label (Horizontal), 4. Tape Label (Vertical), 5. Blue Knob, 6. Green Knob, 7. Grey Knob, 8. Purple Knob, 9. Yellow Knob, 10. Red Knob, 11. Orange Knob, 12. Balance Knob, 13. Old School Knob, 14. The 11, 15. Plastic Red Knob, 16. Plastic Purple Knob, 17. Plastic Green Knob, 18. Plastic Blue Knob, 19. Plastic White Knob, 20. Plastic Yellow Knob, 21. Plastic Orange Knob, 22. Metal Knob, 23. Metal Angled Knob, 24. Metal Knob (Dark), 25. Black Metal Knob, 26. Gold Knob, 27. Expression Pedal, 28. Black Drawbar, 29. Brown Drawbar, 30. White Drawbar, 31. Black Slider (Vertical), 32. White Slider (Vertical), 33. Blue Slider, 34. Red Slider, 35. Purple Slider, 36. Green Slider, 37. Yellow Slider, 38. Orange Slider, 39. Cyan Slider, 40. Black Slider (Horizontal), 41. White Slider (Horizontal), 42. Switch (Horizontal), 43. Switch Red (Horizontal), 44. Switch (Vertical), 45. Switch Red (Vertical), 46. Meter (Vertical), 47. Meter (Horizontal), 48. Pad Button Blue, 49. Pad Button Green, 50. Pad Button Red, 51. Pad Button Orange, 52. Pad Button Yellow, 53. Green Radio Button, 54. Red Radio Button, 55. Sustain Pedal, 56. Toggle Switch, 57. LED Button Green, 58. LED Button Red, 59. LED Button Blue, 60. LED Button Yellow, 61. Red LED, 62. Green LED, 63. Blue LED, 64. Yellow LED.

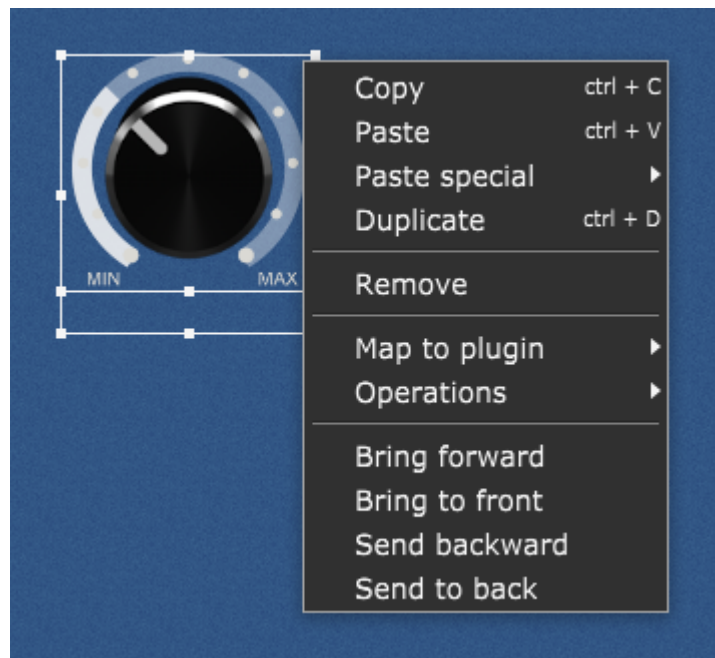
Widgets can be operated independently or in widget groups - refer to the [How to add a widget to a Widget Link group](#) and [How to add a widget to a Radio group](#) chapters for more information.

Historically, if you wanted to control plugin parameters, you had to use MIDI messages. Many plugins supported the ability to learn particular MIDI messages for specific parameters. However, we strongly encourage you to avoid this legacy approach and instead use Gig Performer's host automation support. Using host automation, you will control plugin parameters via widgets and optionally control widgets from your MIDI hardware. This may seem more complicated than it actually is. Gig Performer's ability to learn MIDI automatically makes the process simple while providing much more flexibility than directly controlling parameters via MIDI (please refer to [this chapter](#) for more information). Visit [this blog article](#) to learn the benefits of using the host automation.

**Note:** while some older plugins may not support host automation, plugins that don't support host automation are very rare.

#### 4.6.1 Widget contextual menu

If the Panels view, the Edit mode is active, and you right-click on a widget (or multiple widgets), the contextual menu appears:



**Copy** - copies the selection into the clipboard.

**Paste** - pastes the selection from the clipboard.

**Paste special** - allows you to paste font attributes, color attributes or both onto the selected widget(s). Note that you need to copy a single widget for this feature to work (it will not work if more than one widget is copied to the clipboard).

**Duplicate** - duplicates the currently selected widget. **Note:** duplicated widget does not duplicate the parameter mapping.

**Remove** - removes the currently selected widget (alternatively, press the Del or the Backspace key).

**Map to plugin** - allows you to map one or more widgets to the desired plugin from the list in one go (you still have to assign parameters to each one of them, of course).

**Operations** - allows you to apply a certain operation to one or more widgets. The supported operations are: Show/Hide widgets, Lock/Unlock widgets, Ignore variations/Update on variations, specifying the desired Widget Link group, and specifying the desired Radio group.

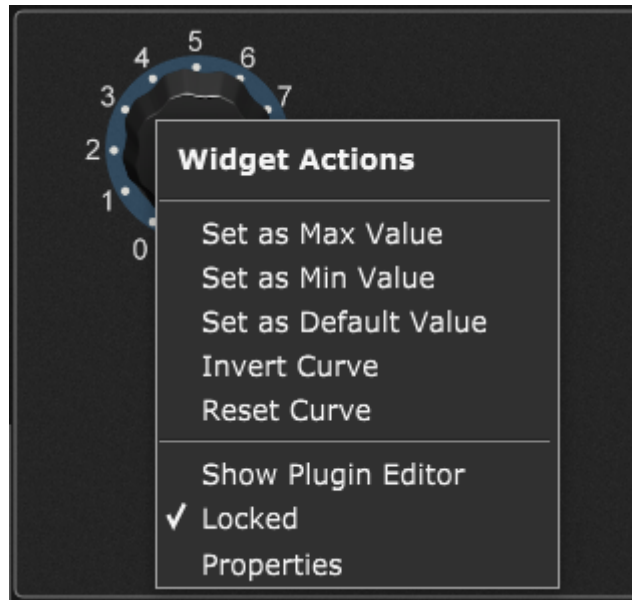
**Bring to forward** - places the currently selected widget one place higher in the order. Refer to the [Widget z-ordering](#) chapter for more information.

**Bring to front** - places the currently selected widget above all others (the highest priority in the order).

**Send backward** - places the currently selected widget one place lower in the order.

**Send to back** - places the currently selected widget below all others (the lowest priority in the order).

**Note:** if you click on a widget while holding the **Shift** key in the Panels view (not in Edit mode), the Widget actions menu will appear:



**Set as Max value** - sets the current widget value as the maximum widget value.

**Set as Min value** - sets the current widget value as the minimum widget value.

**Set as Default value** - sets the current widget value as the default widget value (check the [Widget Properties Inspector](#) chapter to learn more).

**Invert Curve** - inverts the widget's scaling curve.

**Reset Curve** - resets the widget's maximum and minimum values and the scaling curve to linear (which is the default curve).

**Show Plugin Editor** - shows the plugin editor window if this widget is associated with a plugin.

**Locked** - allows you to lock or unlock the widget from this menu (check the [Widget Properties Inspector](#) chapter to learn more).

**Properties** - clicking this menu item opens Edit mode, and the [Widget Properties Inspector](#) will be displayed for this widget.

## 4.7 Plugins

A plugin is a separate piece of software that can be loaded into a DAW or a live performance host such as Gig Performer. Some plugins (called "virtual instruments") generate audio, for example, emulations of musical instruments and synthesizers, while

effects plugins alter incoming audio signals, such as from a microphone, electric guitar, or virtual instrument, offering processes like equalization, filtering, compression, limiting, reverb, delay, flanging, chorusing, phasing, and the like. Many thousands of plugins are available in the audio/music marketplace, both free and commercial. Many free plugins are of extremely high quality.

In Gig Performer, a plugin is also referred to as a plugin block or just "block".

Some of the blocks and their elements in Gig Performer are depicted in the screenshot below:



Double-clicking a plugin opens its editor (1), e.g. here is a third-party plugin called Nick Crow Lead 8505, as indicated in (8). When you insert a plugin, it opens its editor by default; this setting can be changed by turning off the *Show plugin editor when inserted* setting in [Display options](#).

The Audio input block is colored dark orange (2), and the audio output block is colored dark yellow (6); the actual audio inputs and outputs depend on your audio setup. You can review your settings under [Audio I/O Options](#).

Virtual instruments, such as **(3)** and **(7)**, are colored green. The third party-plugin called NeoPiano is bypassed. Learn more about bypassing a plugin in the [Plugin contextual menu](#) section.

Audio processors, such as effects plugins **(4)** and **(8)**, are colored blue. Note the blue line at the lower-right corner of plugin **(4)**; hovering a mouse pointer over this line will show a tooltip stating that *This plugin has more audio channels. Click to adjust visibility*. Refer to the [Plugin contextual menu](#) section to find out more about limiting pins (or plugin ports). To find out more about wiring your plugins and ports, refer to the [Wiring view](#) and [Inserting and wiring your plugins](#) sections.

MIDI plugins such as MIDI Monitor **(5)** are colored orange.

A plugin may have an OSC/GPScript handle assigned to it. Note the Blue3 plugin **(7)** and its handle name "script", located at the upper-left corner of the plugin. Two symbols {S} and "wireless icon" are located at the upper-right corner of the plugin, indicating that the defined handle will be used as a GPScript handle and an OSC handle, respectively. Refer to the [Plugin contextual menu](#) section for more information on setting up a handle.

As mentioned in the chapter [Global Rackspace](#), the gray-colored plugin To Global Rackspace **(9)** is used to send audio signals to the Global rackspace.

When you click on a plugin block, it becomes selected **(11)** (NB: when you mute a [Gain Control](#) plugin, the "M" indicator will appear in the upper-right corner of the plugin). Many integrated Gig Performer plugins, such as Audio Mixer **(10)**, are colored black. A plugin is also colored black if it doesn't report its type or represents a utility plugin, such as some mastering plugins **(12)**. Suppose you hover with your mouse over any plugin. In that case, a tooltip pops up, showing the plugin's name, type, format and latency **(13)** (note: the tooltip can be disabled by turning off the *Show information on plugin hover* toggle box in the [Display Options](#)).

An exception to the common rectangular shape of plugins in Gig Performer is the MIDI Channel Constrainer plugin **(14)** (to learn more about this plugin and set it to block all MIDI traffic, refer to the [MIDI Channel Constrainer](#) chapter).

Suppose the computer you're using to access the gig file is missing some of the plugins that were present when the original file was created. In that case, Gig Performer will create a "chameleon" plugin that mimics all the properties of the original plugin except the actual processing. This will allow you to modify the file, save it and later re-open it on a computer that has all of the required plugins installed:



The chameleon plugin is colored dark red and reports the name of the real plugin followed by the word "(MISSING)". If you double-click this plugin, its window opens and states that this plugin is either missing or disabled on this system. This feature is extremely useful if someone sends you a gig file that includes a reverb plugin (say) that you don't have currently installed. You simply open that gig file and see which plugin is missing. Afterward, you can either install the missing plugin or replace it with your own reverb plugin – right-click on the chameleon block, then select *Replace plugin* and choose the existing alternative reverb plugin on your machine.

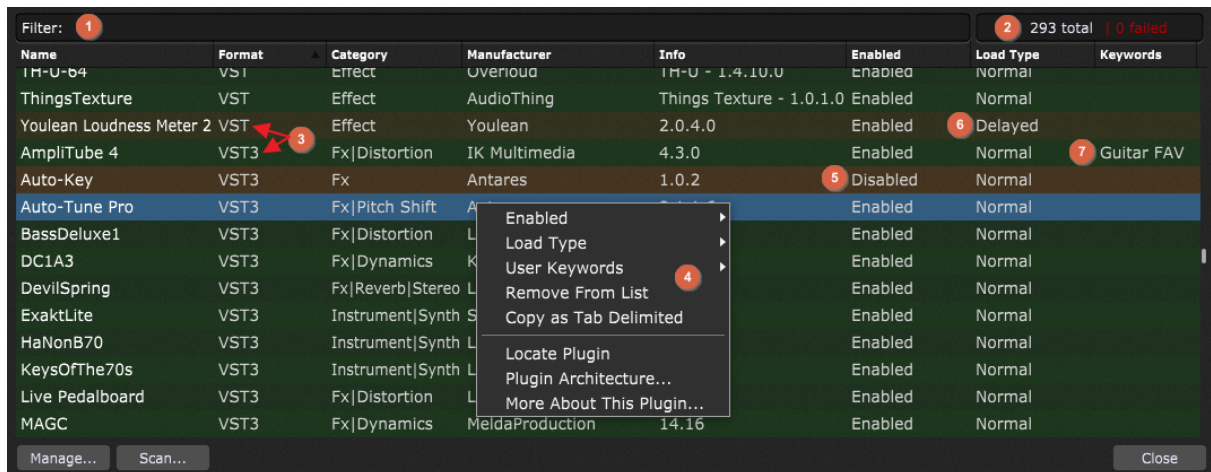
Gig Performer lets you see the list of used plugins for the loaded gig file. Refer to [this chapter](#) to learn more.

If you disable your plugin through the [Plugin Manager](#), it becomes a chameleon plugin. This feature is also useful for troubleshooting; check this blog article to find out [how you can identify a plugin that crashed](#).

Refer to the [Built-in plugins](#) chapter to learn more about the plugins integrated with Gig Performer.

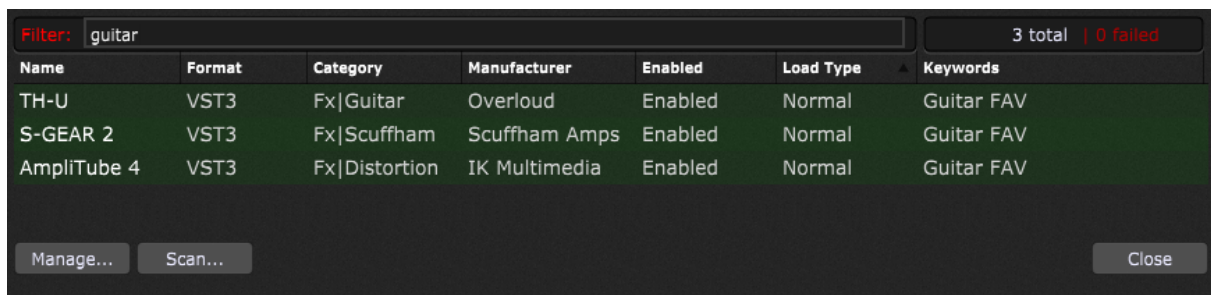
### 4.7.1 Plugin Manager

The Plugin Manager lets you view all of the plugins installed in your system, and can be sorted by name, format (VST, VST3 or AudioUnit), category (Effect, Mastering, Synth, Distortion, etc.), manufacturer name, enabled status, load type or user-defined keywords:



To access the Plugin Manager, press **Ctrl+Shift+P** (**Cmd+Shift+P** on Mac) or choose **Plugin Manager** from the Window menu. Click on the column name, such as "Name" or "Format" to sort the column alphabetically by that category. Click that category a second time to reverse the alphabetical order of the column.

The Filter field **(1)** allows you to filter the plugins list by the desired query:



With **(2)** is represented the total number of plugins, followed by the number of failed plugins.

**Note:** Windows users can install and use VST and VST3 plugins; examples are indicated in **(3)**. MacOS users can install and use VST, VST3 or AudioUnit plugins.



However, be aware that a plugin's manufacturer might report the wrong category or even not report it at all.

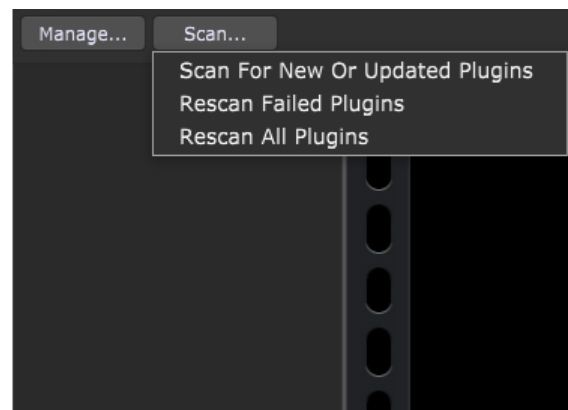
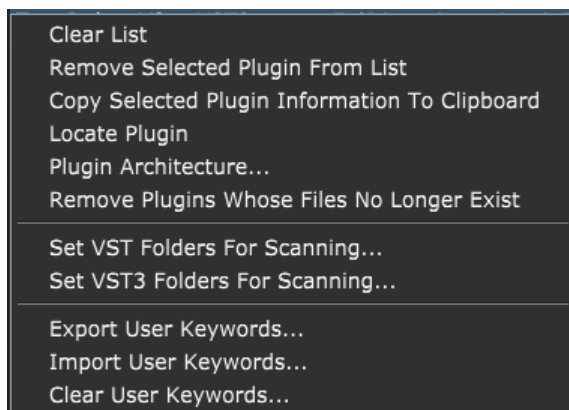
Right-clicking on any plugin name opens a pop-up menu **(4)** that allows you to:

- *Enable* or *Disable* any plugin; alternatively, right-click on the plugin entry, the Enabled column and select Enabled or Disabled, indicated with **(5)**. **Note:** when you disable a plugin, you will need to reload your gig file for the changes to take effect. When you disable plugins you don't need, the only benefit will be finding plugins much faster — the [contextual menu](#) will be smaller, and the [Quick Plugin Finder](#) will return a shorter list. It won't impact loading time at all - Gig Performer only loads plugins needed for the gig.
- specify its Loading type - "Normal" or "Delayed"; alternatively, right-click on the plugin entry, the Load Type column and select the desired option, as indicated with **(6)**. If you encounter third-party plugins that regularly fail to reinitialize successfully (i.e. they don't launch or display with the last saved value when recalling a .gig file), try designating them here as "Delayed" (they will be overlaid with a dark brown see-through bar). Gig Performer will slightly delay their loading time the next time it is launched, which may help solve the problem.
- Define custom user keywords; alternatively, right-click on the plugin entry, the Keywords column (or simply double-click the Keywords area) and type in the desired keyword **(7)**. As shown in the screenshot above, you can filter your plugin list by these keywords.
- Remove the plugin entry from the list.
- Copy as tab delimited, if you want to copy all column values of the entry as tab-delimited values, which can be easily pasted into a spreadsheet program. You will usually use this feature on several plugins or all plugins from the list.
- Locate plugin, i.e. open its containing folder. If you see duplicates of your VST plugins in the list, use this feature and review your duplicates locations to determine where they are installed. Afterwards, remove the unnecessary plugins.
- Determine the plugin architecture (Windows only), which is useful for troubleshooting purposes, i.e. determining whether the failed plugin is 32-bit.
- And finally, see more information about the selected plugin, i.e. visit plugin developer's website.

Plugins labeled "Disabled" in the Plugin Manager are overlaid with a brown see-through bar and will not appear in the list of available plugins when you right-click in the Wiring view surface area. This enables you to reduce clutter if you are sure you won't need to access particular plugins when creating a rackspace.

Additionally, the Plugin Manager includes the **Widget Refresh** column that allows you to explicitly refresh mapped widgets after loading a GP User Preset. This feature is rarely needed and is useful for plugins that don't notify a host for their parameter changes. Check out [this blog article](#) to see this feature in action.

At the bottom of the Plugin Manager window are two buttons named **Manage...** and **Scan...** :



Clicking on the **Manage...** button allows you to:

- **Clear list** - clears all plugins from Gig Performer's Plugin Manager. **Note:** this operation will not erase any plugins from your computer; it simply makes them unavailable to Gig Performer until you perform a new scan. Click on the **Scan -> Scan for new or updated plugins** menu item to begin the scan.
- **Remove selected plugin from list** - removes the plugin from the Plugin Manager list (note: select a plugin first to make this option operational). This menu item does not remove the plugin (and its files) from your operating system. **Note:** to remove a plugin from the list permanently, find the desired plugin that you want to remove and then delete its files (you can use the *Locate Plugin* contextual menu item) from your drive (alternatively, move them to another folder location that is not already specified in the Plugin Manager) and then select the *Remove Plugins Whose Files No Longer Exist* option.
- **Copy selected plugin information to clipboard** - clicking this menu item copies selected entries in the Plugin Manager to the clipboard in the tab-delimited format.

Afterward, simply paste the clipboard contents into a (say) spreadsheet application to create a nicely formatted list of validated plugins on your computer.

Name	Format	Category	Manufacturer	Info
MAGC	VST3	Fx Dynamics	MeldaProduction	14.16
MAmp	VST3	Fx Distortion	MeldaProduction	14.16
MAnalyzer	VST3	Fx Tools	MeldaProduction	14.16
MAutoAlign	VST3	Fx Tools	MeldaProduction	14.16
MAutoDynamicEq	VST3	Fx EQ	MeldaProduction	14.16
MAutoEqualizer	VST3	Fx EQ	MeldaProduction	14.16
MAutoEqualizerLP	VST3	Fx EQ	MeldaProduction	14.16
MAutopan	VST3	Fx Modulation	MeldaProduction	14.16
MAutopanMB	VST3	Fx Modulation	MeldaProduction	14.16

	A	B	C	D	E	F
1	Name	Format	Category	Manufacturer	Description	Version
2	MAGC	VST3	Fx Dynamics	MeldaProduction	MAGC	14.16
3	MAmp	VST3	Fx Distortion	MeldaProduction	MAmp	14.16
4	MAnalyzer	VST3	Fx Tools	MeldaProduction	MAnalyzer	14.16
5	MAutoAlign	VST3	Fx Tools	MeldaProduction	MAutoAlign	14.16
6	MAutoDynamicEq	VST3	Fx EQ	MeldaProduction	MAutoDynamicEq	14.16
7	MAutoEqualizer	VST3	Fx EQ	MeldaProduction	MAutoEqualizer	14.16
8	MAutoEqualizerLP	VST3	Fx EQ	MeldaProduction	MAutoEqualizerLP	14.16
9	MAutopan	VST3	Fx Modulation	MeldaProduction	MAutopan	14.16
10						

Gig Performer also copies custom keywords (if defined) and the file information (i.e. plugin file path) to the Clipboard.

**Note:** the keyboard shortcut **Ctrl+C** (or **Cmd+C** on Mac) also works to copy selected plugin information to the clipboard in CSV format. To select all plugins and then copy plugin information for all validated plugins, you can press **Ctrl+A** and then **Ctrl+C** (**Cmd+A** and then **Cmd+C** on Mac).

**Note:** in the Plugin Manager window, the Info column also contains version numbers of plugins.

- **Locate plugin** - opens the folder containing the selected plugin (note: select a plugin first to make this operational - this only works for VST plugins).
- **Plugin Architecture...** (Windows only) - allows you to easily see not only whether the plugin is 32-bit or 64-bit, but also the plugin file path. This feature is useful for troubleshooting failed plugins and determining their architecture - as already mentioned in [this chapter](#), only 64-bit plugins are supported.
- **Remove plugins whose files no longer exist** - use this option to update the Plugin Manager to reflect the accuracy of what plugins are available.
- **Set VST folders for scanning...** - opens the window which allows you to change, add, or delete the VST plugin folders to be scanned (see the [Troubleshooting](#) section for more information).
- **Set VST3 folders for scanning...** - opens a window allowing you to change, add, or delete the folders to be scanned for VST3 plugins (see the [Troubleshooting](#) section for more information).
- **Export user keywords** - easily export all your defined keywords into a **.userkeywords** file. The best practice is to create a backup of your keywords occasionally.
- **Import user keywords** - allows you to import a **.userkeywords** file, which contains all your defined keywords.
- **Clear user keywords** - clears all defined user keywords.

Clicking on the **Scan...** button allows you to:

- **Scan for new or updated plugins** - this causes Gig Performer to immediately scan your system for any newly installed or updated plugins; any plugins found will be added to the Plugin Manager list.
- **Rescan failed plugins** - rescans any failed plugins. Note that Gig Performer supports only 64-bit plugins; any 32-bit plugin found will fail to initialize.
- **Rescan all plugins** - this option will go through your drive again, looking for all plugins in specified folders.

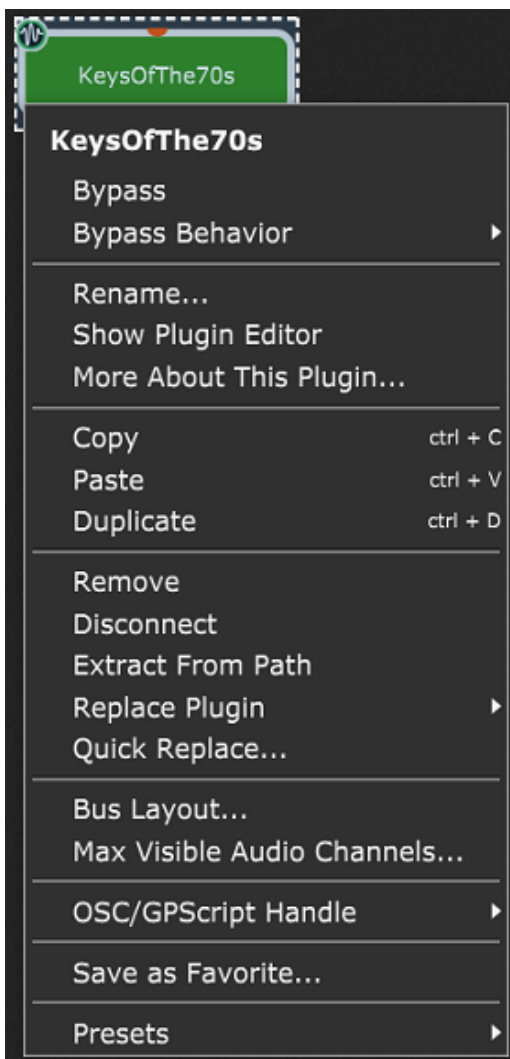
For more information on how to download and use plugins in Gig Performer, check this blog article: [Top websites for free audio plugins and how to use them in Gig Performer](#).

**Note:** double-clicking a plugin in the Plugin Manager inserts that plugin in the Wiring view.

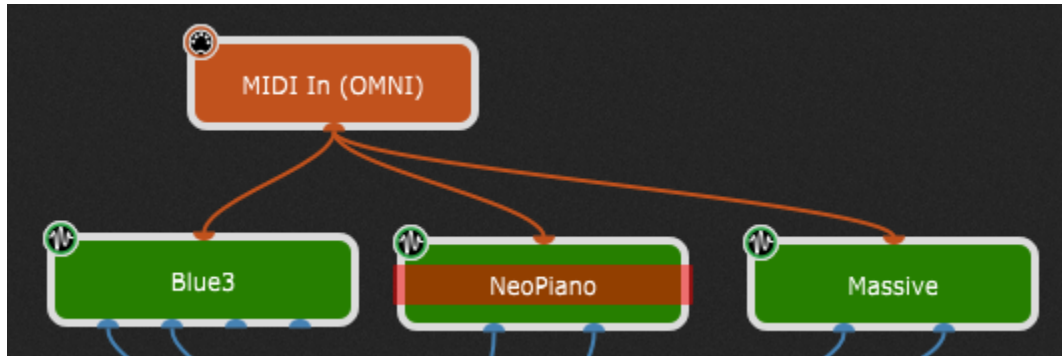
Refer to the chapter [Plugins in Use](#) to see a detailed list of plugins used in the current gig file.

#### 4.7.2 Plugin contextual menu

Right-clicking on any plugin opens a popup menu with many options:



**1. Bypass** - places the plugin in bypass mode. Plugin blocks that are bypassed are overlaid with a red transparent rectangle, as shown in the illustration below:



Similarly, you can select multiple plugins and bypass them.

**Note:** There is a difference between *bypassing* a plugin and *muting* it.

For virtual instrument plugins, *bypassing* turns off processing for that plugin, thus stopping the sound from being generated.

*Muting*, which is accomplished by inserting a Gain Control, Gain and Balance, Gain and Balance (m->s), or one of the audio mixer plugins and clicking the Mute button or setting its volume slider to 0, causes an audio signal to continue to be generated, but with the output silenced.

For example, if you are playing back a loop or click track during a live performance but you don't want the audience to hear it for a period of time, mute the plugin instead of bypassing it. That way, it will continue to play (in your headphones, if so routed) until you're ready for the audience to hear it again.

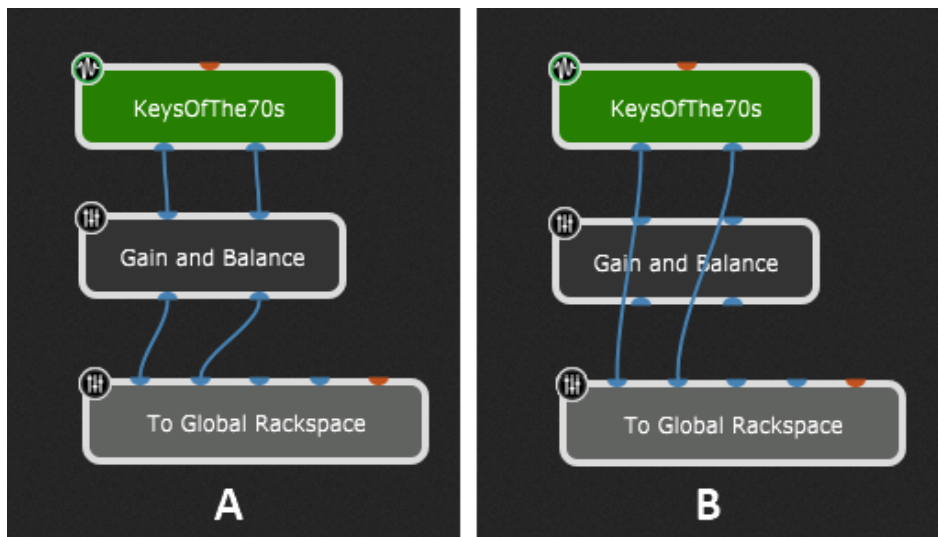
In contrast, bypassing an effects plugin causes incoming signals to pass through the plugin unchanged.

**2. Bypass Behavior** - if a particular plugin makes noises when you bypass it, you can use the "Fade" option to reduce that noise. In most cases it is not necessary.

**3. Rename...** - renames the displayed name (useful if you use the same plugin more than once).

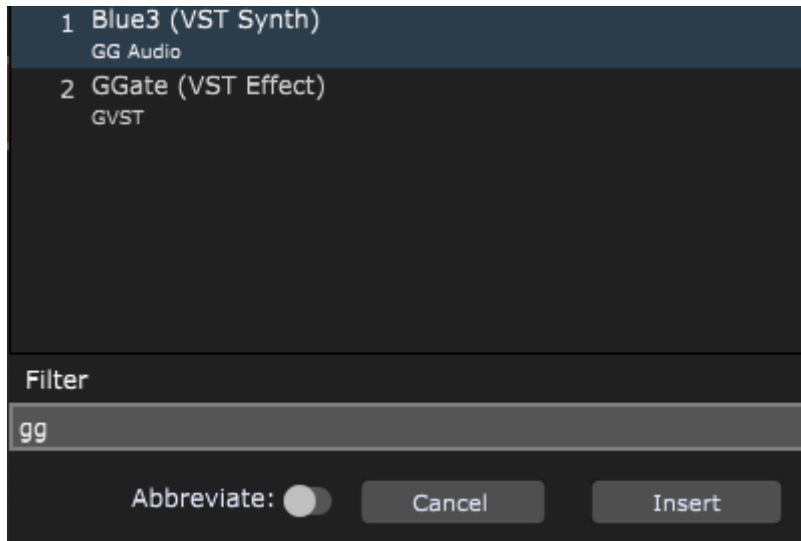
**4. Show Plugin Editor** - opens the plugin's editor (same as double-clicking on the plugin block).

5. **More About This Plugin...** - opens the default Web browser and loads the plugin's homepage (an Internet connection is required to load the plugin's homepage).
6. **Copy** - copies the plugin into the clipboard.
7. **Paste** - pastes the plugin from the clipboard.
8. **Duplicate** - creates a duplicate of the plugin. **Note:** check the [Additional tips](#) chapter to learn how to duplicate a plugin with inputs.
9. **Remove** - removes the plugin from the Wiring area surface. You can also select one or more plugin blocks by dragging over them with your mouse and removing them from the Wiring area surface by pressing either the **Delete** or **Backspace** keys on your keyboard.
10. **Disconnect** - removes all connections to and from this plugin.
11. **Extract from path** - extracts this plugin block from the audio or MIDI path; see the following example **(A)**:



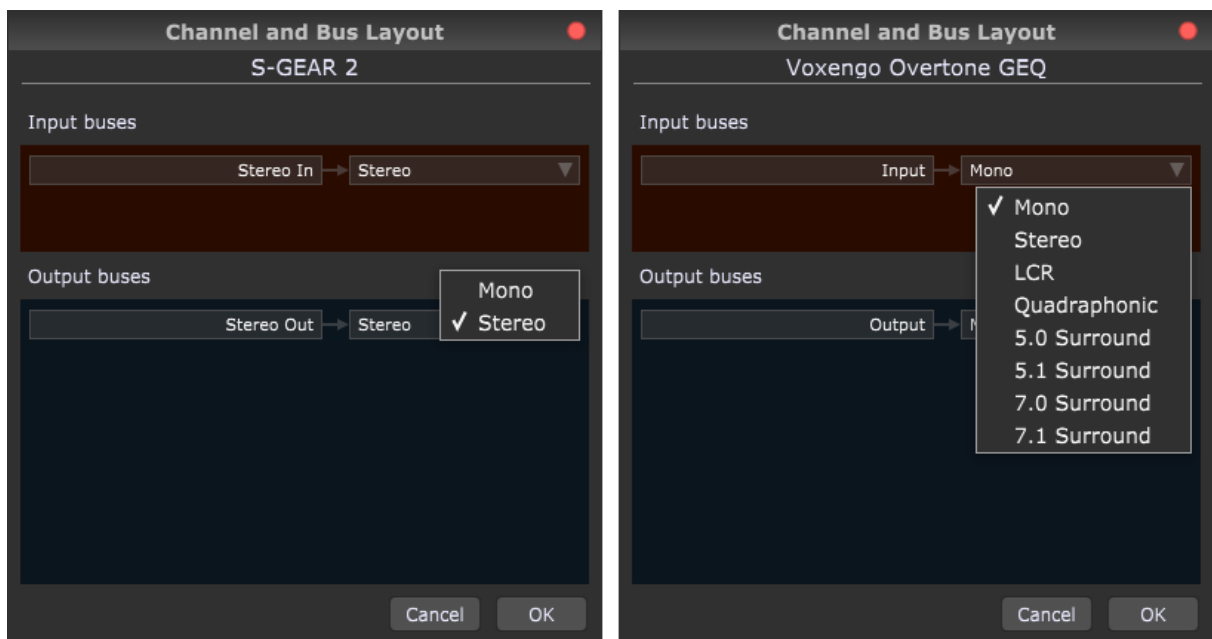
Clicking this menu item will extract the current block (Gain and Balance block) from the audio path, i.e. connect the KeysOfThe70s block with the To Global Rackspace block (see **(B)**).

12. **Replace Plugin** - choose a plugin from the menu that pops up to replace the current plugin with another.
13. **Quick Replace...** - opens a new window that allows you to find plugins quickly based on partial names:



Type in a search query, such as "gg" to quickly find all plugins with that name or their developer's name. You can also type in "synth" to display all your synth plugins.

**14. Bus Layout...** - opens the Channel and Bus Layout dialog:

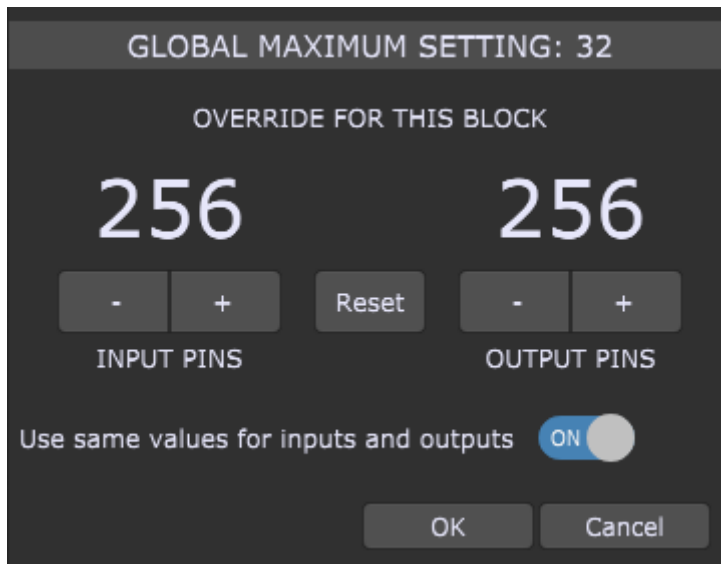


Some plugins support multiple configurations of inputs and outputs, usually mono and stereo, and sometimes multichannel. In a DAW environment, the DAW configures these input and output buses. In Gig Performer, you can access a plugin's bus options directly in the Channel and Bus Layout dialog.



Many plugins will simply show a stereo input/output bus, but some plugins, such as the eight-channel Voxengo Overtone GEQ, shown in the screenshot above, offer many more configurations.

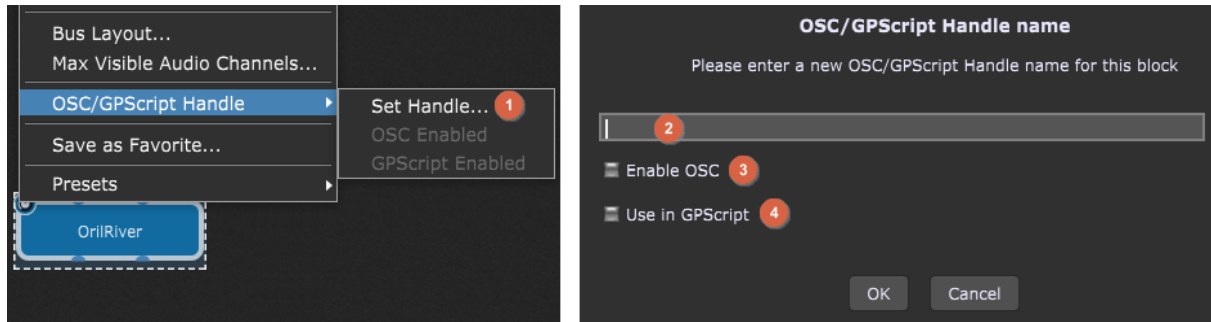
**15. Max visible audio channels...** - this feature allows you to set the maximum number of visible audio pins for the selected plugin block:



The global maximum setting (for all plugins) can be changed in [Display Options](#). This setting can be overridden for the selected block; the default current maximums for the override are 256 input/output pins, and their number can be set up independently if you turn off the *Use same values for inputs and outputs* toggle button. This feature is very useful for certain plugins that expose a large number of channels (64 or even 128), making the plugin block extremely wide. In such cases, you can use this option to force the plugin to display only the first few channels.

**Note:** this feature doesn't change the number of audio channels in the plugin; it just changes how many of them you can see. It was designed to prevent plugins (such as Kontakt) from displaying all 32 or 64 channels and taking up a lot of horizontal space in the Wiring view, when you may only need e.g. 2 or 4 channels. Therefore, this feature applies to channel visibility and does not add more channels than a plugin provides.

**16. OSC/GPScript Handle** - allows you to set a handle to be used by [GPScript](#) and [OSC](#), and to enable OSC and GPScript.



Click on the **Set Handle...** menu item (1); a new window appears, where you can type in a name of the handle (2), and then activate **Enable OSC** (3) and/or **Use in GPScript** (4) if you want to use these advanced features. To learn about handles in Gig Performer, see the blog article [What is a handle and how to use it](#).

**17. Save as Favorite...** - allows you to save a plugin as a favorite. Refer to the [Save one or more plugins as favorites](#) section for more information.

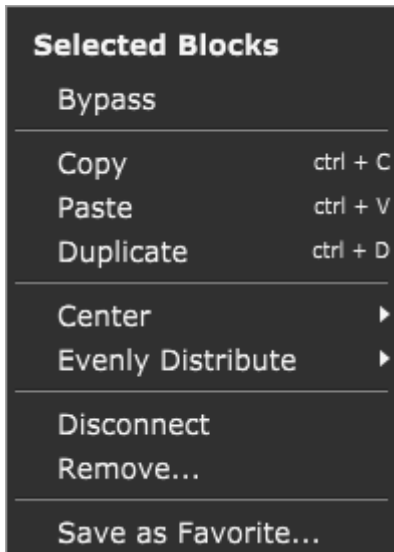
**18. Presets** - allows you to load the plugin's integrated (factory) presets or custom user presets. See the [Plugin Editor More menu](#) section to learn more.

**Note:** the MIDI In plugin block right-click menu is the same as other plugins, but with the addition of:

**Change MIDI Input Device** - reassigns the MIDI In or MIDI Out block to the MIDI controller chosen from the popup menu.

### 4.7.3 Selected blocks contextual menu

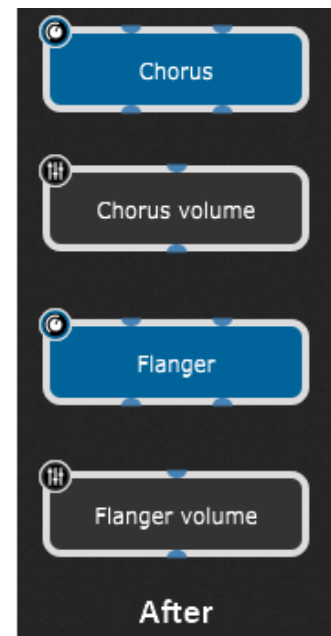
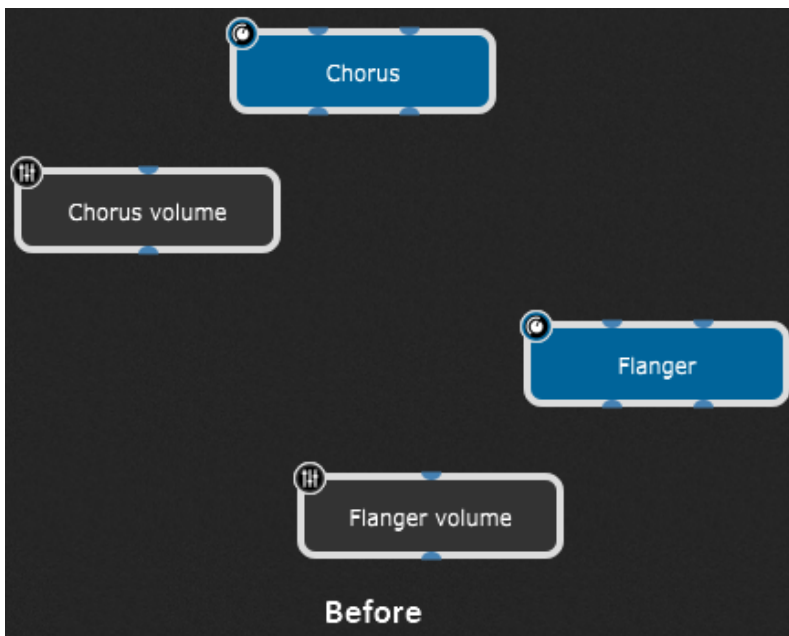
Suppose you select multiple plugins (either by using a selection rectangle or **Shift** + click on the desired plugin blocks) and right-click on any of the selected plugins. In that case, the *Selected blocks* contextual menu will open and allow you to do the following:



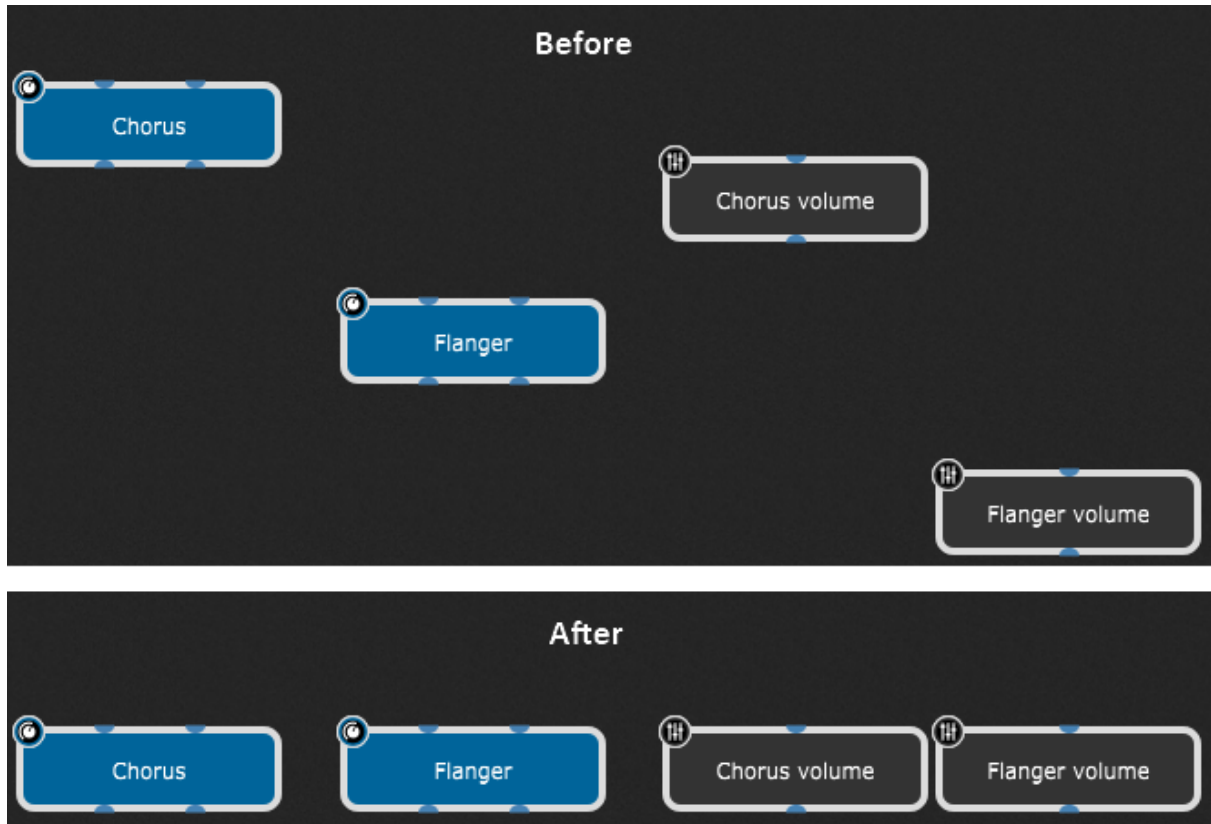
The displayed menu items have the same purpose as described in the [Plugin contextual menu](#) chapter; the new menu items are **Center** and **Evenly Distribute**.

When you hover over the **Center** menu item, two sub-items appear - Vertically and Horizontally - allowing you to center all selected plugin blocks vertically or horizontally.

**Center -> Vertically:**

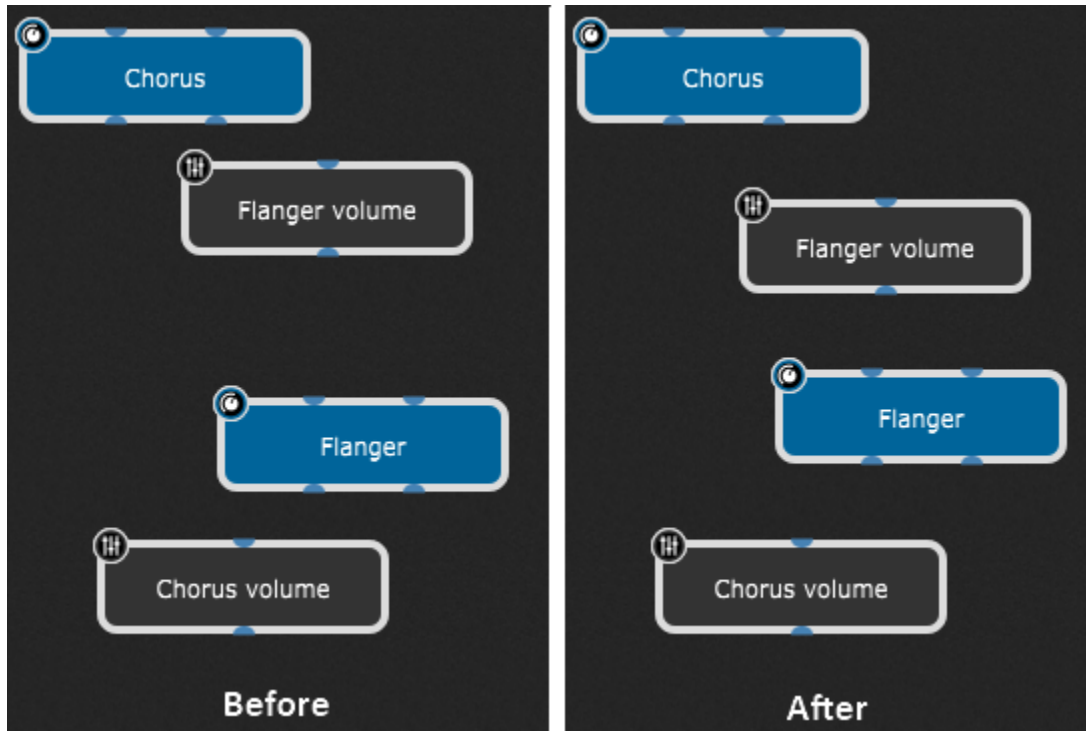


**Center -> Horizontally:**

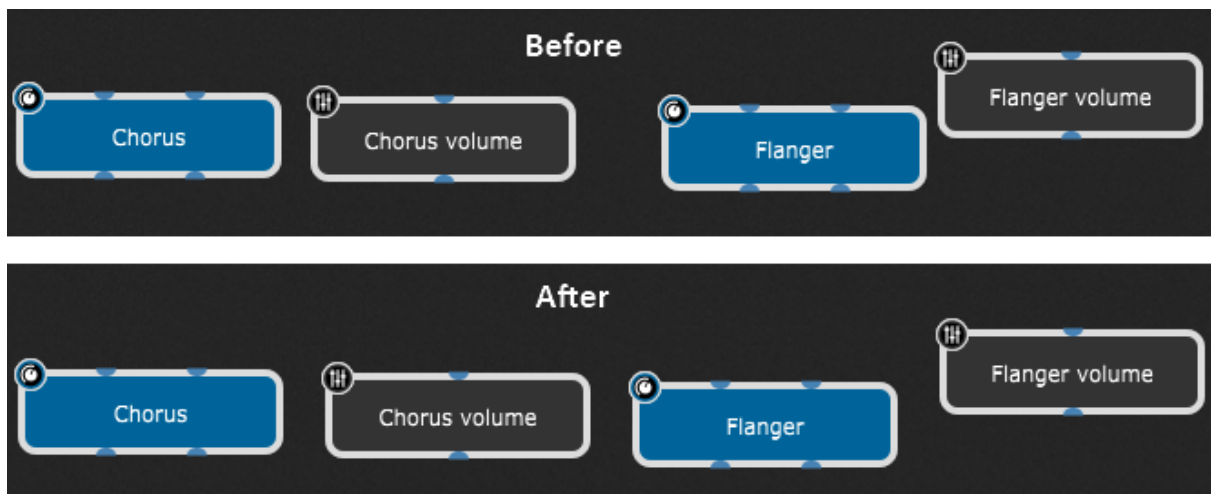


The **Evenly Distribute** menu item will be enabled if you select at least three plugin blocks; when you hover over the Evenly Distribute menu item, two sub-items appear - Vertically and Horizontally, allowing you to distribute all selected plugin blocks evenly, either vertically or horizontally respectively.

**Evenly Distribute -> Vertically:**



Evenly Distribute -> Horizontally:

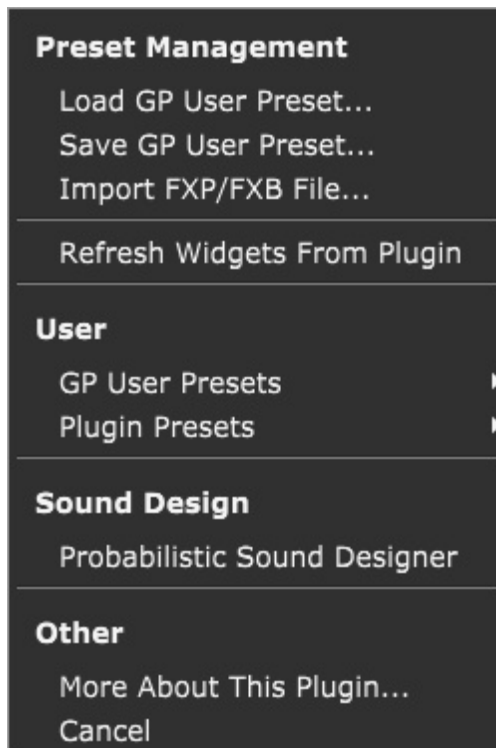


#### 4.7.4 Plugin Editor More menu

Double-clicking a plugin opens its plugin editor:



Clicking on the "..." button (indicated with the red arrow) opens the More menu:



**Load GP User Preset...** - loads a Gig Performer user preset file (gpp\_audiounit, gpp\_vst or gpp\_vst3).

**Save GP User Preset...** - saves a Gig Performer user preset file (gpp\_audiounit or gpp\_vst or gpp\_vst3, where the extension depends on the plugin type). User presets are saved in corresponding subfolders under the Presets folder based on their manufacturer name and the actual plugin name (refer to the [File locations in Gig Performer](#) chapter for more information). The resulting subfolder where user preset files are saved is in the following folder tree: Presets\Plugin\_Manufacturer\_Name\Plugin\_Name. **Tip:** to easily access your Presets folder, click Window -> Open Content Folders; open the Presets folder and browse its sub-folders (organized by plugin manufacturers' names) to find preset files and to perform other file operations on them (rename, delete, etc.).

**Import FXP/FXB File...** - imports an FXP/FXB preset file. FXP is an individual preset, FXB is a bank of presets. This feature is for legacy support and may disappear in a future version of Gig Performer. Not all VST plugins are compatible with the FXP/FXB format. Most of those that are compatible are audio plugins.

**Refresh Widgets From Plugin** - use this command to refresh all widgets that are mapped to this plugin. This feature is useful for a small number of plugins that do not notify their host when a new preset is loaded. If you notice that widgets in your rackspace are not updated when you change this plugin's preset, you can manually refresh widgets mapped to this plugin with this command. Check out [this blog article](#) to see this feature in action.

**GP User Presets** - this menu item is disabled if you have not previously saved any user presets for the current plugin. It will list the names of your saved user presets if it is available.

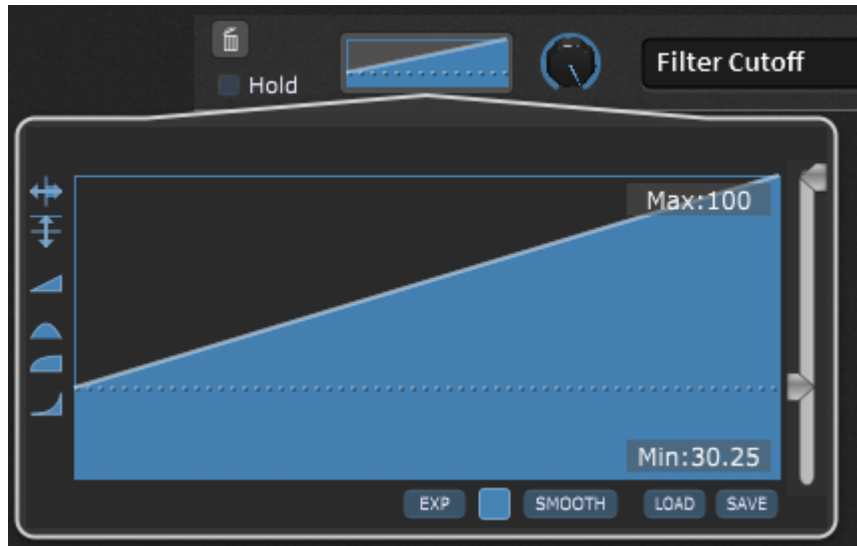
**Plugin Presets** - some plugins include predefined presets, most commonly factory presets. If a plugin has any such built-in presets, then they will show up in this sub-menu.

**Probabilistic Sound Designer** - randomizes selected plugin parameters values to generate random patches. This feature supersedes the Random Parameter Generator, an experimental feature in Gig Performer 3:



1. **Choose the desired parameters** - simply click on a desired control in the plugin editor window (i.e. Gain, Low, or other) and move it, and the corresponding parameter will be automatically recognized and captured (make sure that the **Capture Parameters** toggle button is turned on). The knob's positions in the plugin editor and the Probabilistic Sound Designer will be automatically synchronized.
2. **Randomize** - when you select the desired parameters, click the **Randomize** button to generate random values for the chosen parameters. Every time you click this button, it creates new values for the chosen parameters. **Note:** if the **Hold** check box is ticked, a random value for that parameter will not be generated (e.g. the last parameter in this list will not be randomized).
3. **Capture Parameters** - this toggle button is turned on by default; if you turn it off, new parameters will not be captured.
4. **Remove All** - removes all listed parameters.
5. **Load...** - clicking on this button allows you to load a previously saved Probabilistic Sound Designer configuration (gppsd\_audiounit, gppsd\_vst or gppsd\_vst3).
6. **Save...** - clicking on this button allows you to save the current Probabilistic Sound Designer configuration to a file (gppsd\_audiounit, gppsd\_vst or gpp\_vst3, where the actual extension depends on the plugin type). By default, these files are saved in corresponding subfolders in the PSD folder (refer to the [File locations in Gig Performer](#) chapter for more information).
7. **Remove button** - clicking this button will remove the current parameter from the list.
8. **Hold** - check this checkbox if you don't want the corresponding parameter to get a random value. In the screenshot above, all knobs will get the random value except the output volume.
9. **Curve** - allows you to control the range of possible values and the probability of particular values for the parameter. Refer to [this chapter](#) for more information on curves.  
For example, if you're randomizing a low-pass filter cutoff and there is no other modulator (e.g. no ADSR applied), then it's useful to prevent the filter from getting too low a value as that will just block all audio:





So the graph above says, produce any value equally likely from within the range of 30.25 to 100. The filter value, therefore, will never be set lower than about 1/3 from the bottom.

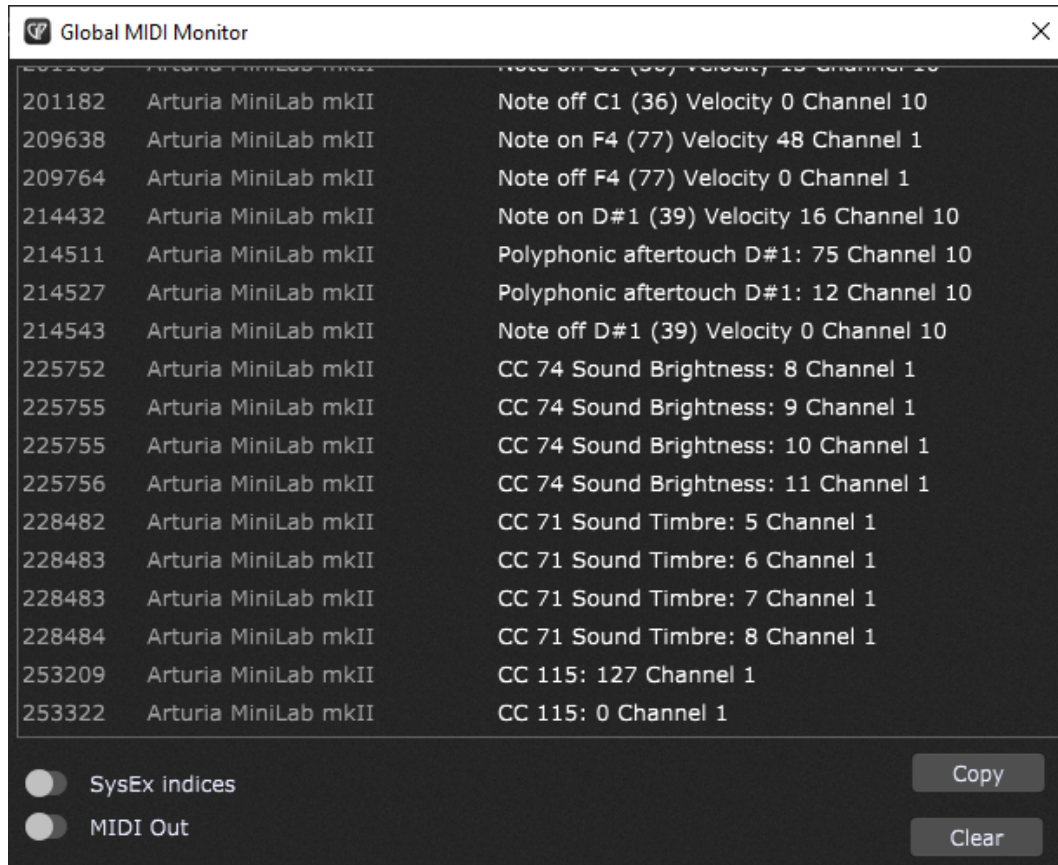
10. **Parameter name** - displays the name of the captured parameter.

**More about this plugin...** - opens the homepage of a plugin's manufacturer in the default Web browser.

**Cancel** - this menu item simply closes the menu.

## 4.8 Global MIDI Monitor

The global MIDI Monitor displays all incoming MIDI events. Click on the Window menu and select the Global MIDI Monitor to open it or press keys **Ctrl+M** (or **Cmd+M** on Mac):



This window displays all incoming MIDI messages received by Gig Performer (the first column represents time in milliseconds); you can optionally turn on the **SysEx indices** toggle button to see the position of each byte of a SysEx message or turn on the **MIDI Out** toggle button to display outgoing MIDI messages, as well.

Clicking the **Copy** button copies this window's contents to the clipboard. To clear the window, click on the **Clear** button.

You can set the colors for different MIDI messages in the [Display Options](#) window.

**Note:** also check the supplied [MIDI Monitor](#) plugin.

## 4.9 Rig Manager

In Gig Performer, a "Rig" is a collection of three sets of information: a set of **physical** MIDI controllers, a set of **virtual** controllers that can be attached to widgets, and a **map** that relates the two sets, thereby connecting your physical controller, through widgets, to parameter controls.

A MIDI Input Device is a physical controller, which may include a keyboard, multiple knobs, sliders, buttons, or other individual controllers. If the device does not include a keyboard, it is normally referred to as a control surface. In the Rig Manager, you define an alias for a MIDI device.

Aliases can also be set to MIDI Controls, representing the virtual controllers mapped to parameters.

Once the lists of MIDI Device Aliases and MIDI Control Aliases are established, they appear as drop-down menus in fields where they are used, for example, as MIDI sources in plugins (see the [How to use aliases](#) chapter for more).

The Rig Manager is designed to speed your workflow by eliminating the need to teach controllers to Gig Performer every time you use a different keyboard or controller setup (provided that you created aliases and saved your setup). **Note:** when you use a completely new MIDI device, you will need to map aliases to the desired controllers, but only once, since you can save them. Afterward, simply switch between the saved rigs.

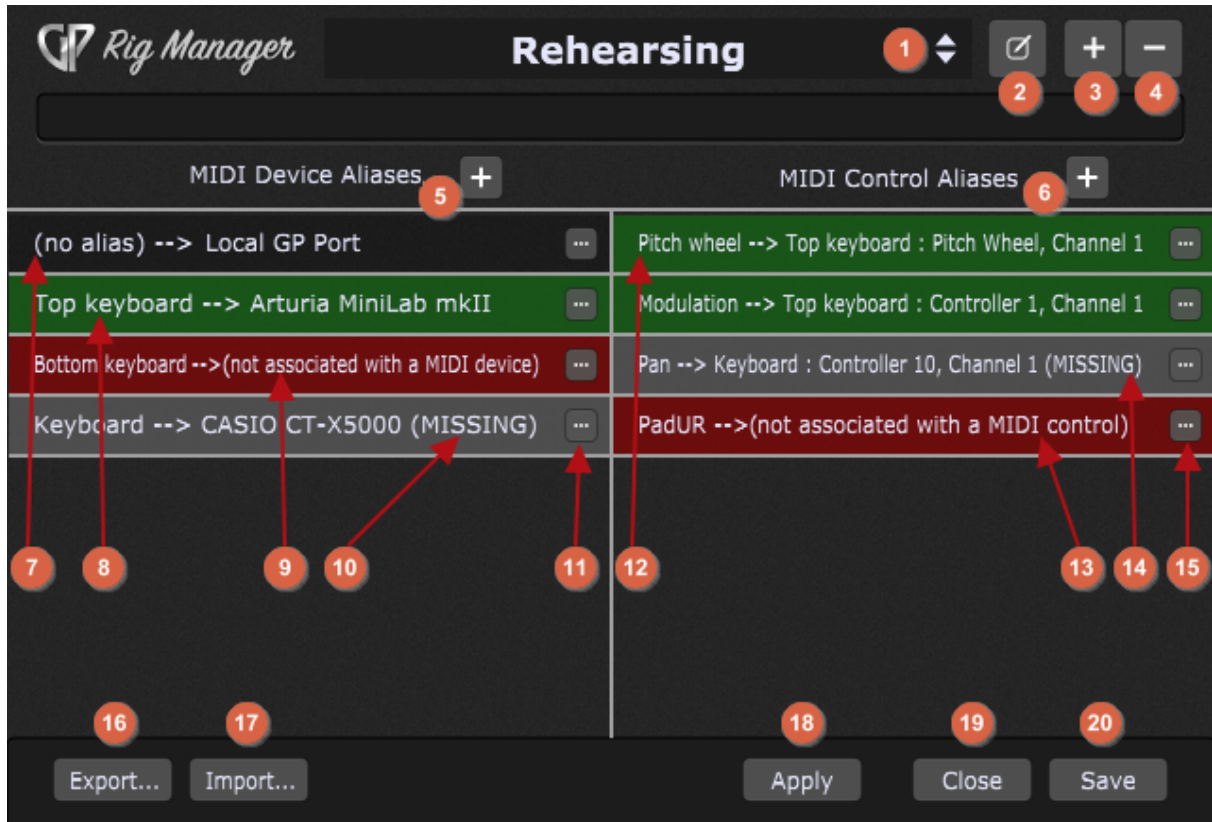
The Rig Manager constructs a map of physical input devices like keyboards and the controllers they host, like sliders, buttons, faders, and knobs, to aliases (virtual controllers) that are linked to widgets. Virtual controllers decouple the physical controller from the control paths in Gig Performer, which makes it easier to swap out different physical controllers while maintaining your usual controls.

Say, for example, that you're sharing backline instruments with another performer, and so have to use someone else's keyboard controller for your performance. The Rig Manager's controller mapping means that you only need to map the new physical controllers to the existing virtual ones and reestablish all of your widget controls. This can be accomplished in a few minutes, even at soundcheck. As an aside, this feature was implemented after one of the developers was on tour and ran into a problem when he had to use a different controller than normal.

Similarly, if you use one controller for rehearsal and a different one for gigs, the Rig Manager can make it much easier to switch between them.

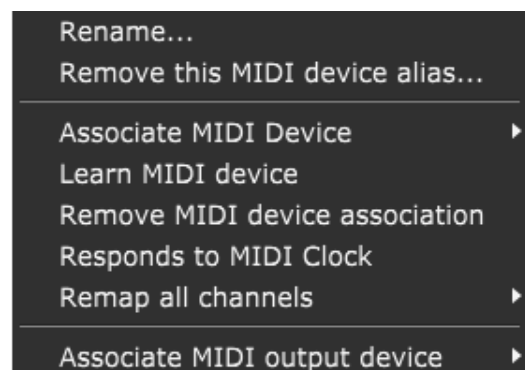
Also, if you play multiple identical keyboards, the Rig Manager can let you quickly designate which keyboard is which. For example, one of the developers often uses three identical controllers, named Top, Middle, and Bottom, in the Rig Manager. When the keyboards are plugged in, he can quickly visit the Rig Manager, and play a note on each of these keyboards so as to tell the Rig Manager which one is now Top, Middle or Bottom.

The Rig Manager window is displayed in the screenshot below:



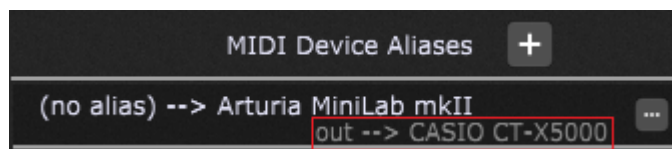
1. **Rig list** - clicking this drops down a menu of currently available rigs. Choose one from the list.
2. **Rename this rig** - click this to customize the name of the current rig.
3. **Add a new rig** - click this button to open a menu that allows you to duplicate the rig (the **Duplicate current Rig** option) or add a new one (the **Add new Rig** option). The difference between creating a new rig and duplicating the current rig can be seen on the left side of the Rig manager - MIDI Device Aliases; when a new rig is created, aliases are not associated with MIDI devices and when the current rig is duplicated - aliases are associated with MIDI devices (the same as in the original rig). **Note:** MIDI Control aliases are defined globally and are present in newly created rigs.
4. **Remove this rig** - click this button to remove the current rig. If you change your mind, a confirmation dialog offers you the chance to cancel.
5. **Add a new MIDI device alias** - click this to add a new MIDI device alias in the list. When you add a new MIDI device alias, it is not associated with a MIDI device (like in **(9)**). See the point **(11)** to see how to associate it. After you successfully associate the MIDI device alias with the desired MIDI device, it is highlighted green like in **(8)**.

6. **Add a new MIDI control alias** - click this to add a new MIDI control alias to the list. When you add a new MIDI control alias, it is not associated with a MIDI control (like in **(13)**). See the point **(15)** to see how to associate it. After you successfully associate the MIDI control alias with the desired MIDI control, it is highlighted green like in **(12)**.
7. **MIDI Input Device with no defined alias** - indicates that Gig Performer recognized this MIDI input device, but no MIDI device alias has been defined for it yet. Gig Performer automatically detects all available MIDI input devices. **Note:** the Local GP port is a special MIDI device, so check the [Local GP port](#) chapter to learn more.
8. **Associated MIDI Input Device alias** - indicates that the MIDI Input Device alias is successfully associated with this MIDI device and that this MIDI device is connected to the computer.
9. **Unassociated MIDI Device alias** - indicates that the MIDI Device alias is created, but it is not associated with a MIDI device.
10. **Missing MIDI device** - indicates that this MIDI device is associated with the corresponding alias but is disconnected. Connecting the corresponding MIDI device will change the display to green, as indicated in **(8)**.
11. **Manage this entry button** - clicking on this button (or right-clicking on any MIDI device alias) opens the contextual menu, where its menu items depend on whether an alias is undefined (screenshot to the left) or already defined (screenshot to the right):



- a. **Define alias for this MIDI device...** - clicking on this menu item opens the Define MIDI device alias dialog, which allows you to enter a new alias for the corresponding MIDI device.
- b. **Rename...** - allows you to rename the MIDI device alias.

- c. **Remove this MIDI device alias...** - removes the MIDI device alias. The MIDI device will display "(no alias)" when connected (as shown in **(7)**).
- d. **Associate MIDI Device** - allows you to manually select a MIDI device from the list and associate it with the MIDI device alias.
- e. **Learn MIDI device** - allows you to quickly assign a MIDI device by simply moving the desired control. The MIDI device alias will be automatically mapped to the detected MIDI device.
- f. **Remove MIDI device association** - clears the MIDI device association for the MIDI device alias.
- g. **Responds to MIDI Clock** - sets the MIDI device associated with this alias to respond to MIDI Clock messages. (if you set a MIDI device to respond to MIDI Clock messages, don't forget to save the changes to the Rig Manager configuration) To check what other MIDI devices respond to MIDI Clock messages, go to the [MIDI Ports Options](#) and check where the **Rx Clock** button is lit.
- h. **Remap all channels** - allows you to remap all channels of the MIDI device under this alias to the desired channel from 1 to 16.
- i. **Associate MIDI output device** - allows you to select the desired MIDI output device from the list:



This is used when the **Sync** option is enabled for a widget in the [Widget Properties Inspector](#) controlled by a MIDI event from an Input device. The **Sync** by default goes back to the same MIDI device, which is what you would want in most cases. Still, sometimes you may want to Sync or "send out" those MIDI messages to a different MIDI device so it can display the values (as indicated with the red rectangle in the screenshot above). **Note:** this is a widget option and not a permanent one; if you don't enable the Sync option for a widget, then selecting a different Out device has zero impact on your setup.

12. **Associated MIDI Control alias** - green background indicates that the MIDI control alias is associated with the physical MIDI control.

13. **Unassociated MIDI Control alias** - indicates that the MIDI Control alias is created, but it is not associated with a MIDI control.
14. **Missing MIDI device** - indicates that this MIDI control is associated with the corresponding alias, but the MIDI device that hosts this control is disconnected. Connecting the corresponding MIDI device will change the display to green, as indicated in (12).
15. **Manage this entry button** - clicking on this button (or right-clicking on any MIDI control alias) opens this menu:



- a. **Rename...** - allows you to change the name of the MIDI control alias.
- b. **Remove this MIDI control alias...** - removing the MIDI control alias also removes this entry from the MIDI Control Aliases list.
- c. **Control mode** - allows you to select between an absolute mode and one of four relative modes:
  - **Absolute** - use this option if you don't have endless encoders (default). An endless encoder doesn't have a "stop" point, so it keeps turning until you stop moving it.
  - **Relative Offset** - positive offsets are sent as offset plus 64, and negative offsets as 64 minus offset. 64 is zero, 65 is +1, 63 is -1.
  - **Signed 1 Bit** - if the most sign bit is set, Then the offset is positive. The lower 6 significant bits are the offset.
  - **Signed Magnitude** - if the most sign bit is unset, Then the offset is positive. The lower 6 significant bits are the offset.
  - **Two's Complement** - positive offsets are sent as normal from 1 to 64, and negative offsets are sent as 2s complement negative numbers. This is a signed 7-bit int.
- d. **Polarity** - allows you to reverse the polarity of a MIDI controller, such as a sustain pedal. When you reverse the polarity for a control, navigate to the

[Widget Properties Inspector](#) and in the MIDI tab and refer to the control by its alias name.

- e. **Learn MIDI control** - allows you to quickly assign the physical control on a MIDI device by simply moving the desired control. The MIDI control alias will be automatically mapped to the detected control.
  - f. **Remove MIDI control association** - removes the association with the physical MIDI control.
16. **Export...** - click this button to save the state of the rig to a *rigsetup* file. This file can be moved and imported to a new computer.
  17. **Import...** - click this button to load an existing rig file. **Note:** changes will not take effect until you click Apply or Save.
  18. **Apply** - click this button to apply the changes you have made.
  19. **Close** - clicking on this button closes the Rig Manager window.
  20. **Save** - click this button to save the state of the rig to the current rig file and close the Rig Manager window.

**Note:** the Rig Manager associations are not part of the gig file. If you are moving your hardware from one computer to another, you should export your rig from the computer where things work and import it from other computers where you plan to use that hardware. Also, refer to [this chapter](#) if you want to export all the options for Gig Performer.

## 4.10 ChordPro

ChordPro is a file format that allows you to represent lyrics and chords using text markup that can then be displayed in a form suitable for performing musicians. Although initially intended for guitarists, it can be used for all kinds of musical purposes.

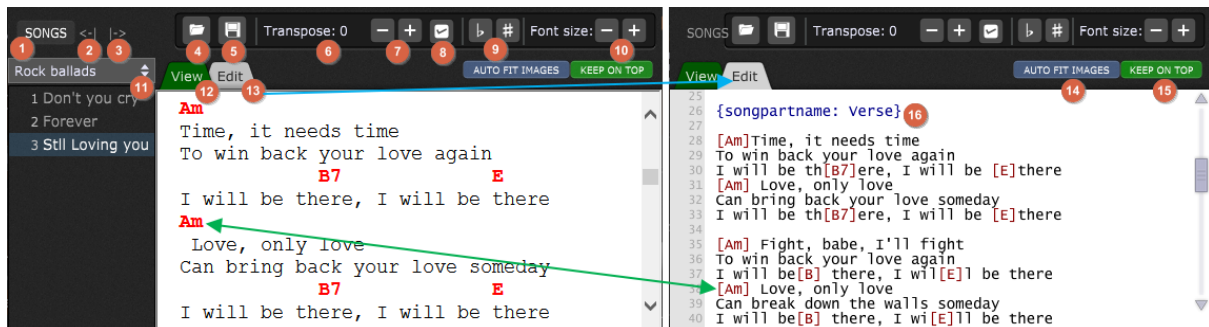
Gig Performer supports the ChordPro lead sheet file format, and ChordPro files can be created, edited, saved, and imported into Gig Performer. A ChordPro file can be linked to a song in the Song Lyrics/Chords Editor. The assigned ChordPro file can be seen in the [Song Properties](#) dialog (it also allows you to clear the association to the ChordPro file).

For more information on ChordPro, please visit [www.chordpro.org](http://www.chordpro.org). For more information on its file format specification, [click here](#).



**Note:** any text editor can be used to create and maintain ChordPro files. These files may have arbitrary types (extensions), and common file name extensions for ChordPro files are **.cho**, **.crd**, **.chopro**, **.chord** and **.pro**. Gig Performer will also list and import the most common **.txt** file extension, so you don't have to change the extension of your ChordPro files that you save with your editor.

In Gig Performer, ChordPro files are created and modified in the **Song Lyrics/Chords Editor**:



1. **Song list button** - click this to show the current setlist (indicated with **(11)**) and its songs. When the song list button is clicked, a song browser will appear to allow you to select the song for which you want to work on a ChordPro document. Click the Songs button again to close the song list.
2. **Make the song list narrower button** - allows you to reduce the width of the song browser pane.
3. **Make the song list wider button** - allows you to widen the song browser pane.
4. **Load from a file button** - clicking this button brings up an Open dialog for loading Chord Pro files. **Note:** files can only be opened if they are in the root folder named **Gig Performer Song Lyrics-Chords** by default; you can specify a different root folder in the [Locations Options](#). Refer to the [File locations in Gig Performer](#) chapter to see the default location for this folder.
5. **Save to a file button** - saves the current contents of the editor to a ChordPro file. Files can be saved anywhere, but can only be opened directly from the root directory.
6. **Transpose amount** - this display shows the current transposition in semitones. It is not editable. Transposition only affects the display of chord names in the ChordPro document.

7. **Increase/decrease transposition amount buttons** - use these buttons to set the transposition amount for chord display in the ChordPro document.
8. **Save transposition button** - clicking this button commits the transposition and makes it permanent in the ChordPro file.
9. **Make all chords sharp/flat buttons** - when there are chords whose root contains an accidental, choosing one of these commands causes those chords to be displayed enharmonically.

Original

**Am**  
I don't need all of this squawking and squabbling,  
**Bbm**                      **Em**  
And I haven't got the time,

-----

Make all chords sharp

**Am**  
I don't need all of this squawking and squabbling,  
**A#m**                      **Em**  
And I haven't got the time,

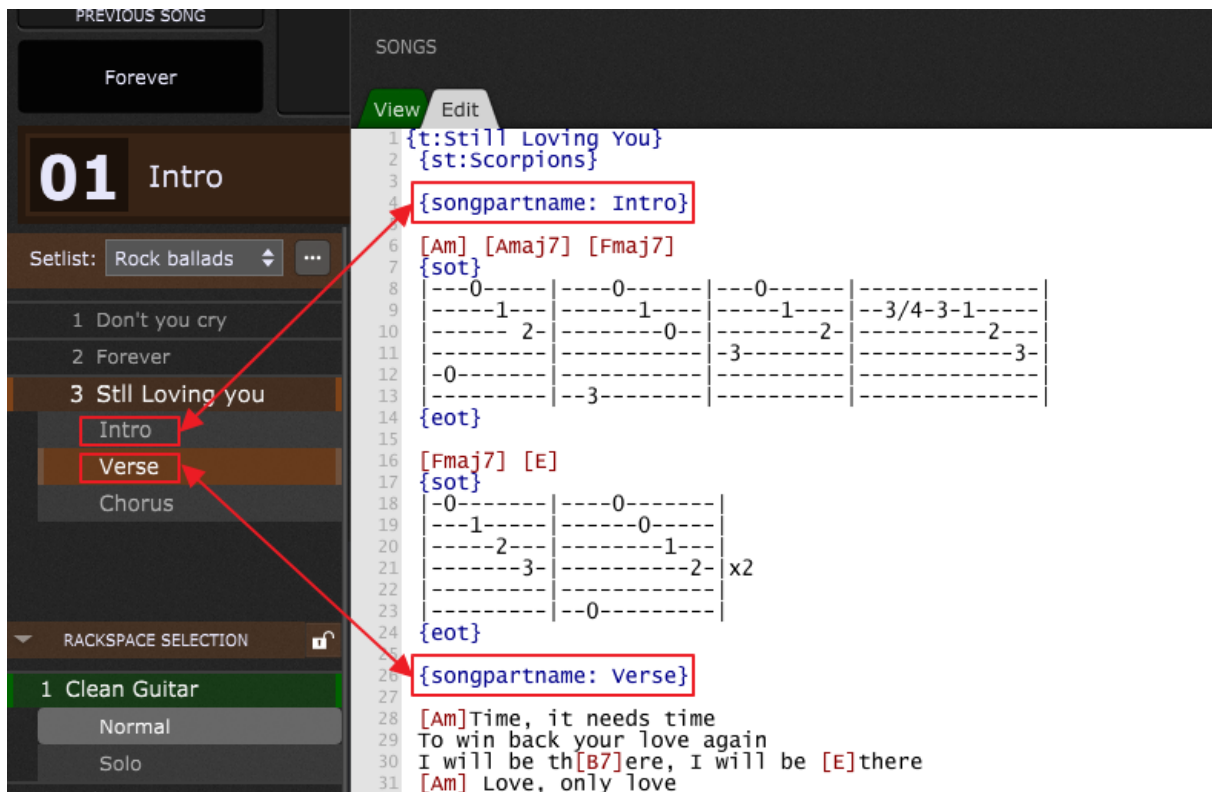
10. **Font size larger/smaller buttons** - these buttons enlarge or shrink the font size on everything in the View mode. Clicking this button in Edit mode will switch you to View mode, where you can size the font.
11. **Setlist chooser** - a drop-down menu for choosing a setlist. All the songs in the active setlist are included below. If one is assigned, clicking on a song displays the linked ChordPro file.
12. **View tab** - this tab enters the View mode, which interprets the code from the Edit mode and displays the result.
13. **Edit tab** - this tab (indicated with the blue arrow) enters the Edit mode, where ChordPro coding is done.
14. **Auto Fit Images** - clicking this button automatically fits images to the dimensions of the Song Lyrics/Chords Editor window.
15. **Keep on top button** - when this button is clicked the Song Lyrics-Chords Editor window is always in the foreground and cannot be obscured by another window. When engaged, the button turns green.

16. **Editing area** - a simple text editor for entering lyrics, chords, and ChordPro directives (read more about ChordPro directives [here](#)). The chords representation in the Edit and View modes is indicated with the green arrow. **Tip:** to place a chord over the middle of a word, simply insert the chord in its brackets in the middle of the word, immediately in front of the letter you want the chord to appear over, with no spaces. For example, see the word "th[B7]ere" in the first screenshot.

A special directive named **songpartname** is used to link a part in the ChordPro file with a corresponding song part.

**Note:** a ChordPro file can be loaded into the song lyrics/chords editor ONLY from the root directory; to load a ChordPro file not made by Gig Performer or not stored in the root directory, use the **File -> Import ChordPro files...** command.

The Song Lyrics/Chords Editor supports automatic scrolling to song parts:

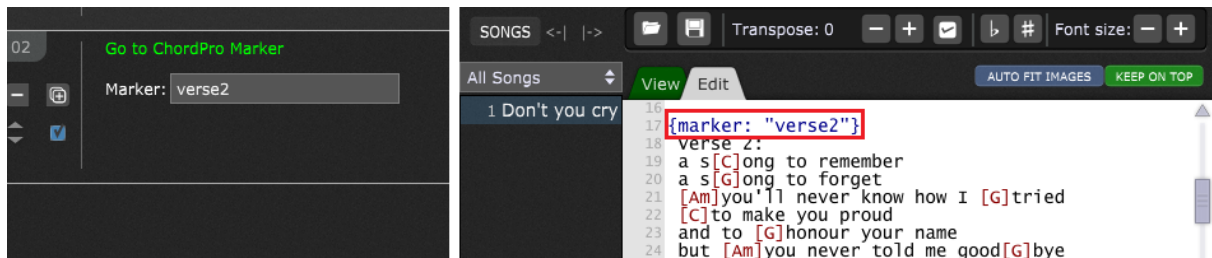


For example, if you have a song with three song parts - Intro, Verse, and Chorus - as in the screenshot above, simply open the Song Lyrics/Chords Editor, click on the Edit tab and enter the corresponding directives:

- for the Intro song part, enter `{songpartname: Intro}`
- for the Verse song part, enter `{songpartname: Verse}`
- for the Chorus song part, enter `{songpartname: Chorus}`

Next, click on the View tab and click on one of the song parts - you will notice that your lyrics file automatically scrolls to that part.

The Song Lyrics/Chords Editor also supports the **marker** directive. For example, if you type in `{marker: "verse2"}` as indicated with the red rectangle in the screenshot below:



then you can automatically scroll to that marker from the [Action Management](#) window. Simply add a *Go to ChordPro Marker* action and enter the marker name without quotes. When the playhead reaches this marker, it will move the ChordPro file to the specified position (in this example, to show Verse 2).

**Note:** if you don't have a ChordPro file linked to your song, you have to find it or create your own ChordPro file.

There are many chord-oriented websites for musicians that support ChordPro files. Refer to this blog article, [How to use ChordPro files in Gig Performer](#) for more information on obtaining ChordPro files from these websites or creating your own ChordPro files and how to use them in Gig Performer.

For PDF support, please visit [this chapter](#) to learn about the free PDFToGPChordPro utility to make your PDF usable in ChordPro.

**Tip:** To learn how you can automatically scroll your lyrics, refer to the blog article [Scrolling your lyrics or lead sheets](#).

#### 4.10.1 PDFToGPChordPro

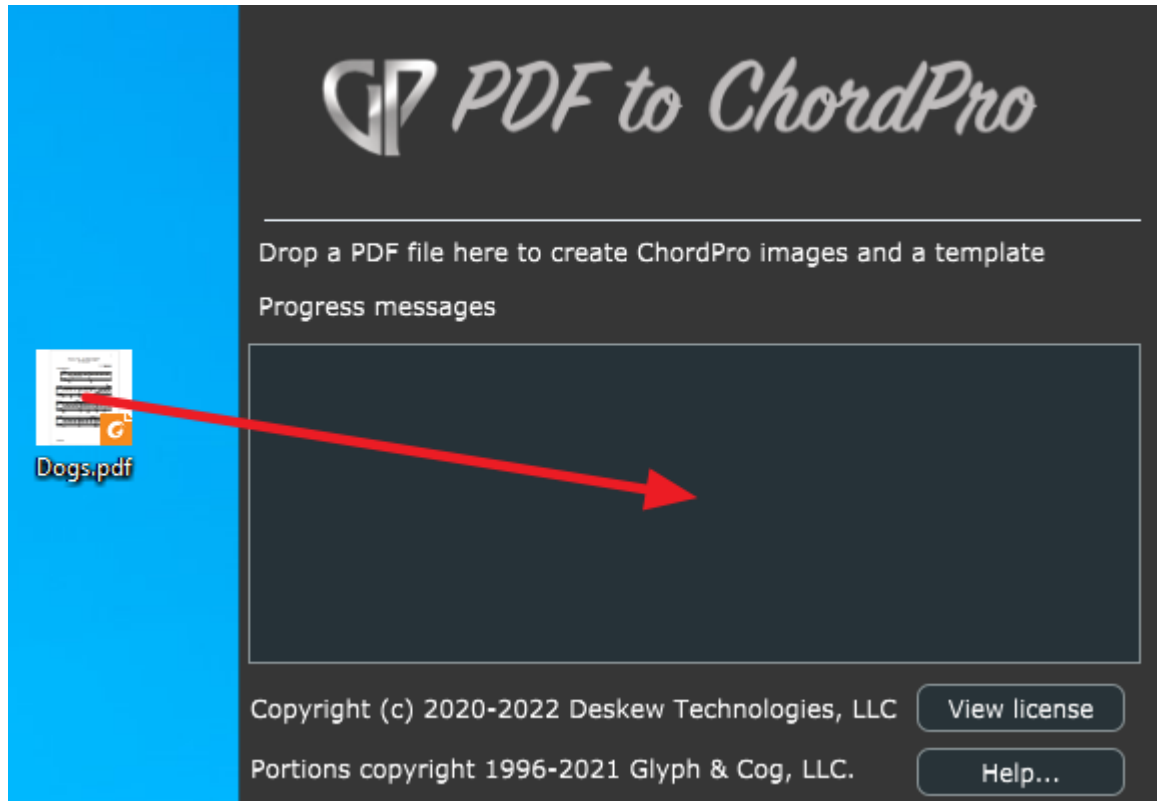
PDFToGPChordPro is a free tool that will convert PDF files into a format suitable for use within Gig Performer's ChordPro.



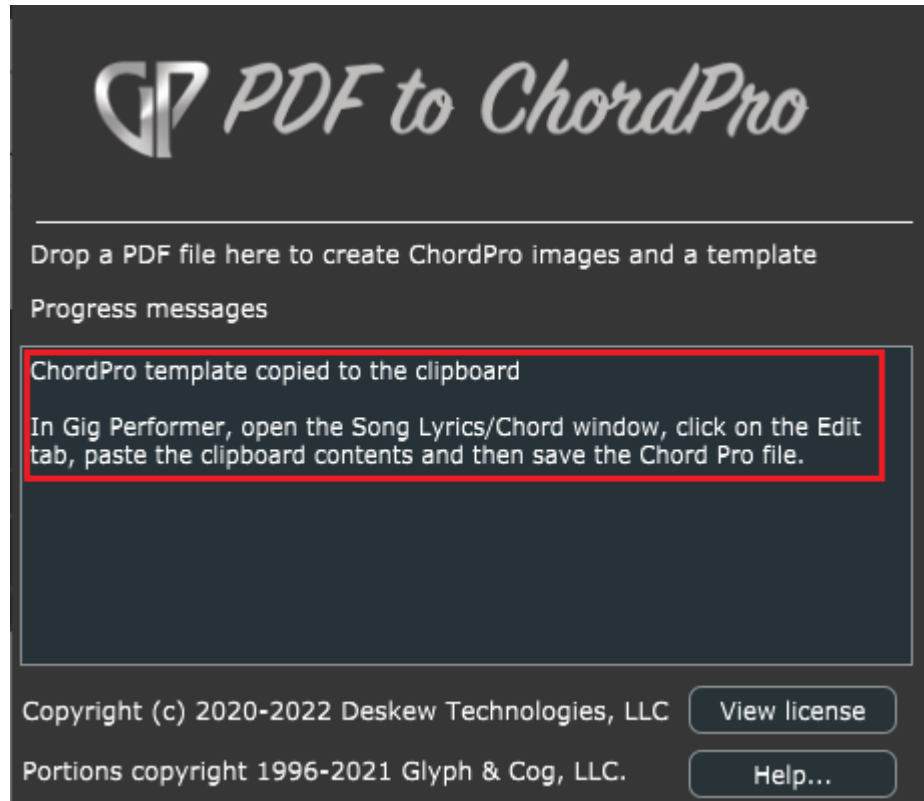
**Disclaimer:** there is absolutely no warranty whatsoever for this program. Use it completely at your own risk. You agree to indemnify Deskew Technologies for any consequences arising out of your use of this free tool.

First, download PDFToGPChordPro: [Mac version](#) or [Windows version](#). Extract the ZIP archive where desired (Mac users may prefer to put it in their Applications folder, and Windows users might prefer to put it in their C:\Program Files folder) and start the application.

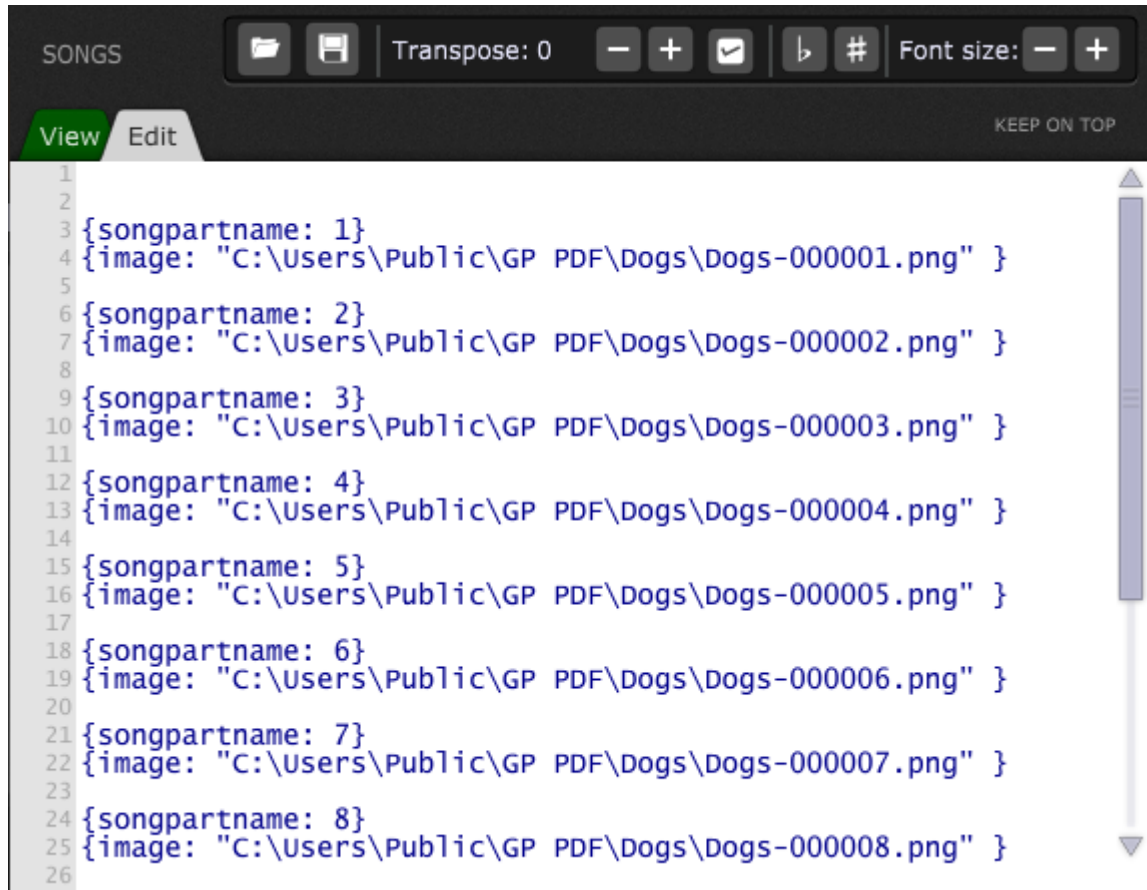
Drag the desired PDF file and drop it into the applications's window:



The conversion process takes place automatically. Upon completion, the ChordPro directives needed to display the converted PDF file are automatically copied to your clipboard. Wait until you see this progress message:



Click on the **Window** menu and select **Song Lyrics/Chords...** to open the Song Lyrics/Chord editor window, switch to the **Edit** tab and paste the clipboard contents:



Click on the **View** tab to see the result:



SONGS [Icons] Transpose: 0 [Icons] Font size: [Icons] KEEP ON TOP

View Edit

### Dogs 2011 Remaster

**Allegro** ♩ = 118

**Weighted:** Synth 1

9 Synth 1

16 Synth 1  
*You gotta be crazy, you gotta have a real need* *You gotta sleep on your toes, and when you're on the street*

23 Synth 1  
*You gotta be able to pick out the easy meat with your eyes closed* *You gotta strike when the moment is right without thinking.* *And after a while, you can work on points for style*

Org. **Bottom:** Auto sustain

34 Synth 1  
*Like the club tie, and the firm handshake*

**Bottom:** Org.

40 **Drawbars: Middle** Org. *have to be trusted by the people that you lie to* *So that*

2 45 *when they turn their backs on you* *You'll get the chance to put the knife in.* **Instrumental**

50 Org.

The last step is to [save your ChordPro file](#).

**Note:** if you examine the ChordPro directives that were pasted into the Edit tab, you'll notice *songpartname* directives with numbers as arguments, as already mentioned in [this chapter](#). Therefore, unless you name your song parts 1, 2, 3 ... you will want to change the entries for each *songpartname* directive in the ChordPro text file to match whatever song part names you have used in your song to enable automatic lyrics scrolling to the corresponding song parts.

Of course, you can insert other ChordPro directives if you need them.

Make sure to check our blog article, [Converting PDF files for use with Gig Performer ChordPro](#).

**Note:** PDFToGPChordPro is built on top of the open-source application [XpdfReader](#).

**Tip:** you can find this tool and other additional tools that we developed on our [Additional Downloads](#) page.

## 5 Inserting and Wiring your plugins

In this chapter, find out how to insert and connect your audio inputs to your plugins, how to connect plugins, and how to connect them to audio outputs.

Gig Performer supports 64-bit VST, VST3 and AU (Mac only) audio plugins. Before you proceed, make sure that you have installed the 64-bit version of your audio plugins and that plugin directories are properly specified in the [Plugin Manager](#).

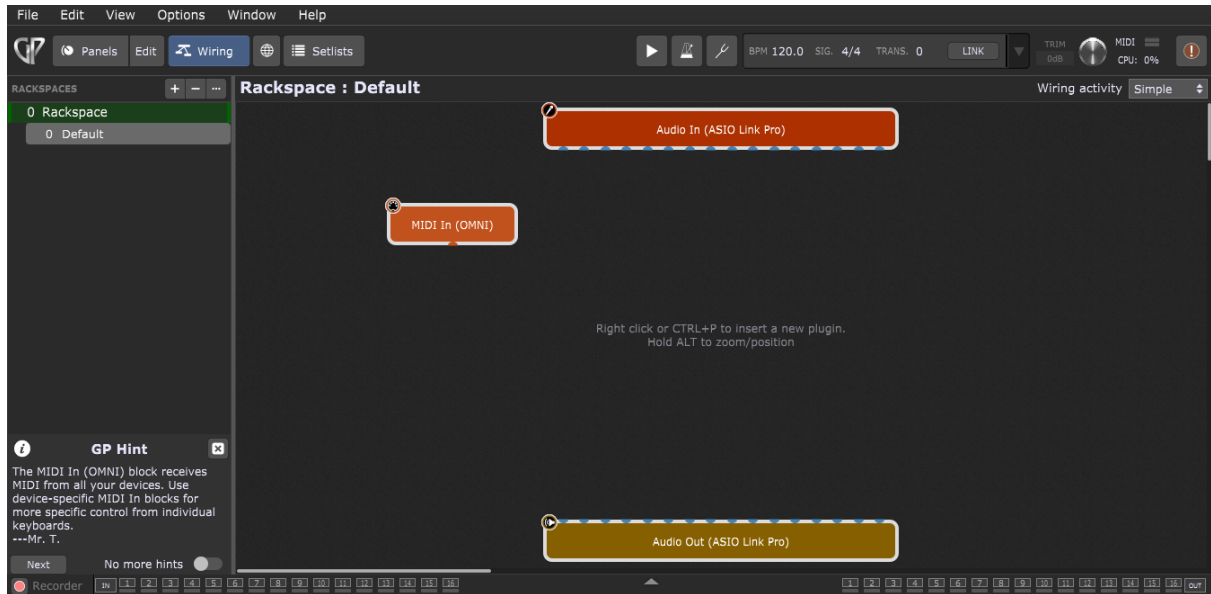
For more information on downloading and using plugins in Gig Performer, check this blog article: [Top websites for free audio plugins and how to use them in Gig Performer](#).

**Note:** make sure that your audio interface is connected. Gig Performer will try to use your computer's built-in input and output if no audio interface is connected. See the [Troubleshooting](#) section in this user guide if an audio interface is connected but does not appear.

### 5.1 How to insert a plugin and connect it

Easy combining, inserting and tweaking your plugins is essential for any serious musician. In Gig Performer, plugins are connected easily using virtual wires. First, make sure that Gig Performer successfully scanned and listed your installed plugins as already advised in the [Inserting and Wiring your plugins](#) chapter.

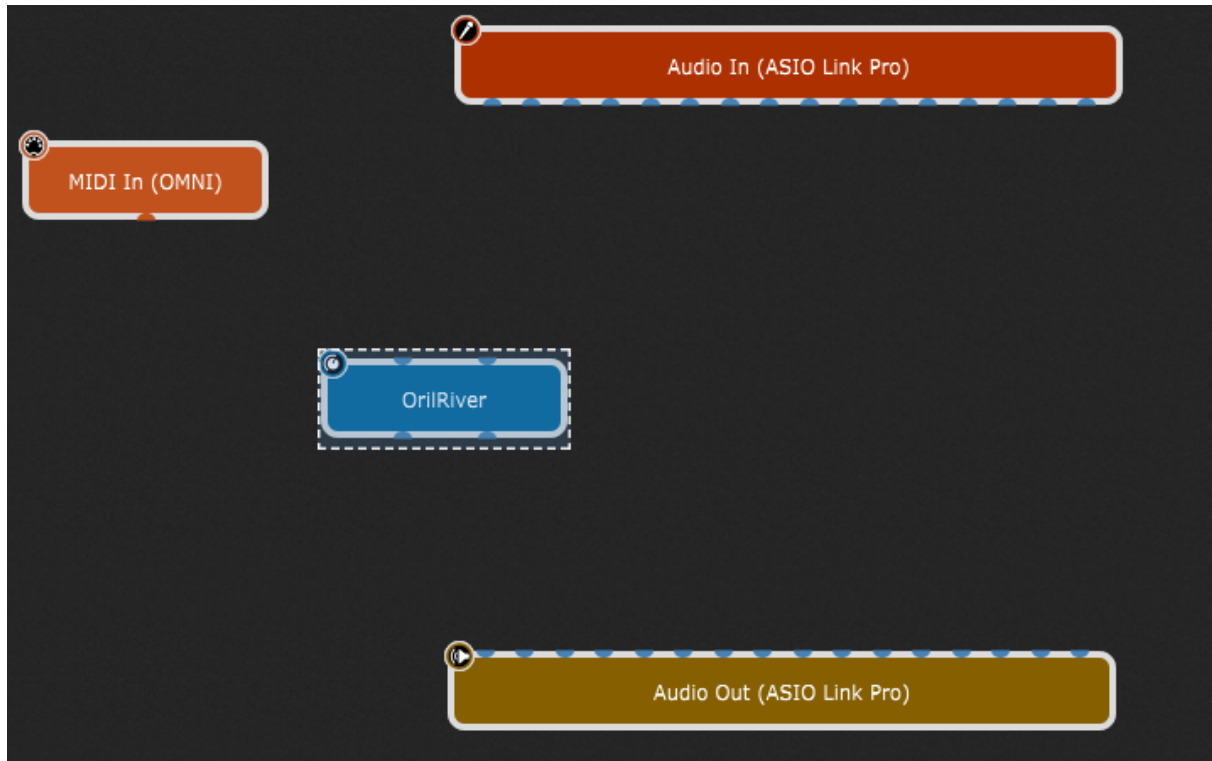
Let's get started! Create a new empty gig and enter the [Wiring view](#) by clicking on the Wiring button:



Various elements in this screenshot are explained in detail in the [Wiring view](#) chapter.

**Note:** your screen may look slightly different from the one shown above, since the audio input and output blocks will reflect the name of your connected audio interface as well as the number of inputs and outputs it provides.

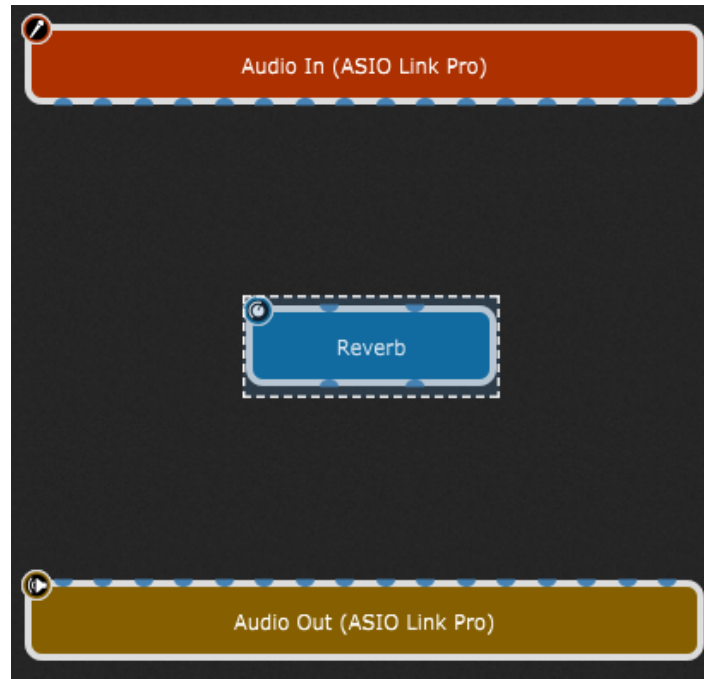
Right-click (or two-finger tap if you are using a trackpad) on the empty surface to bring up the [contextual menu](#) and select the desired plugin under the Installed plugins section, e.g. a reverb plugin:



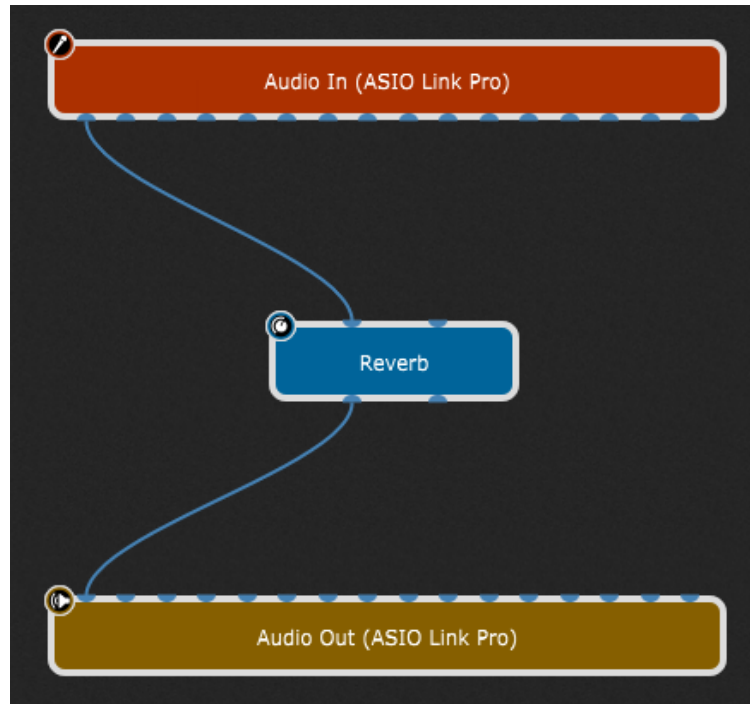
Alternatively, use the keyboard shortcut **Ctrl+P** (or **Cmd+P** on Mac) to open the [Quick plugin, favorite or preset finder](#) window to search for a particular plugin by part or all of its name and insert it.

As soon as you insert the desired plugin block (or double-click on it), Gig Performer, by default opens the editor window for that plugin, allowing you to edit its parameters immediately (note: you can turn off this default behavior in the [Display Options](#)). Make the desired adjustments, or select a preset, and close the window.

Afterward, simply click on the desired plugin block and drag it to change its position on the design surface. By default, the plugin's name is displayed. Still, you can easily rename it with something that represents the actual sound, which is very useful for distinguishing among multiple copies of the same plugin. Accordingly, right-click on the newly inserted plugin block and select the [Rename...](#) option, then type in the name, i.e., "Reverb":



To hear the resulting sound, we need to provide an audio source. If you are a guitarist or vocalist, you might connect an [Audio In](#) pin directly to the Reverb input pin and a Reverb output pin to the [Audio Out](#) block. See the next section if you are a keyboard player using a synth plugin. To make these connections, simply click on an output pin and, while holding down the mouse button (or keeping your finger on the trackpad), drag the cursor over the desired input pin and then release the mouse button (click [here](#) for a shortcut method of making stereo connections):

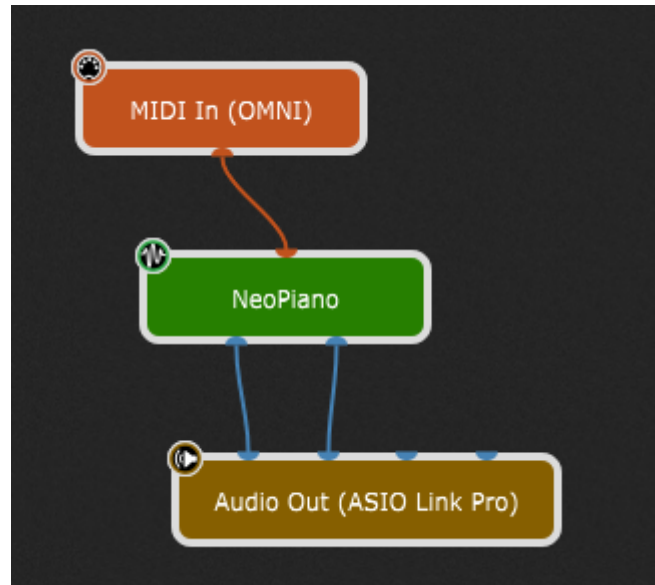


**Note:** make sure you connect the right Audio In pin, as your audio interface has multiple audio inputs where sources, i.e. microphones, instruments connected via DI (such as electric guitars), or other line-level sources can be connected.

Notice that there are blue and orange plugin pins and connections; [click here](#) to find out more.

If you use a MIDI keyboard or a MIDI controller, you can use it immediately thanks to the [MIDI In \(OMNI\)](#) block, which is available by default when creating a new empty gig (see the first screenshot). This block receives MIDI messages coming from any MIDI Input device on your system.

Note that you cannot hear the resulting sound immediately, since this block needs to be connected to the sound generating block first, and then to the Audio Out block:

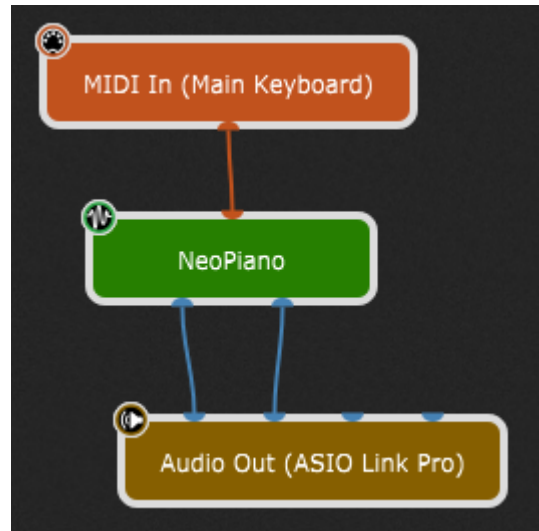


If you play a few notes on your controller (assuming your audio system is turned up), you hear a sound coming from your connected audio interface.

While the OMNI version of the MIDI In block is convenient when just getting started, if you have more than one keyboard or multiple surfaces, we recommend that you use MIDI In blocks specifically associated with those devices rather than the generic OMNI block as this will give you more control over your MIDI signal flow.

To receive MIDI messages from a specific MIDI device, right-click on the MIDI In (OMNI) block, expand **Replace Plugin**, and select the desired MIDI device from the **MIDI Inputs** list. All connected MIDI devices are listed in this menu. Alternatively, select the MIDI In (OMNI) block and press the **Delete** key (or the **Backspace** key) to remove the MIDI In (OMNI) block; afterward, open the [contextual menu](#), insert the desired MIDI device from the [MIDI Inputs](#) list and connect its output pin to the orange input pin of your sound-generating plugin:





**Note:** if you previously created an alias for your MIDI device in the [Rig Manager](#), it will show up in the parenthesis (otherwise, the name of the selected MIDI device will be displayed).

In addition to passing MIDI messages generated by your connected MIDI controller, MIDI In plugins provide a virtual keyboard, so you can play your plugins even if you are not using an external MIDI controller. To do so, double-click the MIDI In block to open its editor, then play the sound by clicking notes on the virtual keyboard.

To remove (disconnect) a connection, right-click on it and select "Remove this connection", or simply drag it until it detaches from a plugin and then let go. Alternatively, hold down the **Backspace** key and move your mouse over the wire to remove it.

To remove all connections from a plugin block, right-click on the block and select "Disconnect" (note: removing a plugin block also removes all its connections; right-click on it and select "Remove" or press either the **Delete** or **Backspace** key).

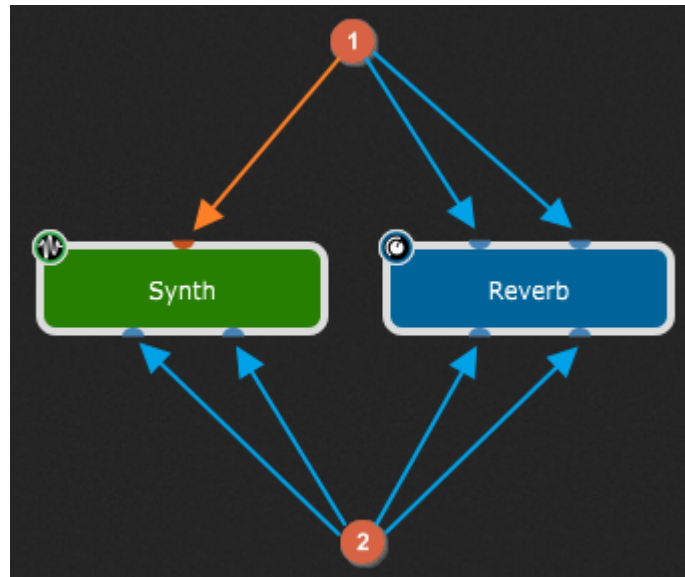
The Wiring view features, such as zooming, scrolling, multi-selection, zoom to fit, etc. are useful while creating your plugin layout. Learn more about these features [here](#).

**Tip:** to learn how to easily insert a plugin block into an audio or MIDI path, check the [Additional tips](#) chapter.

## 5.2 Blue and orange colored connections and pins

First, briefly examine the [Wiring view](#) chapter, where basic concepts such as pins and plugin connections are described.

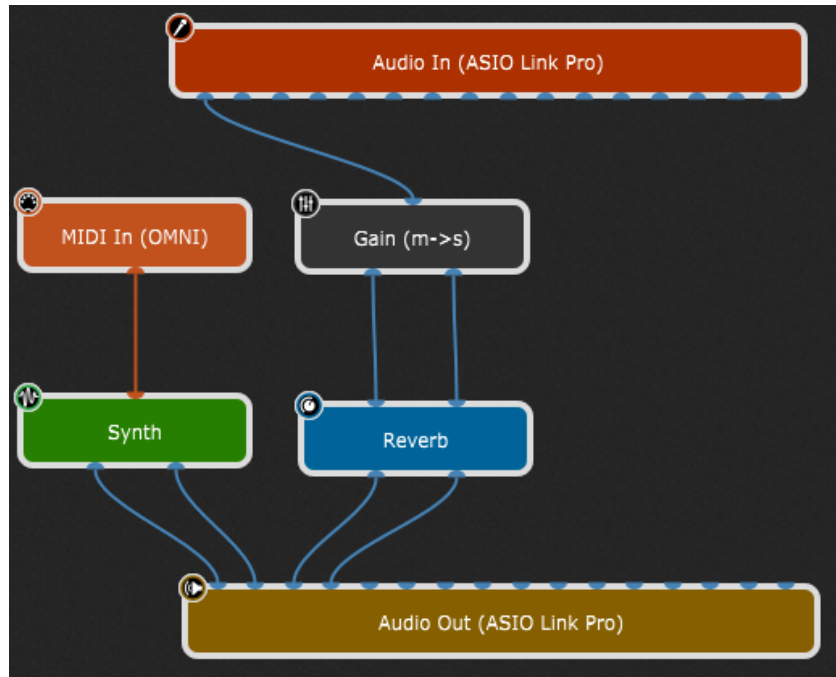
Let's [insert](#) two plugins, a reverb and a synth:



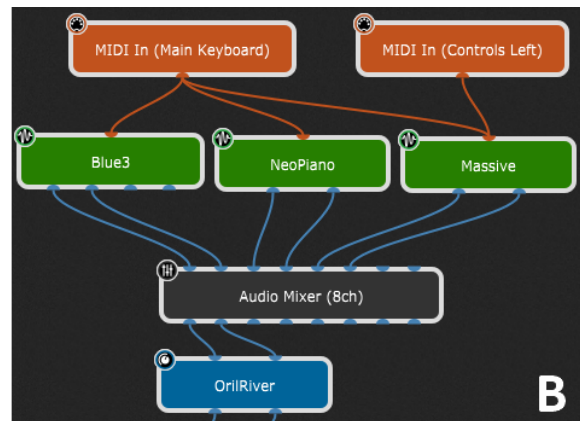
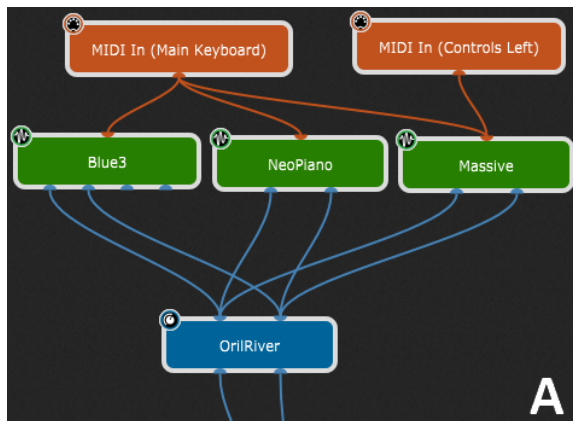
With **(1)** are indicated input pins (or input plugin ports), and with **(2)** are indicated output pins (or output plugin ports). Note that pins can be blue or orange (indicated with orange and blue arrows); blue pins indicate audio ports, and orange pins indicate MIDI ports.

Plugin blocks are connected by virtual wires, where an output pin of one plugin is connected to an input pin of another plugin. Note that orange pins cannot be connected to blue pins (and vice versa). For example, you can connect directly the Audio In block to the Audio Out block, but cannot connect directly the MIDI In block to the Audio Out block.

Likewise, orange connections (orange virtual wires) represent MIDI connections and blue connections (blue virtual wires) represent audio connections:



The power of Gig Performer lies in the fact that you can connect plugins any way you like, e.g. you can control multiple synths with one keyboard (you can connect an output pin to multiple input pins), or you can connect multiple output pins to a single pin:



When you connect multiple output pins to a single pin, signals are mixed automatically **(A)**; to have more control over your sound, use the built-in [Audio Mixer](#) plugin **(B)**.

The number of input and output pins depends on the plugin manufacturer and provided bus layout settings; in the screenshot below, note the difference between mono input and output buses and 7.1 Surround input and output buses for the same plugin:



Gig Performer lets you set a maximum number of pins as a global option (see [Display options](#) -> *Maximum number of audio channels to display in blocks*) and control the maximum number of pins on each plugin individually.

Refer to the [Plugin contextual menu](#) chapter to learn more about bus layout and max pins in plugin blocks.

Check out this blog article to learn more about plugin pins: [My plugin has so many output ports - which ones do I use?](#)

## 6 Panels and Widgets

A widget is an onscreen control that you can associate with a plugin parameter. A panel is a graphic space where widgets can be placed and labeled. Users typically create different panels to organize their controls in simple ways to identify and deal with in live performance.

In this chapter, you will learn more about panels and widgets.

### 6.1 What is a panel?

In Gig Performer, a panel is an area (visually similar to hardware rack-mounted sound modules and signal processors) containing widgets used to control the plugins in a rackspace. A rackspace can contain multiple panels, each containing one or more widgets:



The screenshot above shows a rackspace ("Blues Keyboard" with the "Delay off" variation, as indicated with the red rectangle) with two panels (indicated with the letters **A** and **B**).

In normal operation (i.e. when not in Edit mode), dragging a widget with your mouse (or by remote control from your keyboard or control surface) will change the value of that widget, thereby updating whatever parameter is attached to the widget. If you want to make changes to panels and widgets, activate Edit mode. Refer to the [Panels view](#) chapter for more information. In Edit mode, note the following options as depicted in the screenshot below:



To create a new panel, either:

- click on the **New panel** button (1) and then select the desired option from a drop-down menu (this drop-down menu has the same menu items as the sub menu (4)).
- or right-click on the empty panel surface (2) to open the [panel contextual menu](#) (3), then expand **Add new panel** and choose the option from the sub menu (4).

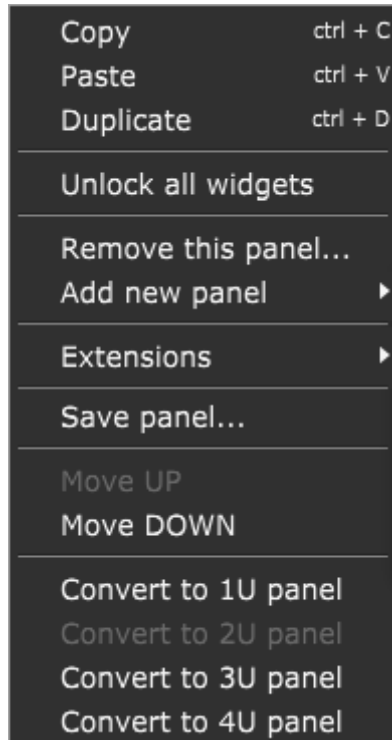
Every time you enter into Edit mode of the Panels view, tips appear in the **Properties** section: you can click on a panel or a widget to show their properties or add new widgets by dragging them onto a panel surface. When you click on a panel you can see **Panel properties** (5). Here, you can change the background style of the current panel. To change a panel's background, click the panel to select it and then either click one of the textures displayed in this area or choose a custom color.

See the [Widgets](#) chapter to learn more about widgets and how to add them to a panel.

**Note:** you can move a widget across panels in Edit mode. Simply drag a widget from one panel to another. This feature currently works only for single widgets, and all assignments to parameters are kept.

### 6.1.1 Panel contextual menu

If you are in the Panels view and are in Edit mode, if you right-click on a panel, the contextual menu appears:



This contextual menu allows you to perform the following operations:

**Copy** - copies the widget into the clipboard (alternatively, click on the **Edit** menu and select **Copy**).

**Paste** - pastes the widget from the clipboard (alternatively, click on the **Edit** menu and select **Paste**).

**Duplicate** - duplicates the currently selected widget (alternatively, click on the **Edit** menu and select **Duplicate**).

**Unlock all widgets** - unlocks all locked widgets on the current panel.

**Remove this panel...** - clicking on this menu item allows you to remove the current panel from the rackspace.

**Add new panel** - allows you to add a new panel; expand this menu item to see items in the sub-menu:

- **New 1U panel** - clicking on this menu item allows you to create a new 1U panel; "U" is an abbreviation for rack unit - a unit of measure defined as 1 ¾ inches (or 44.45 mm).
- **New 2U panel** - clicking on this menu item allows you to create a new 2U panel. If you create a new empty gig in Gig Performer, the default panel size is 2U.
- **New 3U panel** - clicking on this menu item allows you to create a new 3U panel.
- **New 4U panel** - clicking on this menu item allows you to create a new 4U panel.
- **Load panel...** - clicking on this menu item allows you to select a **gppanel** file and load the panel.

**Extensions** - this menu item is hidden by default and shows up if a third-party extension is enabled and provides one or more custom panels. Refer to the [Extensions](#) chapter for more information.

**Save panel...** - clicking on this menu item allows you to save a current panel as a **gppanel** file, which can be easily reused. Widgets and their MIDI mappings are saved as well.

**Move UP** - moves the current panel one place higher to the beginning of the rackspace.

**Move DOWN** - moves the current panel one place lower to the end of the rackspace.

**Convert to 1U/2U/3U/4U panel** - clicking this option converts the current panel to a 1U/2U/3U/4U panel, respectively. Note that converting the unit size (e.g. from 2U to 1U) may reposition and reduce the size of your widgets.

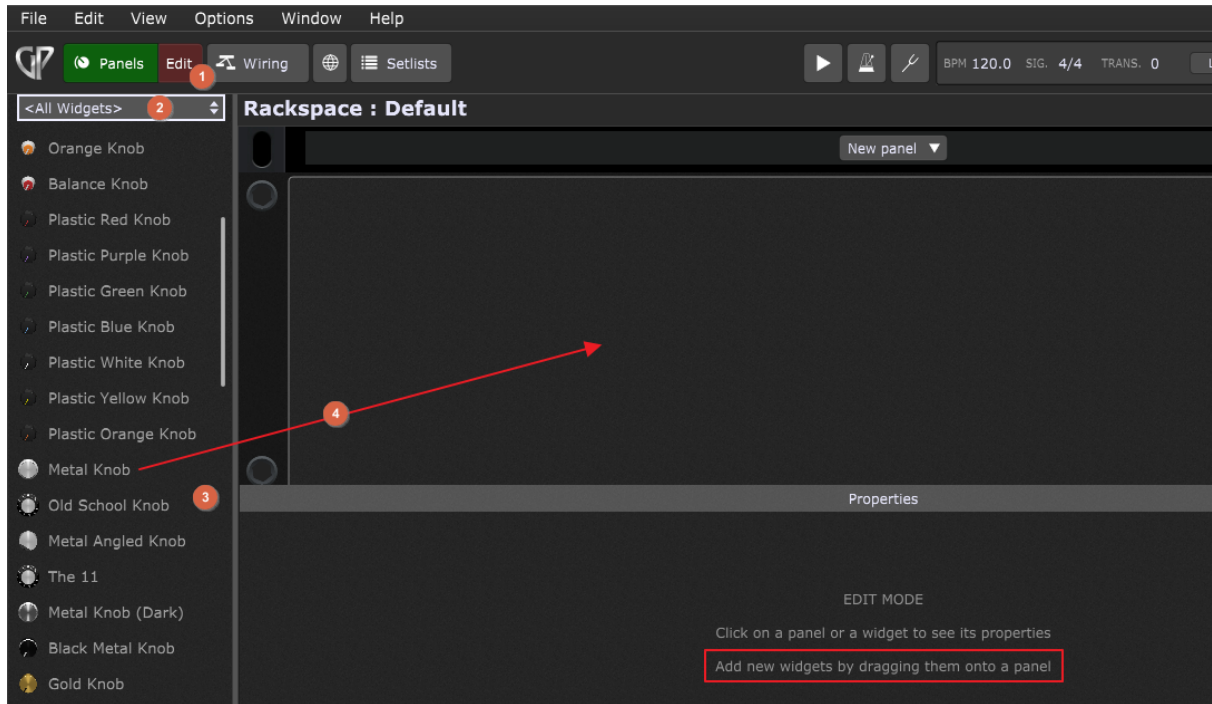
## 6.2 Widgets

Basic information about widgets is available in the main [Widgets](#) chapter. This chapter will cover widgets in more detail, providing in-depth explanations of various widget operations.

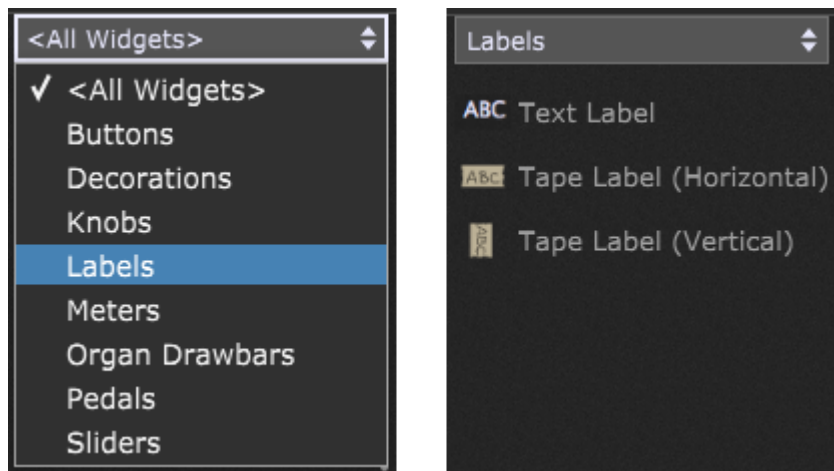
### 6.2.1 How to add widgets to a panel?

A widget is added to a panel by either dragging the desired widget from the widget list to that panel or by copying and pasting (or duplicating) an existing widget:





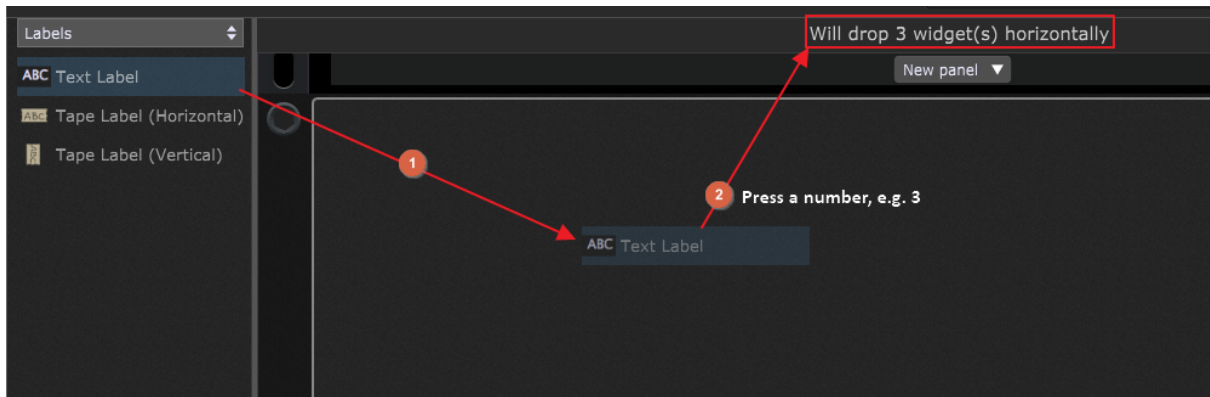
To add a widget to a panel, first ensure you are in **Edit mode (1)**. Once you enter Edit mode, you can either immediately drag a widget **(4)** from the widgets list **(3)** onto a panel (also note the tip indicated with the red rectangle) or click on the Widget types list **(2)** to restrict the list to a particular class of widgets:



For example, if you want to add a label widget, simply select Labels from the drop-down box and then simply drag the desired label widget from this list to the panel.

To add multiple copies of a widget at once, drag a widget onto the surface, but don't release the mouse button **(1)**; while holding the mouse button, type the number of copies (up to 9) you want onto the panel **(2)**. A message will flash at the top of the

window (indicated with the red rectangle) to confirm the number of copies that will be made:



Then release the mouse button. **Note:** press the desired number again, e.g. 3 while dragging to drop 3 widgets vertically.

**Tip:** if you add 9 drawbars at once, you will get the perfect combination of a black, white, and brown 9 drawbar set.

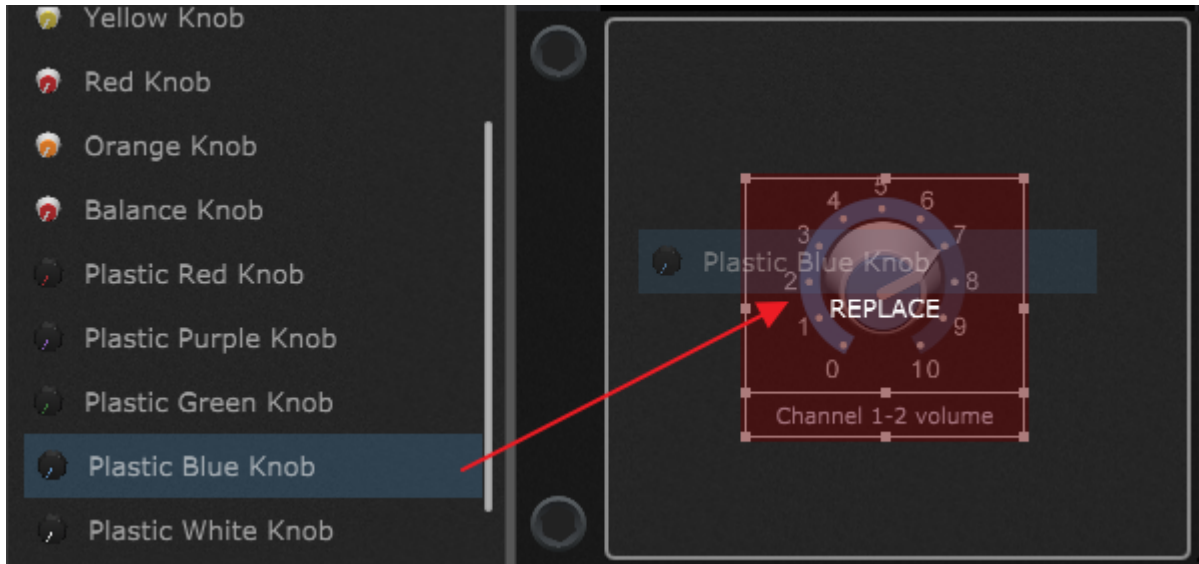
Existing widgets already placed onto the panel can be easily duplicated, copied or pasted by clicking on the **Edit** menu and selecting the **Duplicate**, **Copy** or **Paste** menu items, respectively. Alternatively, use key combinations **Ctrl+D** (or **Cmd+D**), **Ctrl+C** (or **Cmd+C**) or **Ctrl+V** (or **Cmd+V**) respectively.

**Note:** A confirmation message flashes when copying a widget or panel. MIDI associations are preserved whenever you copy and paste a widget or a panel.

You can resize a widget by dragging any corner points (the smaller section directly beneath the white rectangle displays a caption for the widget).

As you may notice, a name of more than a few characters may not display properly within the default caption space; as mentioned earlier, the white rectangle that appears around the selected widget (and caption below) in Edit mode allows you to resize it (and the accompanying caption) by dragging any of the corner points.

To replace an existing widget with another one, simply drag the widget over the existing one and when the REPLACE label appears, release the mouse button:



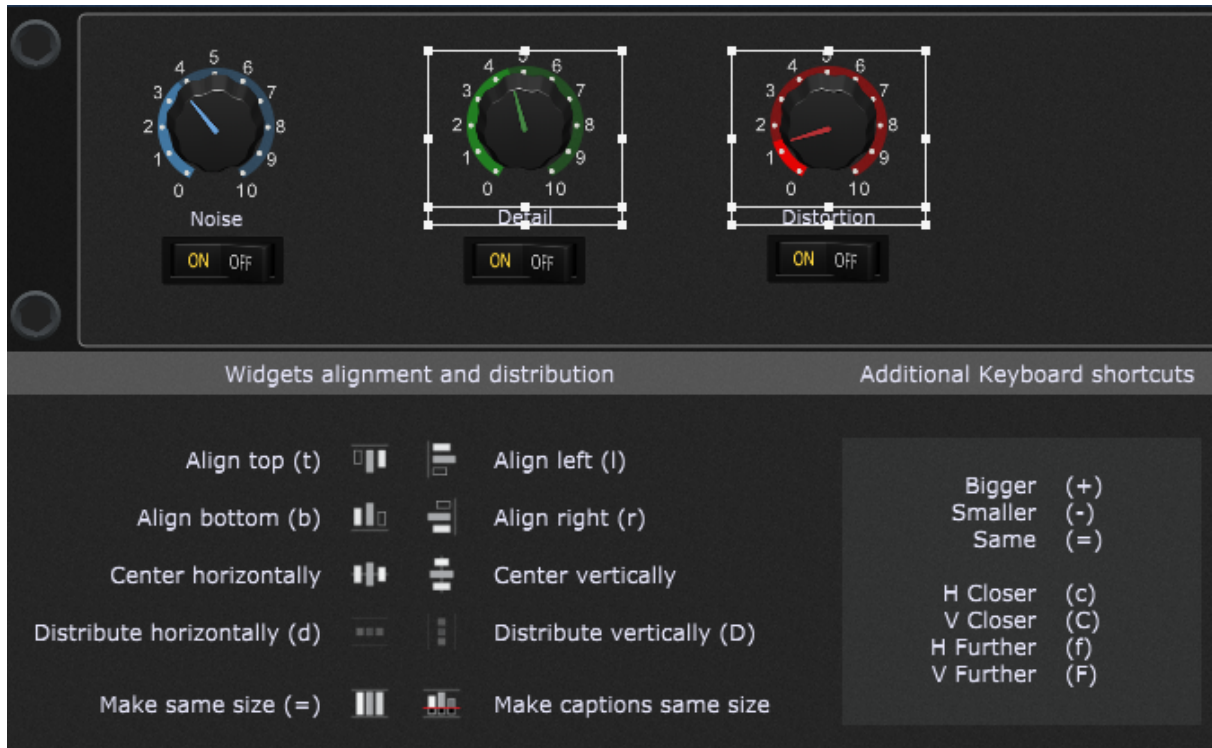
In the screenshot above, the Blue Knob (the existing widget) is being replaced by the Plastic Blue Knob. Note that you cannot replace knob widgets with buttons, switches etc. If the replace operation is not supported, the REPLACE label will not appear.

All MIDI and Parameter mappings and other widget settings are saved when replacing a widget.

**Note:** moving a widget across the panels is also supported. While in Edit mode, simply drag a widget from one panel to another. This feature currently works only for single widgets, and all assignments to parameters are kept.

### 6.2.2 Widget alignment and distribution

When multiple widgets are selected, widget alignment and distribution tools become available:



Click the associated button or use the displayed key shortcut to apply the alignment or distribution function to the selected widgets:

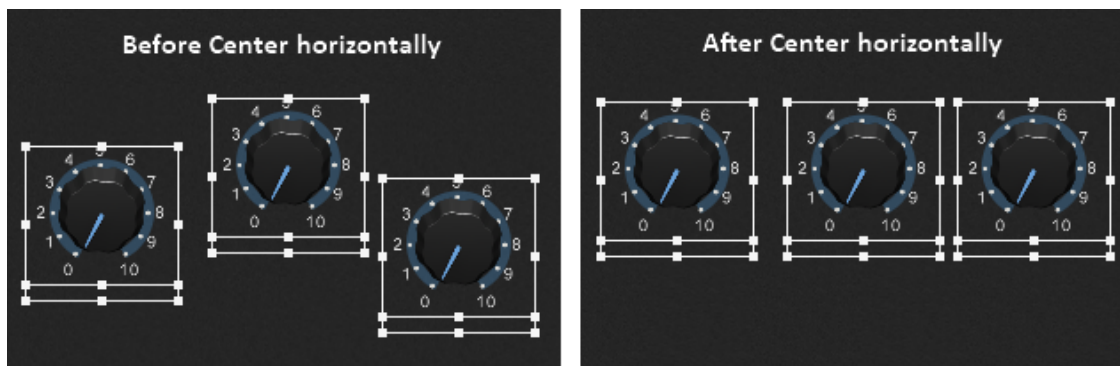
**Align top (t)** - align the tops of the selected widgets.

**Align bottom (b)** - align the bottoms of the selected widgets.

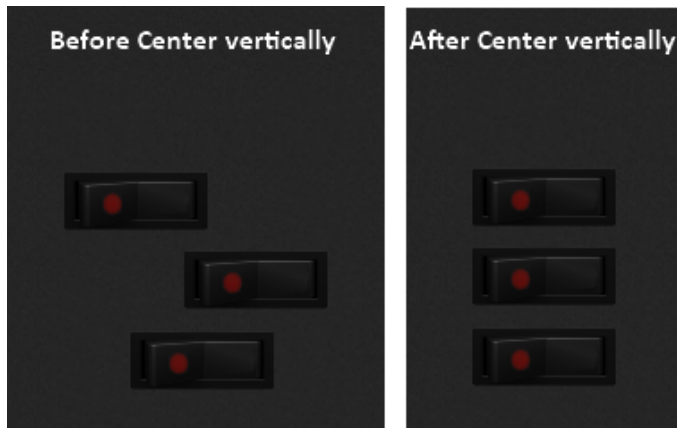
**Align left (l)** - align the left sides of the selected widgets.

**Align right (r)** - Align the right sides of the selected widgets.

**Center horizontally** - aligns the selected widgets so they are all lined up on the X axis at the same value of Y.



**Center vertically** - aligns the selected widgets so that they are all lined up on the Y axis at the same value of X.



**Distribute horizontally (d)** - the selected objects are moved to have even horizontal spacing between them. Note: the widgets are distributed evenly across the space between the left side of the leftmost widget and right side of the rightmost widget.

**Distribute vertically (D)** - the selected objects are moved to have even vertical spacing between them.

**Make same size (=)** - clicking this button makes all of the selected widgets the same size as the first selected widget.

**Make captions same size** - this button changes the size of the caption text areas for all of the selected widgets to match the size of the first selected widget.

Additional keyboard shortcuts that also apply to selected widgets:

**Bigger (+)** - grow selected widgets.

**Smaller (-)** - shrink selected widgets.

**Same (=)** - make all selected widgets the same size as the first one selected.

**H closer (c)** - move the selected widgets closer to each other horizontally (along the X axis).

**V Closer (C)** - move the selected widgets closer to each other vertically (along the Y axis).

**H further (f)** - move the selected widgets further from each other horizontally (along the X axis).

**V Further (F)** - move the selected widgets further from each other vertically (along the Y axis).

### 6.2.3 Widget z-ordering

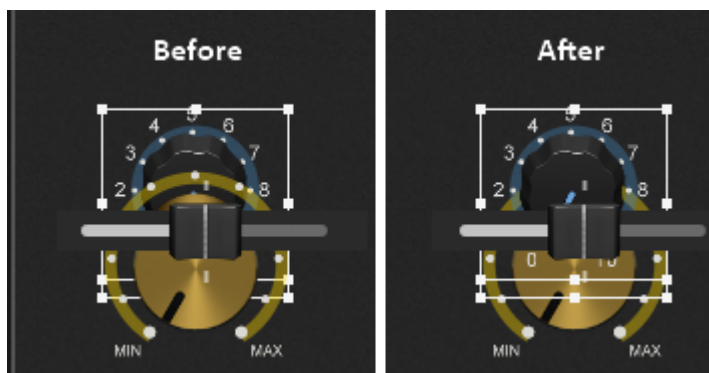
If you [insert a widget](#), e.g. a plastic blue knob, and then insert another one, e.g. a gold knob, and try to place them in the same position, you'll notice that the gold knob overlaps the plastic blue knob:



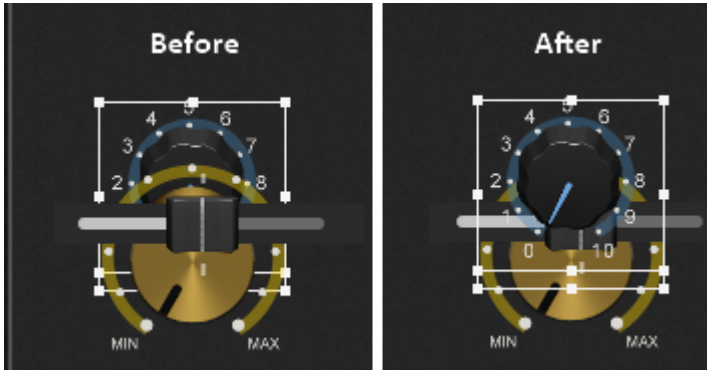
Similarly, suppose you insert another widget, e.g. a horizontal black slider and place it at the same place where the existing widgets are located. In that case, you'll notice that the horizontal black slider overlaps the existing ones. The most recently inserted widget will be positioned above earlier widgets at the same location. Note that the hierarchy is preserved here: you can drag the gold widget away, but the horizontal black slider will remain above the plastic knob; dragging the gold knob back to the same place will place it between the plastic blue knob and the horizontal black slider.

This order can be controlled by right-clicking on a widget and selecting one of the following [contextual menu](#) items:

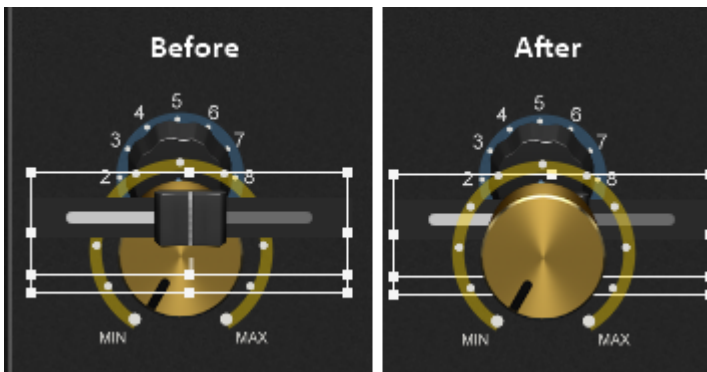
- **Bring forward** - places the currently selected widget one place higher in the order.



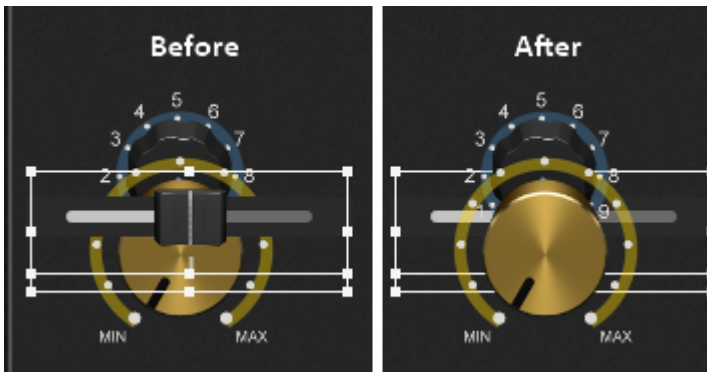
- **Bring to front** - places the currently selected widget in front of all other widgets.



- **Send backward** - places the currently selected widget one place lower in the z-order.



- **Send to back** - places the currently selected widget behind all other widgets.



## 6.2.4 Widget Properties inspector

Adding or selecting a widget in [Edit mode](#) opens the Widget Properties inspector, which is divided into two sections: **Widget Properties** and **Mapping**.

### Widget properties

The Widget properties section is divided into four tabs - **General**, **MIDI**, **Value** and **Advanced**.

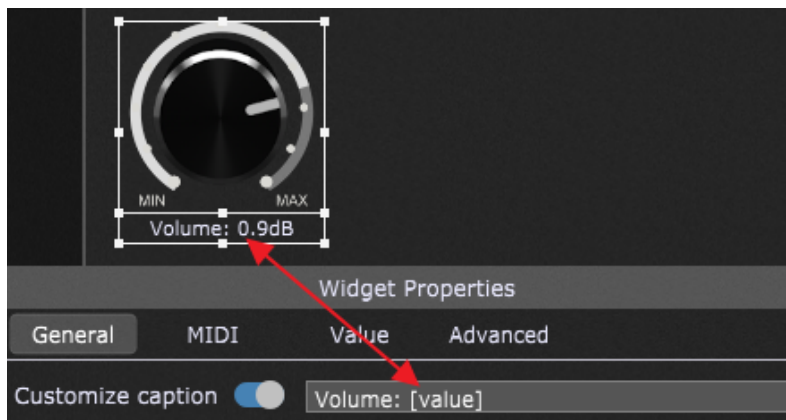
**General tab:**



The number of options and their availability depend on the type of widget. The screenshot above shows available options for a button. We'll describe all the options below.

**Customize caption** - enable this toggle button to have the widget display a custom caption and then type in a new name. If this toggle button is disabled, the default caption is defined by the selected parameter name (refer to the Mapping section below for more information).

**Note:** if you add the "[value]" tag (without quotation marks) to your caption, then the parameter value will also be included in the caption:



If you add the "[variation]" or "[rackspace]" tag (without quotation marks) to your caption, then the name of the corresponding variation or rackspace, respectively, will be included in the caption. Finally, if you add the "[notename]" tag (without quotation marks), then values from 0 to 127 will be converted to note names, from C2 to G8, respectively.

Another supported tag is "[nl]":





This tag allows you to add new lines in a widget caption. (see the example screenshot above)

**Display range** - this option applies to knobs and sliders and is enabled by default; when disabled, the widget doesn't show the ring around the outside of the knob or slider:



When this toggle button is enabled, you can notice two colors: the total range (dimmed) and the scaled widget value (brighter color).

**Show value in caption** - this option is disabled by default; when enabled, it overrides the defined caption and displays only the parameter value.

**Momentary touch** - Sustain pedal, Buttons and Switches include the *Momentary touch* option to control the momentary behavior on mouse clicks or touch. Disabling this property will keep the sustain pedal up or down until you click on it again. Enabling this property will keep the pedal pressed until you release the mouse click.

**Lock** - this feature allows you to lock a widget so that it cannot be adjusted using a computer mouse, touch screen or trackpad. Locked widgets have an overlay in the upper-left corner:



By default, this overlay is just displayed in Edit mode. If you want to show it always, turn on the *Always indicate locked widgets* toggle button in the [Display Options](#).

**Note:** Label widgets are locked by default. If you unlock a label widget and map it to a plugin parameter (e.g. *Channel 1-2 volume* from an Audio Mixer plugin), then clicking on that label widget when you're not in Edit mode allows you to change the parameter value for that plugin:



This way, it's easy to enter a specific value that you want a parameter to have and always set it to that value. Additionally, entering MIDI values is also supported so as to make it easy to set some parameter to a MIDI value - if a plugin supports program change as a host parameter, you could easily specify MIDI value 42m rather than having to turn a widget to find the value.

**Widget link** - each widget can only be associated with a single plugin parameter. However, the 26 widget link groups (from A-Z) enable you to control multiple parameters from any or all widgets assigned to that link group. Click on the dropdown list (right to the "Widget Link") and select a letter ranging from A to Z to assign a widget to that link group (e.g. "V"). All widgets with the same letter are linked together so that moving one widget will cause all the other widgets in that link group to move as well, adjusting various plugin parameters simultaneously:



**Note:** each widget can have its own scale curve so that the actual values sent to the individual plugin parameters do not have to match. Check the [How to add a widget to a Widget Link group](#) chapter for more information. A widget cannot be a member of both radio group and a widget link group.

If you want the widget link and radio group labels to stay always, i.e. even if you are not in Edit mode, then turn on the *Always show Widget Link and Radio Group labels* toggle button in the [Display Options](#).

**Tip:** Pressing a character (A-Z) while holding the **Alt** key (or **Option** on Mac) down will assign all selected widgets to that character's widget link group. Alternatively, select the desired widgets and add them to a link group using the [Widget Contextual Menu](#).

**Show back plate** - this option is enabled by default on widgets that support it; when disabled, the "back plate" information such as numbers, Min/Max or an "on/off" legend is not displayed:



**Note:** this option is enabled only when you are editing a label, a knob, a toggle switch or a LED button.

**Hide temporary value** - when this toggle button is turned on, the parameter value is never displayed when a widget changes.

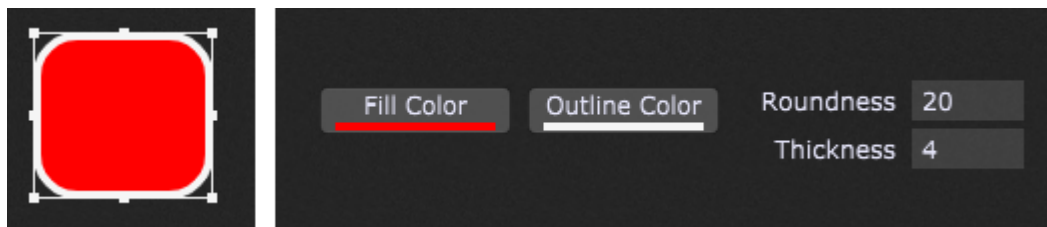
**Hide** - clicking this toggle button will hide the widget when Edit mode is inactive. **Note:** hidden widgets are semi-transparent in Edit mode for easier identification.

**Radio group** - allows you to group multiple widgets where only one widget in the group is turned on at a time. Supported widgets are buttons, LEDs, switches, and the Sustain

pedal. You can define up to 99 radio groups. A widget cannot be a member of both radio group and a widget link group. To learn more about this feature, visit the chapter [How to add a widget to a Radio Group](#).

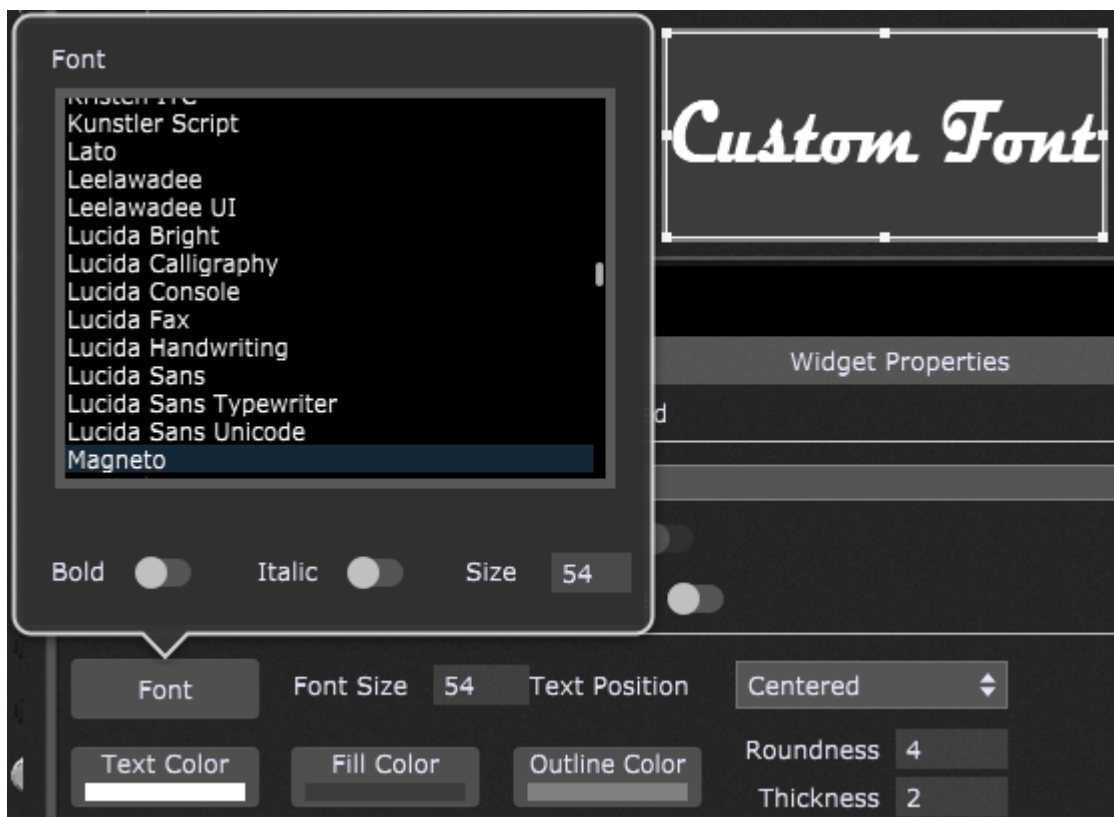
Some widgets include the additional properties in the General tab; these are:

1. **Shape widget** - this widget includes additional properties: *Fill Color*, *Outline Color*, *Roundness* and *Thickness* (of the border):



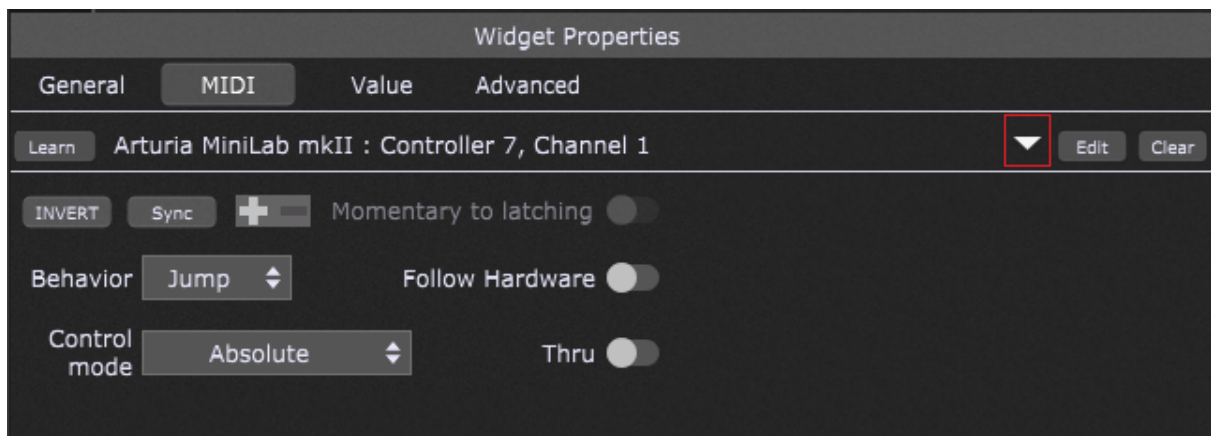
Simply click on the corresponding buttons and then pick fill and outline colors (as well as their transparency).

2. **Label widgets** - these widgets (Text Label and Tape Label Horizontal/Vertical) include properties described in 1. plus additional properties:



- **Font** - clicking on the Font button allows you to select the desired font, bold and italic styles and font size. By default, only Gig Performer standard fonts are listed; to allow the use of other fonts from your computer, turn on the *Allow local fonts toggle in labels* toggle button in the [Display Options](#).
- **Font Size** - allows you to select the desired font size between 4 and 256.
- **Text Position** - allows 9 text positions: Centered, Centered Left, Centered Right, Centered Top, Centered Bottom, Top Left, Top Right, Bottom Left, and Bottom Right alignment.
- **Text Color** - allows you to select the desired text color and the transparency.

#### MIDI tab:



**Learn button** - click on this button to have Gig Performer learn the physical control you want to associate with the selected widget. When actively in Learn Mode, the button is lit red; to exit Learn Mode, click on it again, and it will return to its normal gray color. Refer to the [How to connect a widget to a MIDI device](#) chapter to learn more.

**Note:** the only CC messages that Gig Performer cannot learn are **CC 0** and **CC 32**, as these two messages are reserved for bank selection.

**Drop-down menu** - clicking this drop-down button (indicated with the red rectangle) opens a menu allowing you to select a MIDI control alias. Check the [How to use aliases](#) chapter for more information.

**Edit button** - click on this button to manually associate a MIDI device, MIDI message type and other parameters (depending on the message type, e.g. Note, Channel, CC number) with the selected widget.

**Clear button** - click on this button to clear the current controller assignment.

**Invert button** - when turned on, the widget's behavior is reversed; for example, a switch or a button will cause a parameter to go to minimum when in the ON (lit) position, and vice versa.

**Sync button** - pressing this button engages bidirectional communication between Gig Performer and the controller mapped to the control. This lets physical controls on control surfaces reflect changes executed in Gig Performer, and Gig Performer widgets show changes made on the physical control to which it is linked.

**Controller polarity** - displays the polarity (+ or -) for a controller. To invert the polarity of a MIDI controller such as a sustain pedal, first create a MIDI Control alias in the Rig Manager and set the desired polarity ([click here](#) and check point 15 d) to learn more). Afterward, select that MIDI control alias by clicking on the red rectangle (see the screenshot above).

**Momentary to latching** - this toggle button is displayed only when editing switches, buttons, LEDs or a sustain pedal. When checked, the widget remains ON until you click on it (or touch the associated physical controller) a second time. This is useful when you want to have a widget carry out an operation until stopped, i.e. staying at a fast Leslie speed until you are ready to return to a slow Leslie speed.

**Behavior** - this drop-down menu allows you to choose between *Jump* and *Catch* modes. When a widget is in Jump mode, moving the associated physical controller will cause the widget to jump to the position of the controller immediately. When a widget is in Catch mode, the widget will not respond to the associated physical controller until the controller's value matches the widget's current value, at which point the widget will follow the controller. **Note:** if the **Follow Hardware** toggle button is turned on, this menu will change to *Initial Behavior*.

**Follow Hardware** - to enable this toggle button you must learn or manually assign a MIDI device and a CC message. This feature allows you to set a widget to the last seen value of the hardware (pedal, slider, knob, etc.) when you switch to the rackspace that contains this widget. When you turn on this toggle button, then:

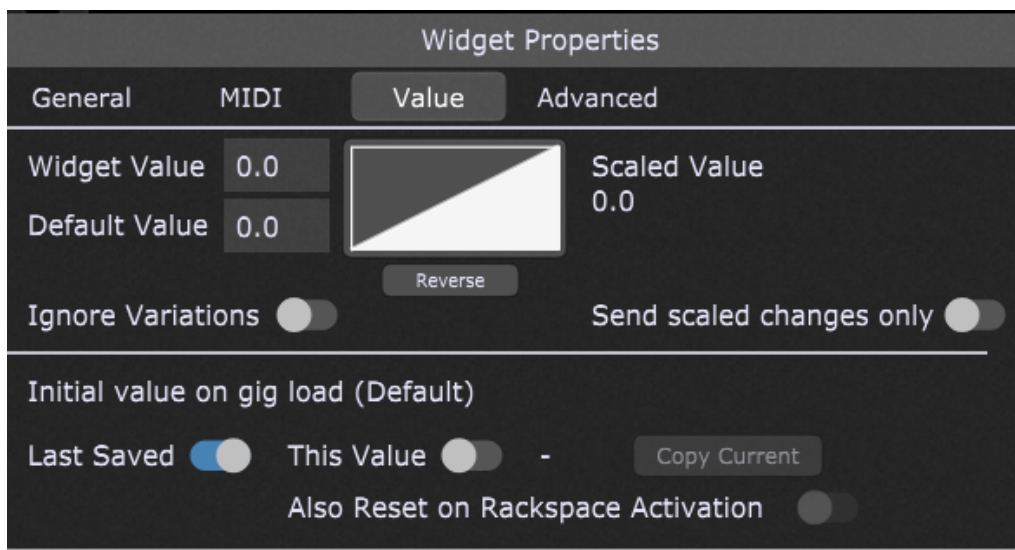
- *Ignore Variations* toggle button in the Value tab will be hidden,
- *Also Reset on Rackspace Activation* toggle button in the Value tab will be hidden.

**Control mode** - allows you to set either absolute (default) or relative control mode for a widget. There are 4 different relative modes: Relative Offset, Signed 1 Bit, Signed Magnitude, and Two's Complement.

**Note:** If you previously selected a MIDI control alias (using the drop-down button indicated with the red rectangle), the Control mode is managed in the Rig Manager. Refer to the [Rig Manager](#) chapter to learn more about the Rig Manager and various control modes.

**Thru** - by default, this toggle button is turned off, so a MIDI message learned by a widget will not be forwarded to plugins. Turn on this toggle button to send out the learned MIDI message to plugins. For example, you can both play a note and have that note turn on your Leslie effect.

#### Value tab:



**Widget value** - each widget has a range of 0 to 100. Note that this is a percentage, not an absolute value. For example, suppose the plugin parameter it is controlling has an absolute range from 0.0 to 1.0. When the widget is at its "100" setting, it actually changes that parameter to 1.0 (i.e. 100% of 1.0). The value of widgets from the Buttons category (buttons, switches, and LEDs) can only be 0 or 100 - there is nothing in between, and the only values that a button can emit are the values at those positions. Anything in between will be ignored. You can also directly enter values using MIDI numbers by adding the letter "m" at the end of the value. For example, if you type in "64m", Gig Performer will interpret it as an exact value of 50.39 (roughly 50% of the maximum MIDI value of 127).

**Default value** - continuous widgets allow changing the default value; this value cannot be set on others. When you double-click on the widget (when not in Edit mode), the widget value is reset to the specified default value.

**Scaled Value** - this value is the actual value that is sent to the plugin parameter. The default scaling for a newly inserted widget is the same as the actual widget value, as shown by the triangle shape of the curve. Scaling is based on the defined curve; find out more about curves in the [How to control the widget value scaling](#) chapter.

**Reverse button** - inverts the current curve horizontally. For example, if your curve has a default linear shape where widget value 20 results in scaled value 20, clicking the Reverse button will scale the widget value 20 to be sent out as the value 80.

**Ignore variations** - this toggle button is turned off by default; when turned on, the selected widget will be excluded from variation changes and will remain at its current value. This is useful if you have, for example, several different variations that change the widget settings controlling effects such as phasing or flanging. Still, you want the widget controlling overall volume to remain constant. **Note:** if this widget is part of a radio group, turning on the *Ignore variations* toggle button will also apply to all widgets in the radio group. Also note that this toggle button will be hidden if the *Follow Hardware* toggle button is turned on in the MIDI tab.

**Send scaled changes only** - turning this toggle button on prevents widgets from sending out values if the new value would be the same as the previously sent value. This feature is intended, for example, to allow expression pedals to be used for switching to the next song. Next, in conjunction with the appropriate scaling, this mechanism also allows you to use continuous controllers in a stepped mode, effectively thinning incoming MIDI CC messages.

**Send on Radio button off** - this option is displayed only on widgets in a [radio group](#); it allows you to send out the value when this (currently selected) widget turns off.

**Initial value on gig load (variation\_name)** (not available in the Global rackspace) - Gig Performer by default restores all widgets to their last positions in variations when a gig file is saved (the **Last saved** toggle button will be enabled in this case).

You can, however, click on the **This value** toggle button to take a snapshot of the current position of the selected widget so that position gets recalled and loaded instead. This can be useful if you are playing, for example, a song where the volume of your piano always starts softly but gets quite loud by the end. When you come back to that song again, you don't want the piano to still be loud - you want it always to start soft. The concept is that, even if you play around with widgets during your live performance, your originally stored values will return when you next reload a gig file.

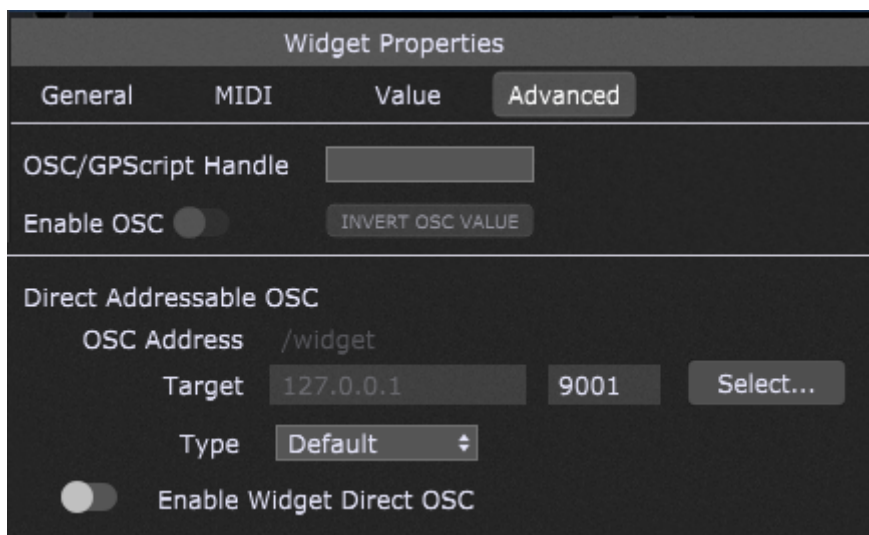
Click on the **Copy Current** button (it is enabled only when the "This value" toggle button is enabled) to take another snapshot of the current widget position.



Additionally, clicking the **Also reset on rackspace activation** toggle box resets the widget value to "This value" on the rackspace activation. Note that this toggle button will be hidden if the Follow Hardware toggle button is turned on in the MIDI tab.

**Note:** Widget values are always automatically saved and reloaded in your gig file per variation. Using the explicit load value, the current widget will be reset to a specific value when your gig loads, even if you change it later and save the gig file.

#### Advanced tab:



**OSC/GPScript Handle** - allows you to set a name for use in GPScript, OSC, Timeline and Song Part actions and Gig Performer extensions. To learn about handles in Gig Performer, see the blog article [What is a handle and how to use it](#).

**Enable OSC** - click on this toggle button to make the OSC/GPScript Handle field available to OSC. For more information on OSC, [click here](#).

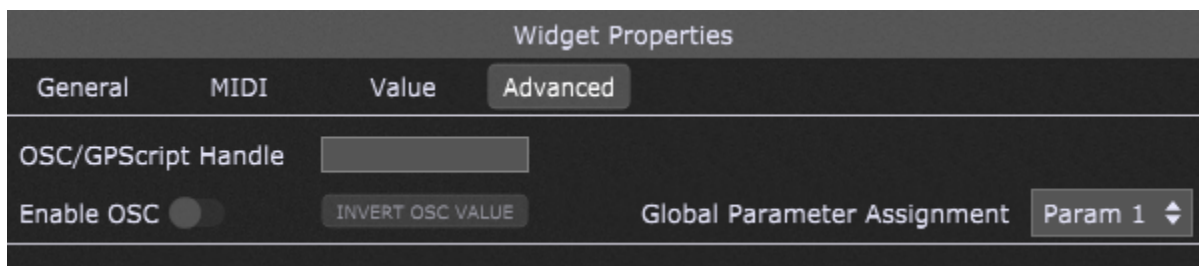
**Invert OSC value button** - clicking on this button inverts the OSC value. Click again to revert.

**Direct Addressable OSC** (this section is only visible when the *Enable Widget Direct OSC* option is enabled in [OSC Options](#) and is intended for advanced control of remote devices):

- **OSC Address** - type in a valid OSC address of the remote device.

- **Target** - type in a valid target IP of the remote device and afterward type in the listening port of the remote device. Clicking the **Select...** button allows you to easily select an OSC target from the list that must be previously set in [OSC Options](#).
- **Type** - allows you to set the type of the argument for an OSC message (Default, Float, Integer, and String).
- **Enable Widget Direct OSC** - clicking on this toggle button allows you to send out widget values as OSC messages for external processing. **Note:** OSC Address and Target must be defined to enable this toggle button.

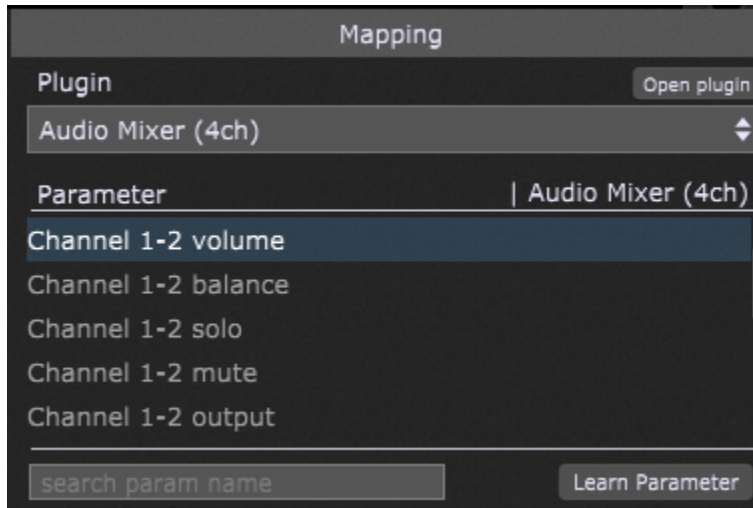
**Note:** global widgets (widgets in the [Global rackspace](#)) have one additional setting - *Global Parameter Assignment*:



Refer to [this chapter](#) to learn how to use this setting.

## Mapping section

The mapping section is used to connect widgets to plugin parameters:



**Plugin** - click on this drop-down to view the list of plugins in the current rackspace, and then select the plugin you want the widget to control.

**Open plugin button** - click this button to open the editor for the selected plugin.

**Parameter** - parameters are displayed only when a plugin is selected; select the one from the list of available controllable parameters that you want to have the widget control. Alternatively, you can search the parameter name by typing a partial parameter name in the *search param name* search box or, even easier, use the **Learn Parameter** button.

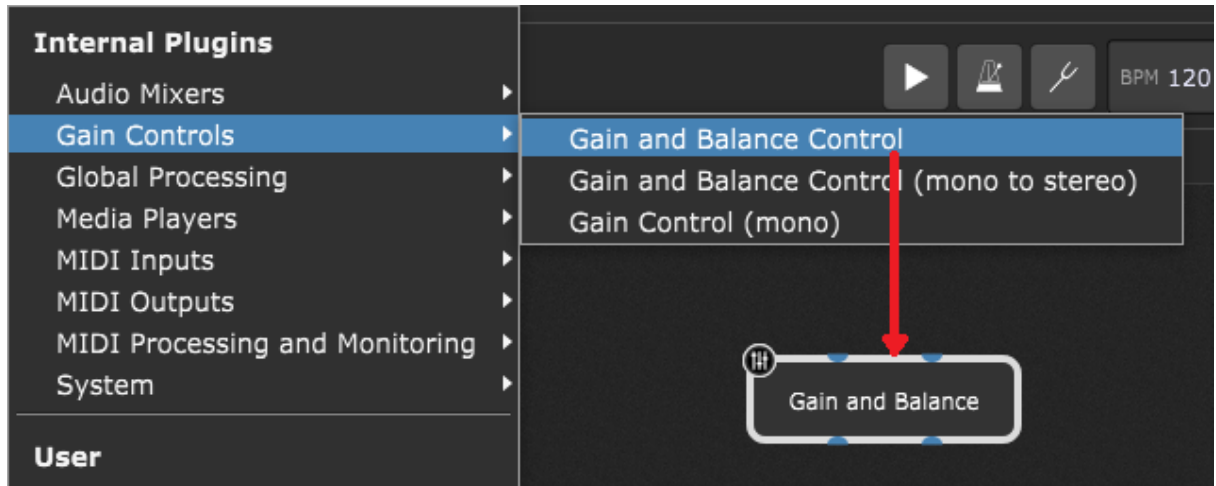
**Learn Parameter button** - click this button to have Gig Performer learn the plugin parameter you want to associate with the widget. This button is displayed only when a plugin is selected. When in Learn Mode, the button is lit red; to exit Learn mode, click on it again, and it will return to its normal gray color.

**Note:** widget assignments to physical controllers in the Widget Properties area override the [Global MIDI Options](#) assignments.

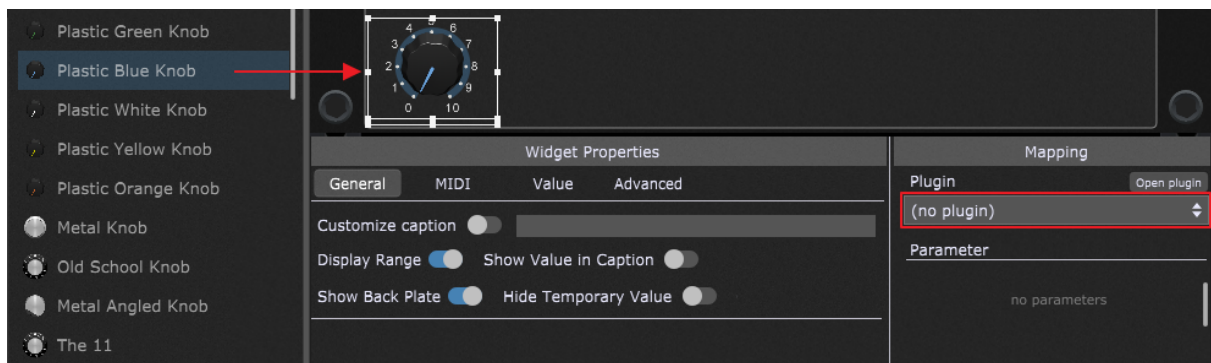
Refer to the [How to connect a widget to a plugin parameter](#) chapter for more widget mapping and parameter learning information.

### 6.2.5 How to connect a widget to a plugin parameter?

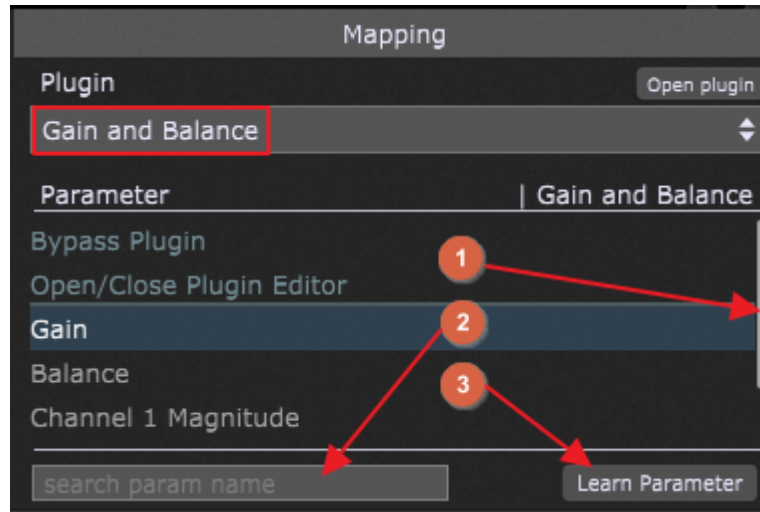
To connect a widget to a plugin parameter, first, switch to the [Wiring view](#), right-click on the empty surface to open the [contextual menu](#), and then select and insert the desired plugin (e.g. a [Gain and Balance Control](#), as indicated with the red arrow):



Alternatively, you can use the [Quick Plugin, Favorite or Preset Finder](#) to add plugins in the Wiring view. Afterward, switch to the [Panels view](#), Edit mode and drag the desired widget onto a [panel surface](#) (e.g. a Plastic Blue Knob, as indicated with the red arrow):

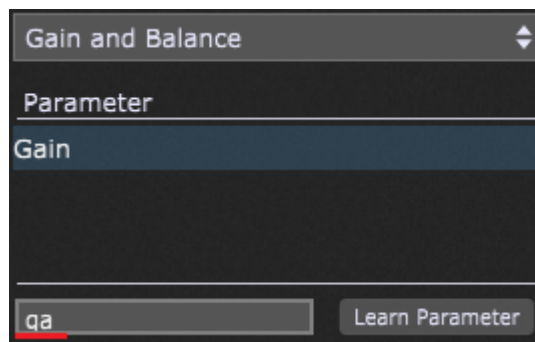


Click on the plugin drop-down menu in the Mapping section of the [Widget Properties inspector](#), (as indicated with the red rectangle) and select the desired plugin (in this case, the Gain and Balance plugin, as indicated with the red rectangle):

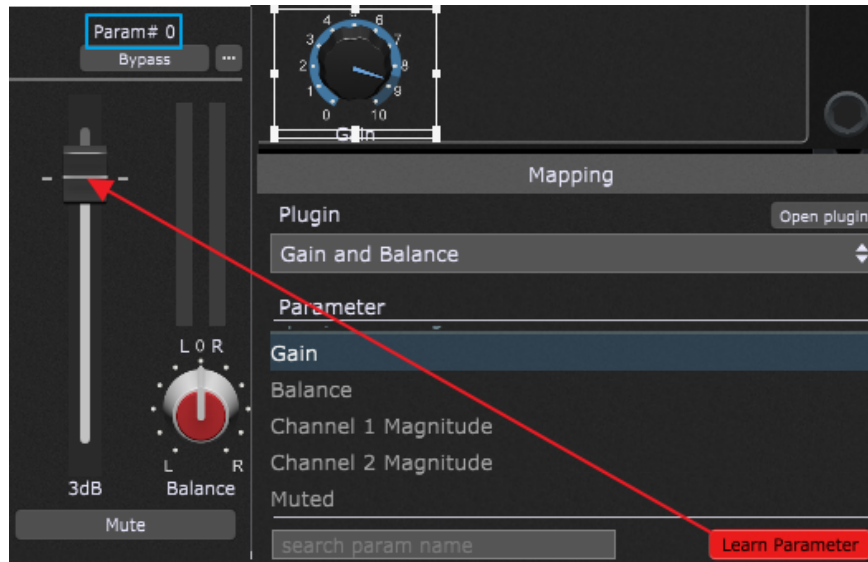


To select the actual parameter, do one of the following:

1. Simply navigate through this list using the scroll bar and select the desired parameter (e.g. Gain),
2. Start typing part of the parameter name into the search box to restrict the visible list of parameters to those matching what you have typed so far:



3. Click on the **Learn Parameter** button; a plugin's editor opens automatically and the background of this button turns red, indicating that you entered into the Learn mode. Simply click on the desired control (indicated with the red arrow), and the parameter will automatically be selected in the parameter list:



Click on the Learn Parameter button again to turn off the learning mode.

**Note:** every time you move the widget control, its parameter number displays, e.g. "Param #0" (indicated with the blue rectangle); this means that this parameter is the first in the parameter list (counting starts from 0) and below the common parameters "Bypass Plugin" and "Open/Close Plugin Editor". This display is specifically intended for advanced users wishing to control a parameter directly from [GPScript](#) or [OSC](#) and need to know the actual parameter number.

Test your configuration by moving the Plastic Blue Knob widget - if you successfully learned the parameter, the gain control in the plugin window should move accordingly (and vice-versa).

**Tip:** you can use a Text label widget to display a transparent note. Leave the "no plugin" mapping and change the widget value to 0 for the highest transparency; increasing this value (up to 100) decreases the transparency effect.

To clear the parameter mapping from a widget, simply select "(No Plugin)" from the combo box in the Mapping section.

If Gig Performer detects unnamed plugin parameters, it displays them as e.g. Param 5, Param 6, Param 7 (visit [this KB article](#) to learn more).

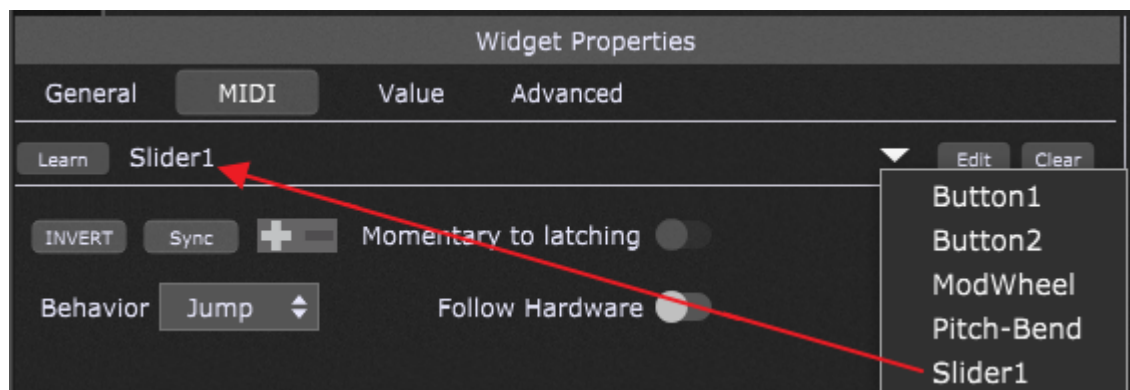
### 6.2.6 How to connect a widget to a MIDI device?

This chapter naturally builds on the previous chapter. When you insert plugins, create widgets, and connect them to the plugins' parameters, the next step is to control those widgets with your MIDI controller or remotely via OSC commands.

To associate a widget with a MIDI device, first enter into the [Panels view](#), Edit mode, and click on the desired widget (e.g. a Plastic Blue Knob, if you followed the example from the previous chapter; if you created a new empty gig, drag the Plastic Blue Knob widget onto a [panel surface](#) and click on it):

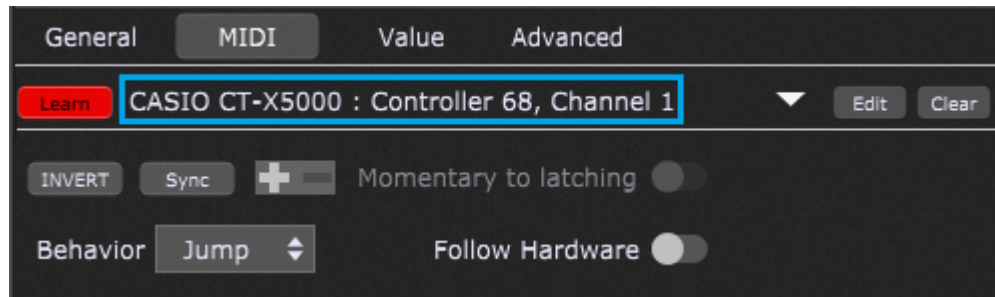


Click on the MIDI tab (indicated with the red rectangle) in the [Widget Properties inspector](#) and note that no MIDI assignment has been made yet **(2)**. If you defined MIDI control aliases in the [Rig Manager](#), you can easily click on the drop-down menu **(3)** and select the desired alias:



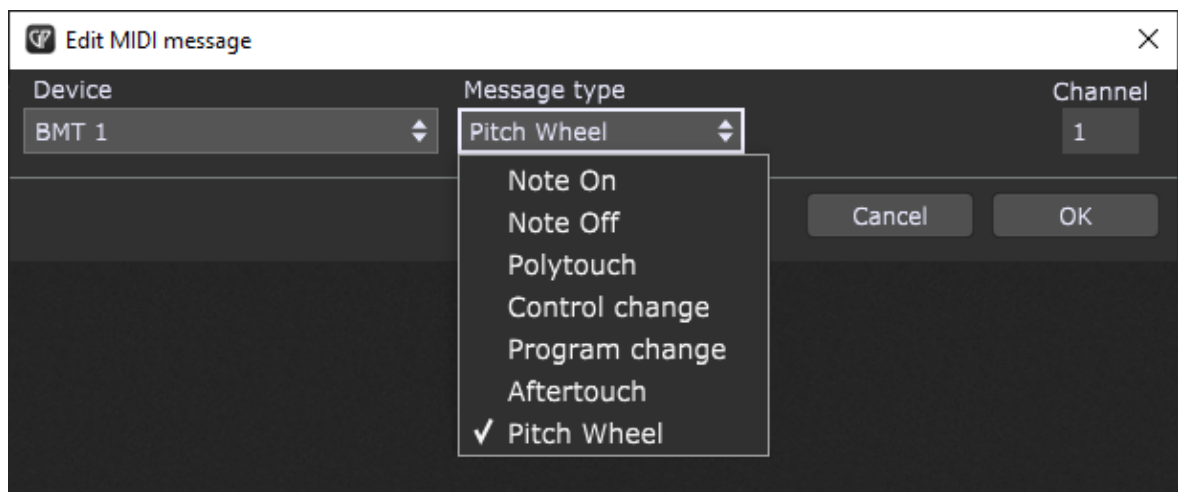
Note how the "(no MIDI In assignment yet)" message changes to match the name of the selected alias.

To have Gig Performer learn the physical controller you want to associate with the widget, click the **Learn button (1)**; when actively in Learn Mode, the button is lit red. While in Learn mode, move the physical controller. If you already have an alias defined in the Rig Manager for that control, that alias will display; if you didn't define an alias (such as "Slider1" in the screenshot above), then the name of your MIDI device will show up, as well as the associated MIDI controller and the channel:



Ensure that you exit the Learn Mode by clicking on the Learn button again **(1)**, and it will return to its normal gray color.

If you already know what MIDI message you need and/or don't have immediate access to your controller to use Learn mode, you can define your message manually by clicking on the **Edit button (4)**:



Simply select a MIDI device, MIDI message type and other parameters (depending on the message type, e.g. Note, Channel, CC number) and click the OK button to confirm.

Clicking the **Clear button (5)** clears the current controller assignment.

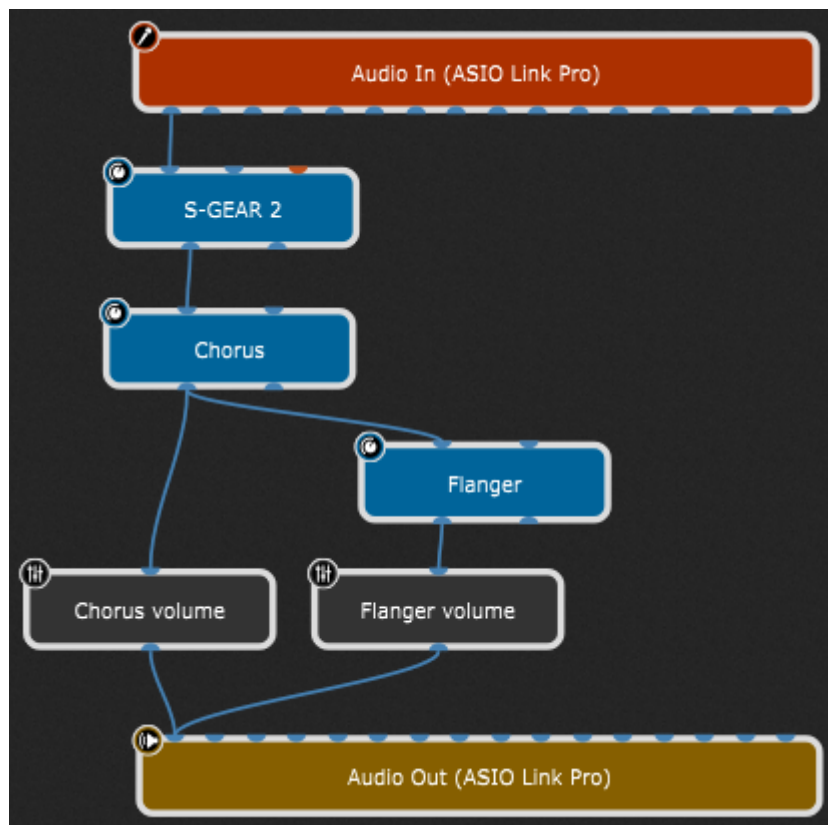
**Note:** if you assign a MIDI message to a widget in the [Global rackspace](#), and you assign the same MIDI message to a widget in a local rackspace, the local rackspace will override



the Global rackspace, which will not respond to the particular message while the local rackspace is active. If you switch to a different rackspace that does not use that MIDI message - the Global rackspace assignment will work again.

### 6.2.7 How to add a widget to a Widget Link group?

The concept behind Widget Link groups is simple: any widget assigned to a Link group can be controlled by any or all other widgets assigned to the same group. This enables you to change multiple plugin parameters together. For example, take a look at this simple layout (refer to [this chapter](#) for more information on how to insert and connect plugins):



Chorus volume and Flanger volume are [Gain Control \(mono\)](#) plugins. You can easily create a Widget Link group to crossfade between the chorus and flanging effects.

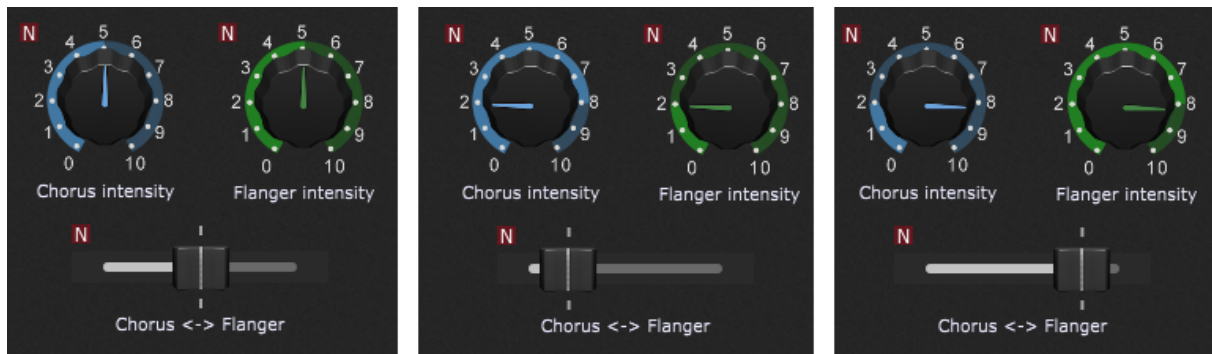
First, we insert three widgets: a Plastic Blue Knob, a Plastic Green Knob and a Black Slider (Horizontal) and then map the Plastic Blue Knob to the Gain parameter in the Chorus volume plugin and map the Plastic Green Knob to the Gain parameter in the Flanger volume plugin. The Black Slider (Horizontal) will not be mapped to any of the plugins, and serves only to display the chorus/flanger ratio better (see [this chapter](#) for more information on how to map a widget to a plugin parameter). **Note:** you can assign a

widget to a Widget Link group without mapping it to a plugin parameter; you can add the mapping later.

Customize the caption of the Plastic Blue Knob, the Plastic Green Knob and the Black Slider (Horizontal) widgets to "Chorus intensity", "Flanger intensity" and "Chorus <-> Flanger" respectively (see [this chapter](#) for more information).

Make sure that you click on the Reverse button in the Value tab in the [Widget Properties inspector](#) for the Plastic Blue Knob widget.

There are 26 different Widget Link groups available in each rackspace (A-Z); to quickly assign all the widgets to the Widget Link group "N", select all the widgets and then press the keyboard key N while holding the **Alt** key (or **Option** on Mac). Alternatively, select the desired widgets and add them to a link group using the [Widget Contextual Menu](#). You can set the Widget Link group for each widget in the Widget Properties inspector separately. If you move the black slider (or any other widget in this group), you'll notice how the widgets' values change:



**Note:** we inverted values for the Plastic Blue Knob, but you can select a different curve for each widget in a group to scale their values (see [this chapter](#) for more information).

Because all three widgets are assigned to the same Link group, they will all move together. When the green knob widget is turned up, it will, as expected, cause the volume of the flanging effect to increase, but turning up the blue knob widget will do the opposite because it is set to reverse value: it will cause the volume of the chorus effect to decrease, and vice versa.

As a result, when you move the horizontal slider from left to right, you'll hear the chorusing effect crossfade into the flanging effect, and vice versa.

There are two other ways to control multiple plugin parameters simultaneously:

- one is to assign the same MIDI control to multiple widgets,
- the other is using [GScript](#) to control any number of parameters with a single widget.

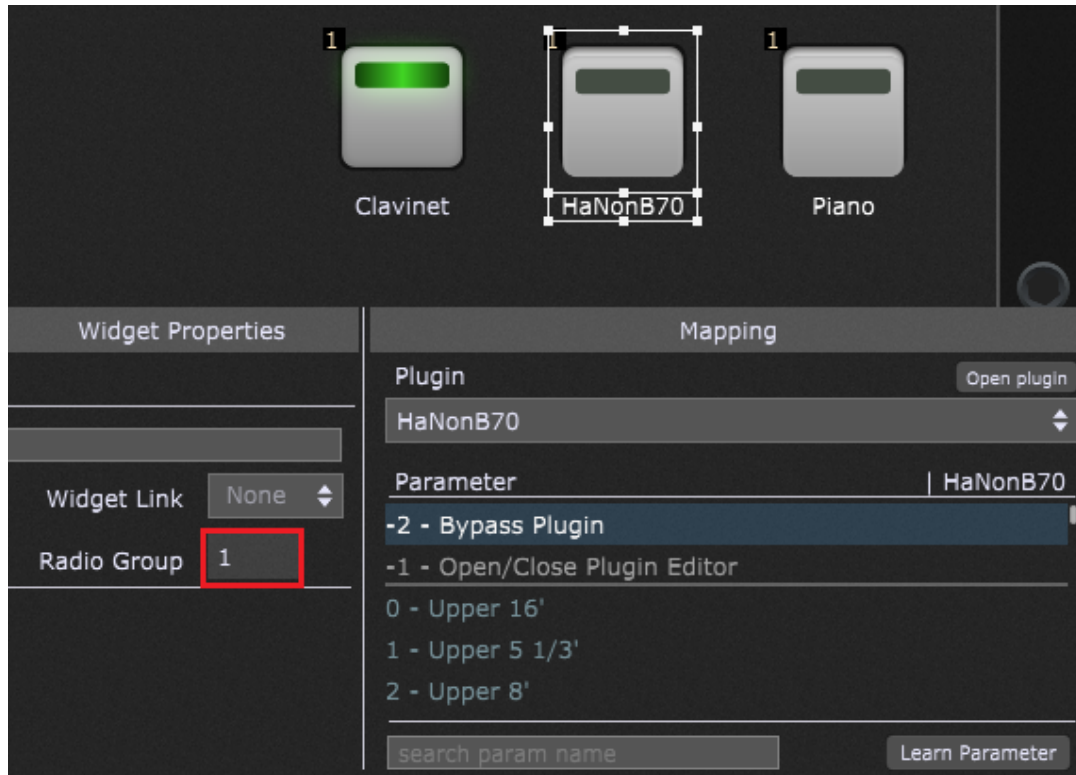
### 6.2.8 How to add a widget to a Radio Group?

A radio group is a widget group where only one widget is turned on at a time. Supported widgets are buttons, LEDs, switches, and the Sustain pedal. You can define up to 99 radio groups. A widget cannot be a member of both radio group and a [widget link](#) group.

There are several main use case scenarios for radio buttons:

1. **Multiple radio buttons attached to the same parameter** — In this case, you use the scale curve to set a specific value for each button. As you pressed different buttons, the parameter value would be changed. In this case, when you press one radio button, you do not want the previous button to send an **Off** value when it pops out - you only want the button you press to send a value - so the option "Send value on on Radio Off" would be disabled in the [Widget Properties inspector](#), the Value tab.
2. **Multiple radio buttons, each one attached to a different parameter, often associated with different plugins** — For example, you might have two radio buttons, each associated with the **Bypass** of two different plugins. In this case, when you press the Bypass on one radio button, you want the other Bypass to be turned off — in that case, you want "Send value On Radio Off" to be enabled because you want the radio button that turns Off to send an Off message to its associated Bypass parameter.

To add, for example, a LED button to Radio Group 1, simply select it in Edit mode and in the General tab of the [Widget Properties Inspector](#), select Radio Group 1 (as indicated with the red rectangle):

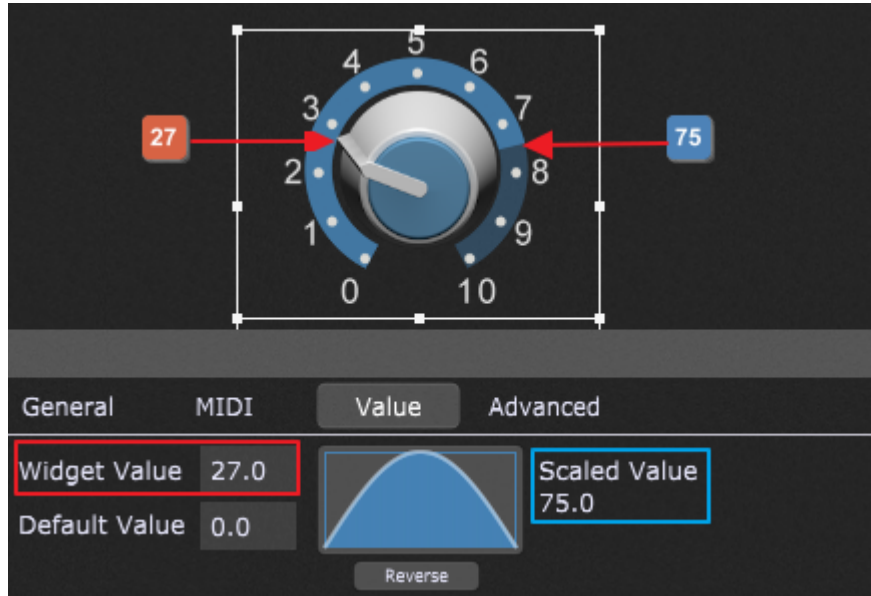


**Tip:** to easily have multiple widgets in the same group, simply add the first widget to the desired radio group and then duplicate it any time you wish - all widgets will be in the same radio group. Afterward, map all widgets to the desired plugin parameters (refer to [this chapter](#) for more information). Alternatively, select the desired widgets and add them to a radio group using the [Widget Contextual Menu](#).

To remove a widget from a radio group, ensure the Radio Group value is 0.

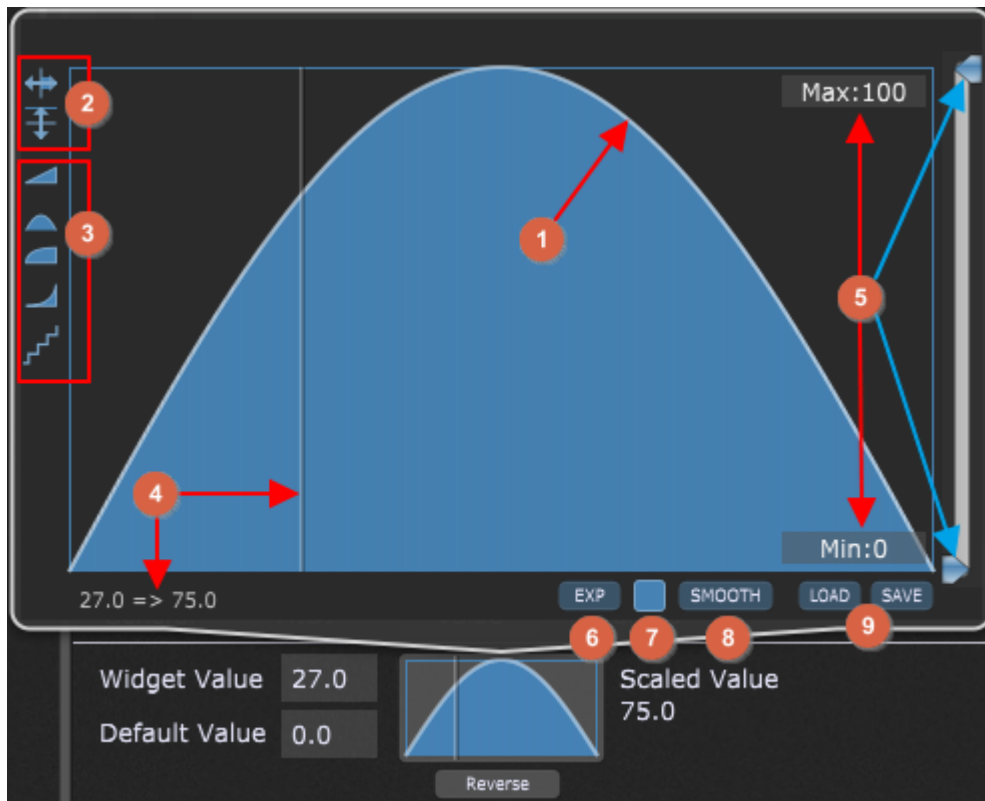
### 6.2.9 How to scale widget values with curves?

The scaling component was mentioned in the [Widget Properties inspector](#) chapter and also other basic terms i.e. the widget value and the scaled value. In the screenshot below, you can see how these values are represented on the Blue Knob widget:

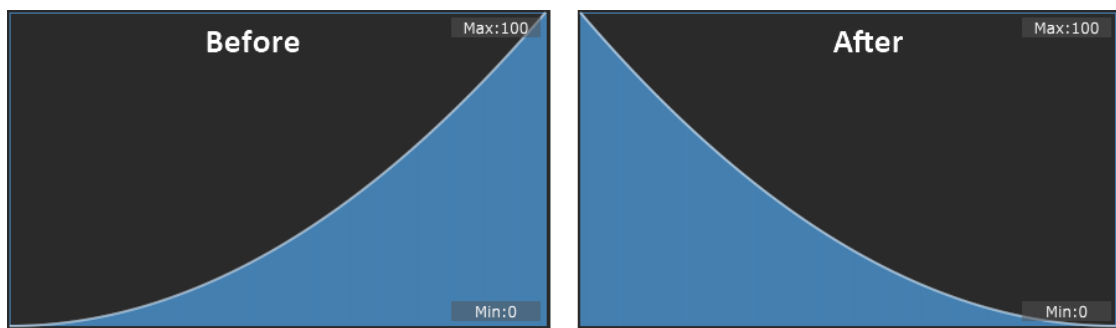


Here it is important not to confuse the **widget value** with the **scaled value**. The widget value is used as the input to the scale component and it is not sent to the plugin parameter. The scaled value is the output from the scale component, which is the value sent to the plugin parameter.

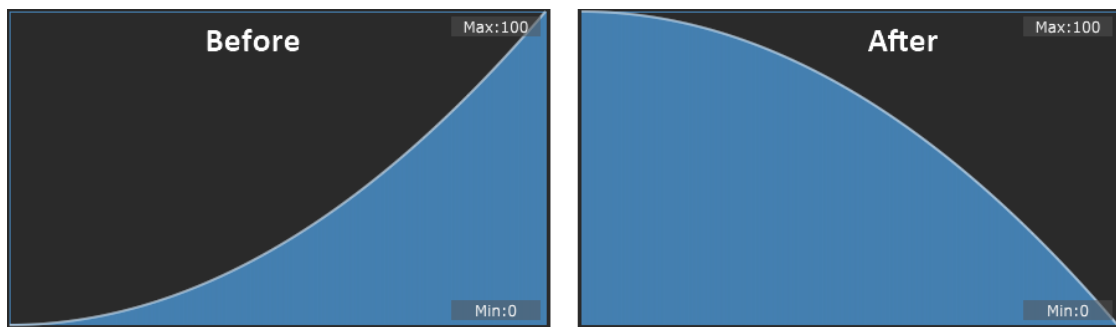
Clicking on the scale component brings up the following window:



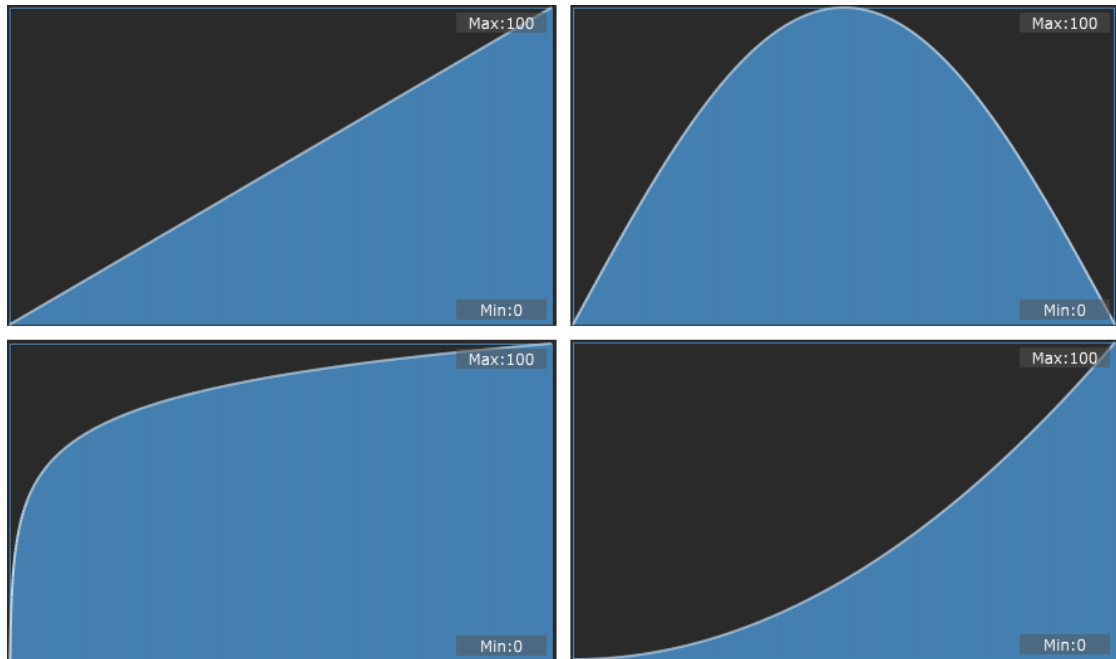
1. **Curve** - the scaling curve is used to transform the widget values into scaled values. To define a curve:
  - a. select one of the supplied basic shapes **(3)** (you can also reverse or invert them),
  - b. click and hold a mouse to draw a custom shape (like in **(8)**),
  - c. use the Curve Designer **(6)**.
2. **Reverse and Invert buttons** - click on the Reverse button to reverse the curve (invert it horizontally), for example:



Click on the Invert button to invert the curve vertically, for example:

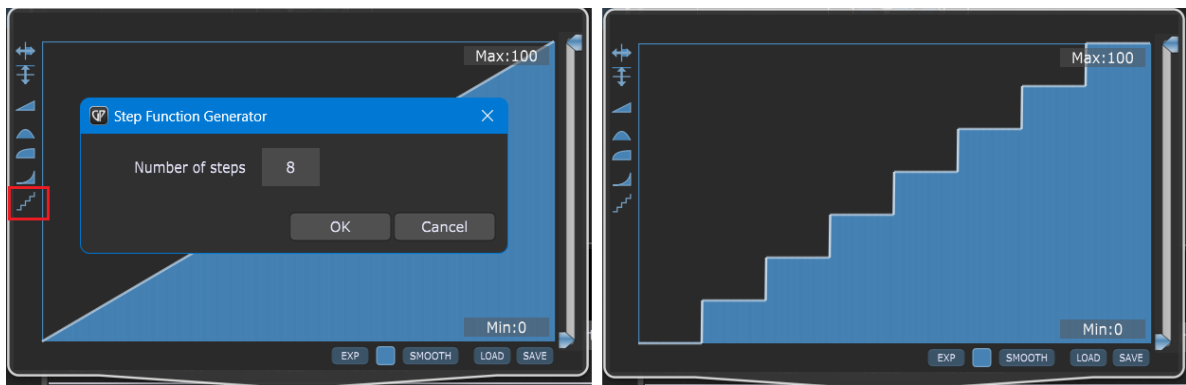


3. **Basic curve shapes** - the provided shapes are linear, sine, Log, and  $X^2$ :



Simply click on the desired shape to apply it.

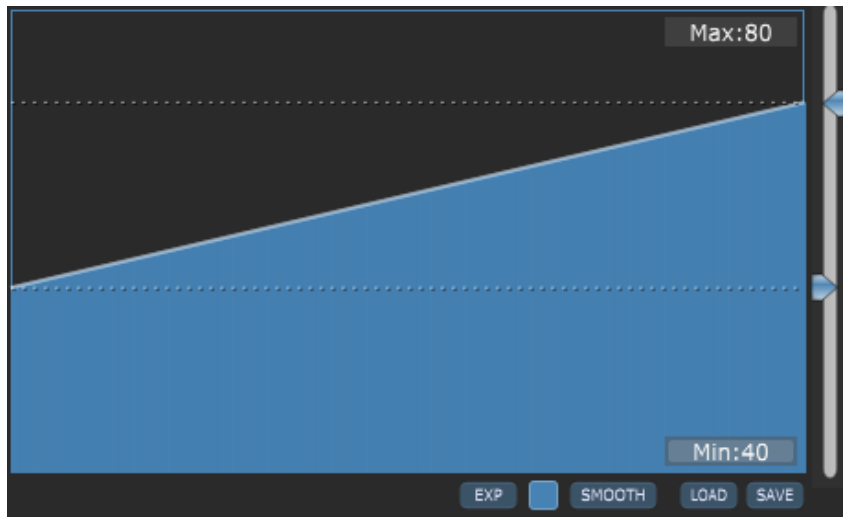
You can also create a step function. Click on the icon indicated with the red rectangle and then specify the desired number of steps:



NB: in Gig Performer 5.0.28 and earlier, you can access this feature by clicking on the linear curve icon while holding down the **Shift** key. To learn more about this feature, and see examples, check out [this post](#).

4. **Indicator and values** - if you hover with your mouse over the curve, a vertical indicator will appear, and the corresponding value will be displayed: **Widget Value** => **Scaled Value** (e.g. 27 => 75, as shown in the screenshot). The shape of the curve defines what value will be output for any widget value. The vertical indicator lets you easily see what values will be used.

5. **Max and Min fields** - you can set a maximum and a minimum value for your curve by entering the desired values in the Max and Min fields or by adjusting the sliders (indicated with the blue arrows). Besides knobs, defining Min and Max values is also useful for the momentary widgets, i.e. buttons or switches. Although their widget value can be only 0 or 1, you don't necessarily want the "on" to always set the plugin parameter to 1.0. For example, you might want the button to cause the plugin parameter to toggle between 0.4 and 0.8 (you can define a min and max volume between which you could toggle):

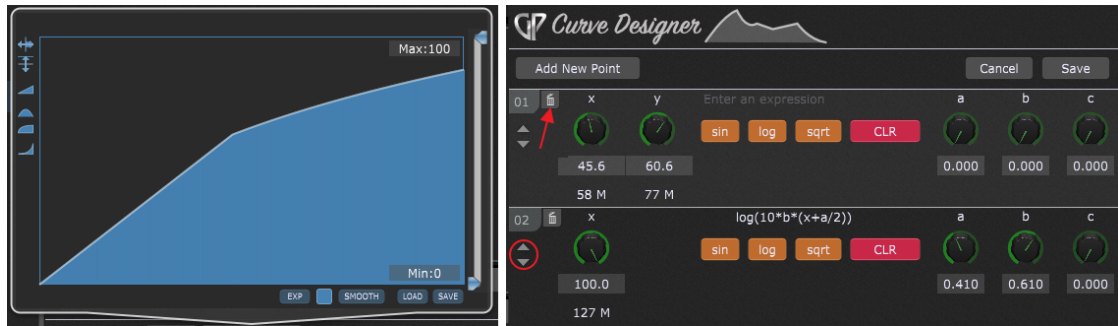


You can also directly enter values using MIDI numbers in the Min and Max fields by adding the letter "m" to the end of the value. For example, if you type in "64m", Gig Performer will interpret it as an exact value of 50.39 (roughly 50% of the maximum MIDI value of 127).

**Note:** holding the **Alt** key (or the **Option** key on Mac) while moving Min or Max sliders allows you to fine-change the values. Holding the **Shift** key while moving min or max sliders allows you to move both sliders simultaneously.

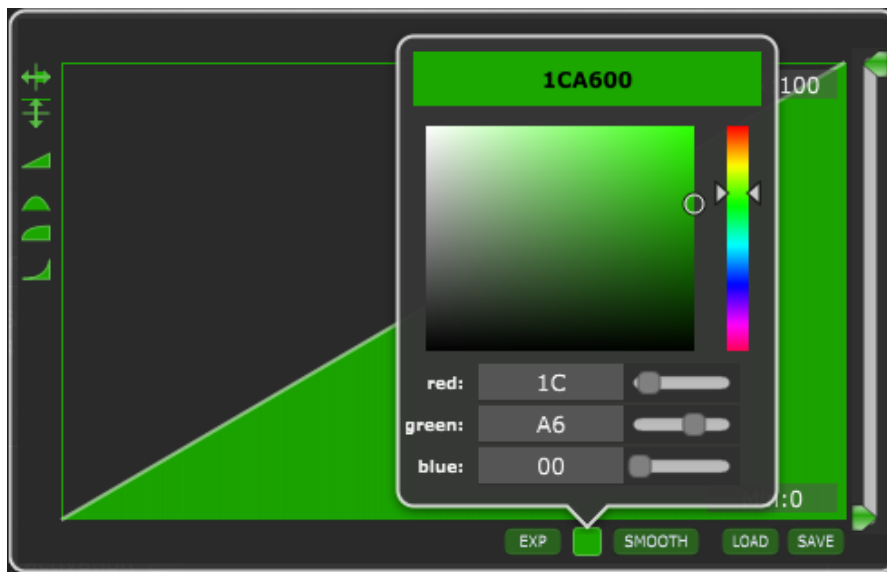
6. **Curve Designer** - clicking on the EXP button opens the Curve Designer:



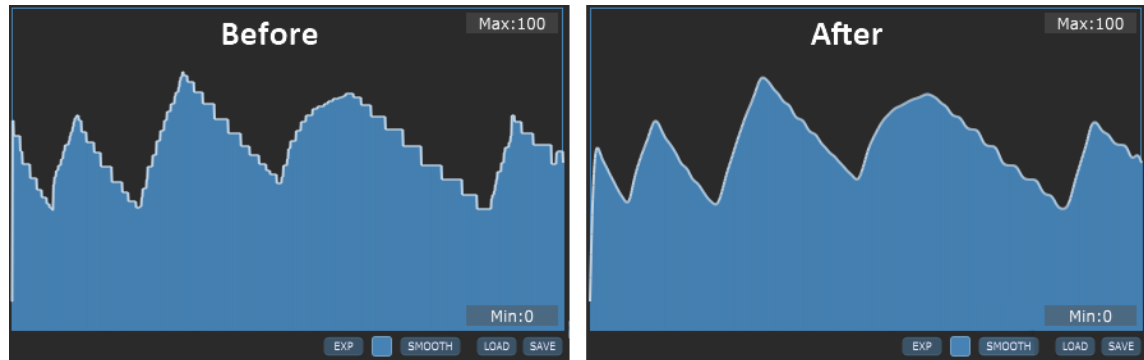


Type in the desired function or click one of the provided buttons (sin, log, or sqrt) to insert a supplied function. You can also use other functions, such as cos, tan or cot. Move the provided knobs (x, y, a, b, or c) or type in the values directly to adjust your curve. If you want to create a curve with multiple points, simply click on the **Add New Point** button and enter the desired function. Click on the Trash can button (indicated with the red arrow) to remove the point. Click on the up/down arrows (indicated with the red circle) to move the point up or down. When you finish adjusting the curve, click on the **Save** button.

7. **Color selector** - click on this button to choose the desired color for the scale curve:



8. **Smooth button** - clicking on this button smooths the curve a little bit:



You can click multiple times to continue smoothing the curve.

9. **Load and save buttons** - clicking on these buttons allows you to load a previously saved gpscale file or to save scale values to a [gpscale file](#). This file can also be used in the scaling component found in the [MIDI In](#) block or the [Probabilistic Sound Designer](#).

**Tip:** holding the **Alt** key (or the **Option** key on Mac) while creating a curve with your mouse constrains the horizontal position, thereby producing a flat horizontal line as you move the mouse.

### 6.2.10 How to control widgets from the Global rackspace in a regular rackspace?

In this chapter you will learn how to configure global and local widgets so that changing widget values in a local rackspace also change widget values in the Global rackspace (and vice versa).

First, create a simple layout in the [Wiring view](#) in the [Global rackspace](#); for this example, we assume that you will use an [Audio Mixer \(4ch\)](#) in your configuration. While you are still in the Global rackspace, open the Edit mode in the [Panels View](#) and insert two knobs (e.g. a Green knob and a Blue knob):



Click on the Green knob; note that the [Widget Properties inspector](#) includes one additional setting for the global widgets: Global Parameter Assignment. Select a value from the drop-down menu (from Param 0 to Param 127), e.g. **Param 1** (as indicated in the red rectangle). In the mapping section, choose the **Audio Mixer (4ch)** plugin (as indicated in the blue rectangle) and select the **Channel 1-2 volume** parameter.

Click on the Blue knob; in the Widget Properties inspector, select a different Global Parameter assignment, e.g. **Param 127**. In the mapping section, choose the **Audio Mixer (4ch)** plugin and select the **Channel 3-4 volume** parameter.

Now switch to a local rackspace (click on View -> Show/Hide Global Rackspace), and in the Wiring view, insert the [From Global Rackspace](#) block. Then switch to the Panels view, Edit mode and insert two knobs (e.g. a Plastic Green knob and a Plastic Blue knob):



Global parameters are exposed to a local rackspace as parameters of the [From Global Rackspace](#) plugin (alternatively, you can also use the To Global Rackspace block, provided that you had inserted it in the Wiring view). Therefore, in the Mapping section for both knobs, select "From Global Rackspace" (as indicated with the blue rectangle).

Select the **Channel 1-2 volume** parameter for the Plastic Green knob and the **Channel 3-4 volume** for the Plastic Blue knob.

**Note:** the actual parameter position in this list depends on what you selected in the above-mentioned Global Parameter Assignment setting. Since Param 1 and Param 127 were selected for the green and blue knob, respectively, these positions become available in the From Global Rackspace plugin. If you want to prefix parameter positions in these

parameter lists, enable the *Show parameter numbers in widget parameter selection list* toggle button in [Display Options](#).

Finally, switch to the Panels view and move local knobs - the corresponding global knobs will move, as well (and vice versa).

**Note:** if the *Global Parameter Assignment* parameter position (e.g. Param 126) is not assigned to a global widget, the From Global Rackspace block will display "<not assigned>" for that position (see the screenshot above). If a global parameter position is assigned, the parameter name that you will see in the From Global Rackspace block depends on the global widget settings in the Widget Properties Inspector:

GLOBAL WIDGET		Customize caption		
		Off	On, blank	On, custom_text
Plugin mapping	(no plugin)	<no name>	<no name>	custom_text
	Mapped, parameter not selected	<no name>	<no name>	custom_text
	Mapped, parameter selected	parameter_name	parameter_name	custom_text

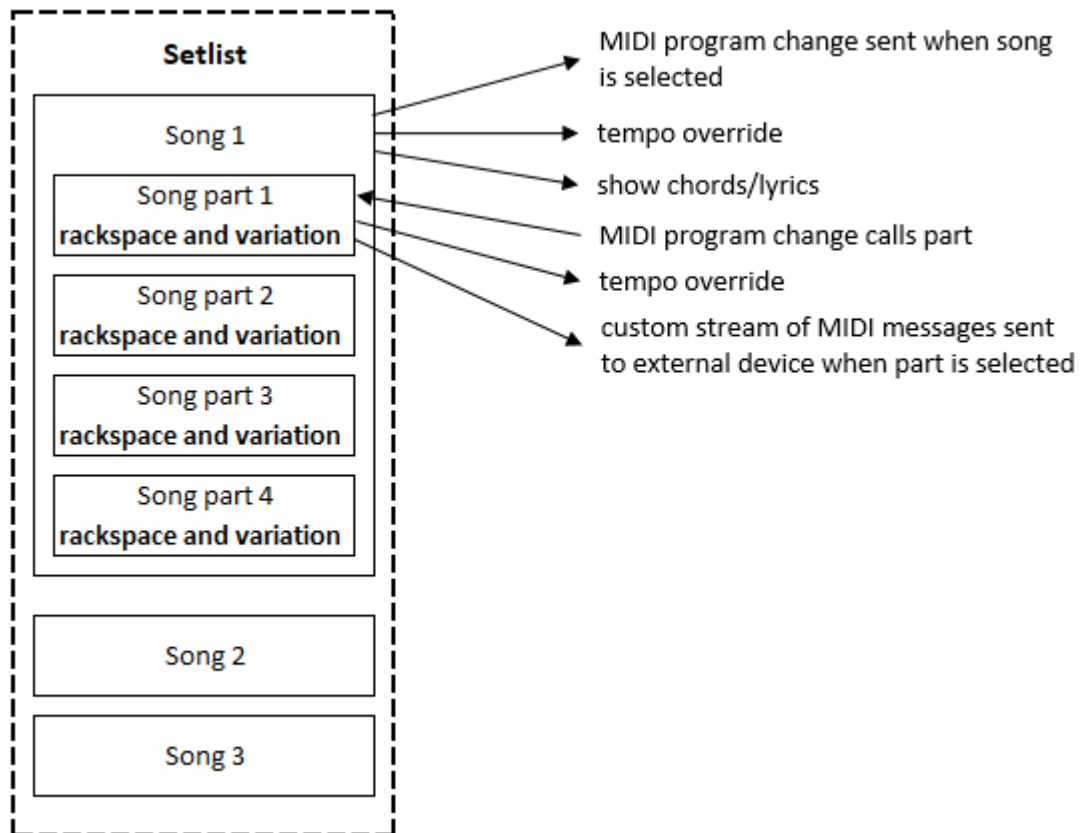
You can copy widgets from the Global rackspace and then paste them into a local rackspace using the Copy and Paste menu options. Suppose a global widget is mapped to a parameter in the plugin in the Global rackspace. In that case, the pasted widget will be automatically mapped to the actual parameter values of the widgets in the Global rackspace. If it doesn't already exist, this action will automatically insert a "[To Global Rackspace](#)" plugin into the local rackspace.

## 7 Songs and Setlists

In this chapter, you will learn more about creating songs, song parts and setlists in Gig Performer, how variations are related to song parts, and more.

### 7.1 Rackspaces/variations and Song Parts relationship

Gig Performer allows you to quickly and easily organize all of your rackspaces and variations to meet the needs of your performances by letting you build setlists of your songs. Each song consists of one or more song parts, such as verses, choruses, bridges, interludes, and so forth:



A song part is basically a reference to a specific variation in some rackspace that will be used when the part is selected, though there are other song part settings, as well:



The screenshot above shows how two song parts (Verse and Chorus) refer to two variations (Basic and Delay+Reverb), respectively.

**Note:** Gig Performer remembers the variation selected when you switch away from a song part, so if you are trying different variations, be sure that the correct variation for the song part is selected before you switch away.

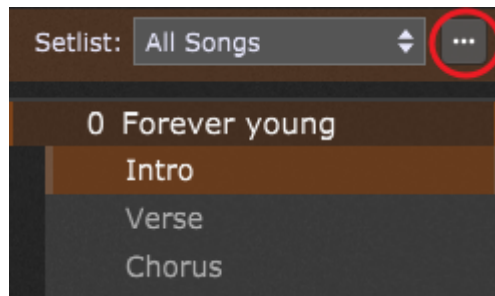
In the next chapter, you will learn how to create a song.

## 7.2 How to create a song?

To create a new song, if you do not already have a suitable rackspace to use, first create a layout in the Wiring view (refer to [this chapter](#) for more information on how to insert and

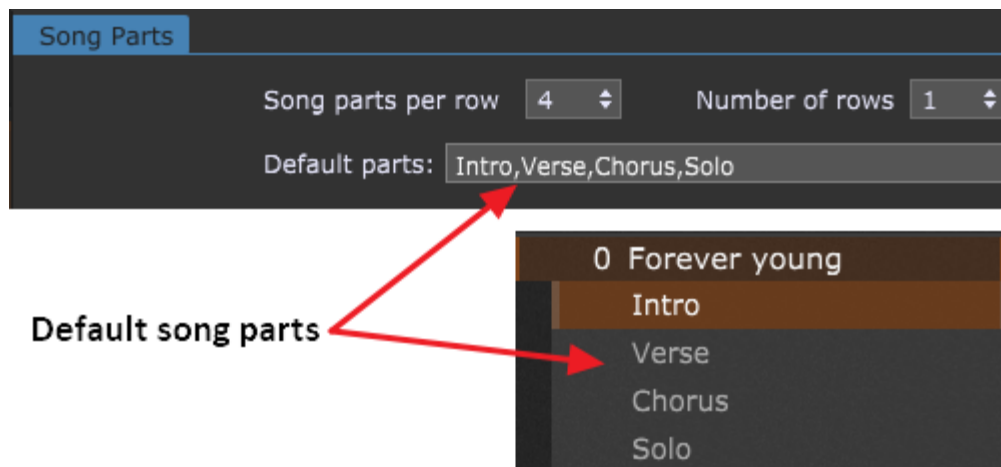
connect your plugins); in this [blog article](#), you can see an example setup, or use one from the built-in templates.

Afterward, switch to the [Setlist view](#) and press **Ctrl+Shift+N** (or **Cmd+Shift+N** on Mac) or choose the **Create New Song...** menu item from the Setlists [contextual menu](#). The [Song Properties](#) dialog appears; enter a name in the dialog that appears (i.e. Forever young) and click on the OK button. Notice that three song parts are automatically created for your song - Intro, Verse, and Chorus:



You can easily create new songs/song parts, rename or remove the existing ones, and perform other operations by clicking on the "More..." button (indicated with the red circle) or by right-clicking on a song/song part; learn more about supported operations in the [Manage setlists contextual menu](#) chapter. Also, rearrange song parts by simply dragging them to positions you prefer.

**Note:** you can set the default parts created with a new song by choosing [Setlist/Song Settings](#) from the Options menu:



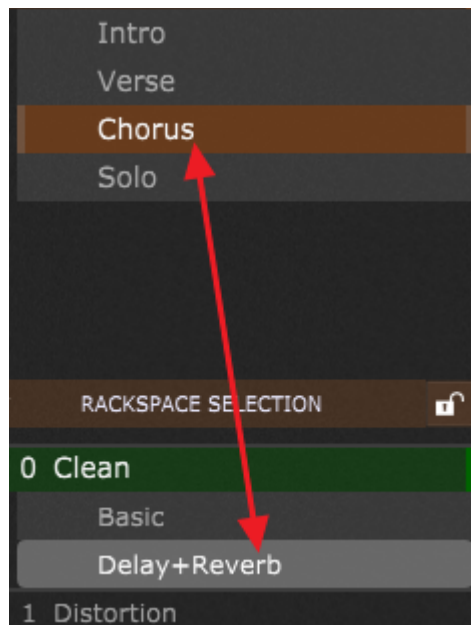
In the screenshot above, the Solo song part has been added (make sure to separate song parts with the comma or the semicolon).

The new song has also been added to the All Songs list.

Ways to switch to a song part:

- Click on a tile in the part bar at the top of the Setlist/Song view.
- Click on the part name in the Setlist browser.
- Select a song part with a MIDI program change message assigned in the [Song Part Properties](#) dialog. Refer to the blog article [Program Change Management](#) to learn more about the program change messages.
- Use the up/down arrow keys on your computer keyboard to select the song part above/below the current one respectively.
- Use parameters of the [System Actions](#) plugin such as *Next Song*, *Next Songpart*, *SelectPartOrVar*. Refer to the System Actions chapter to learn more.
- Define actions in the timeline using the [Action Management](#) window.

Click on the variation you want attached to the song part to select it:



**Note:** [click here](#) to find out more about how song parts and variations are related.

When you select a song, the first song part will be active. Note that rackspace variations have different behavior (if explicit PC numbers are not used in variations, the last selected variation will be called when using the PC for that rackspace).



Gig Performer supports the ChordPro lead sheet file format with both an editor/viewer to enable creating, editing, and viewing ChordPro documents, and the ability to attach a ChordPro document to a song, so that selecting the song opens the ChordPro document to put the chords and lyrics right in front of you. Refer to the [ChordPro](#) chapter for more information. Make sure also to check this blog article, [How to use ChordPro files in Gig Performer](#), for detailed guidelines.

Make more songs and song parts: create new songs, song parts, and assign ChordPro files to your songs by following the guidelines above.

**Note:** although a song can have as many parts as you need, you can only assign MIDI events to the first 24 parts. [Click here](#) to learn more.

In the next chapter, you will learn how to create a setlist of your songs.

### 7.2.1 Song Properties

When you create a new song or double-click on an existing song (or select Song Properties... from the [contextual menu](#)), the Song Properties dialog appears:

Name: Black Swan Rider  
Artist: Tornado  
Key: C major  
Transpose: 0 semitones  
Override global tempo  
Tempo: 120.0 Time Signature: 4 / 4  
Send program change number when song activates  
PC # 0  
Bank # 0 MSB 0 LSB  
Lyrics/Chord (ChordPro) File: <none> Clear I  
Cancel OK

**Name** - type in the song name.

**Artist** - type in the name of the artist/band who performed the song.

**Key** - choose the key center (root note) and quality (major/minor) of the song from the drop-down menus. These entries are for your information only.

**Transpose** - click and select a transpose value in semitones.

**Tempo** - allows you to set a tempo for the song. Turn on the Override toggle button to make this active tempo when the song is selected.

**Time signature** - allows you to set the desired time signature for the song; click on the numerator, select the desired value and then choose a value for the denominator from the drop-down menu. Turn on the Override toggle button to make this the active time signature when the song is selected.

**Override global tempo** - when this toggle button is turned on, the selected song tempo will override the global tempo. This tempo will be applied to all song parts of the current song if they do not have their own overrides - a tempo for a song part (specified in the [Song Part properties](#) dialog), if turned on by its Override toggle button, has a higher priority and will override the song tempo.

**Send program change number when song activates** - clicking on this toggle button enables you to send a MIDI program change number when the current song activates; set the desired program change number and bank (MSB and LSB) values. When the song is selected, the program change number is sent out using the MIDI Device and Channel selected in the [Setlist tab](#) of the Options dialog.

**Lyrics/Chord (ChordPro) File** - displays a file in ChordPro format. Use the Song Lyrics/Chords viewer to load your lyrics/chords file for this song. Gig Performer will automatically remember and auto-load that file when you switch to this song again. Refer to the [ChordPro](#) chapter for more information. **Note:** files can be loaded only from the root directory specified in the [Locations tab](#) of the Options dialog. If the file you want to use is not in that directory use the File -> Import ChordPro Files... command.

**Clear** - clicking on this button clears the association to the ChordPro file.

**"i": (information)** - clicking this button opens a dialog stating that ChordPro documents must be opened and assigned in the Song Lyrics/Chords Editor.

### 7.2.2 Song Part Properties

When you double-click a song part or select Song Properties.. from the [contextual menu](#), the Song Part Properties dialog appears:

Name: Solo

Override global tempo  
Tempo: 200.0 Time Signature: 4 / 4

Assign program change number for this song part  
PC # 1  
Bank # 0 MSB 0 LSB

MIDI messages to send when this song part gets activated  
Edit... Send Now

Actions... 1 Cancel OK

A song part consists of a rackspace variation plus a collection of properties. The variation may be as stored, or it may have been edited just for that song part.

**Name** - displays the song part name (type another one if preferred).

**Tempo** - allows you to set a tempo for the song part. Turn on the Override global tempo toggle button to activate this tempo when this song part is selected.

**Time signature** - allows you to set the desired time signature for the song part; click on the numerator, select the desired value, and then choose a value for the denominator from the drop-down menu. Turn on the Override global tempo toggle button to activate this time signature when this song part is selected.

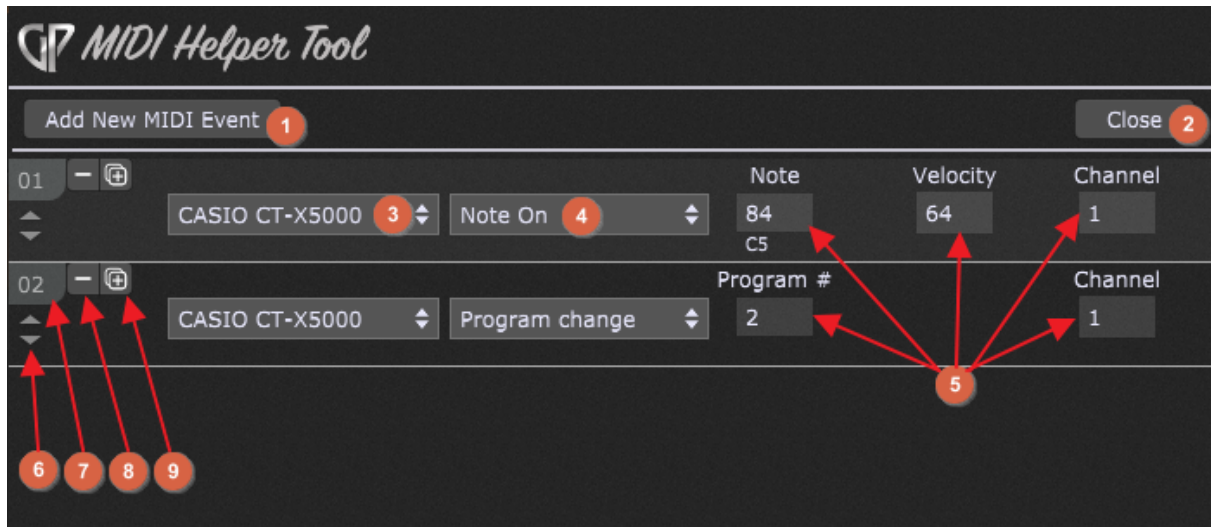
**Override global tempo** - when this toggle button is turned on, the selected song part tempo will override both the global tempo and the song tempo (see the [Song Properties](#) dialog for more information). Hence, the Song part tempo is the highest priority.

**Assign program change number for this part** - clicking on this toggle button enables you to assign a MIDI program change number to the current song part; set the desired program change number and bank (MSB and LSB) values.

**MIDI messages to send when this song part gets activated** - allows you to send MIDI messages to one or more physical MIDI Out devices. Click on the **Edit...** button to open

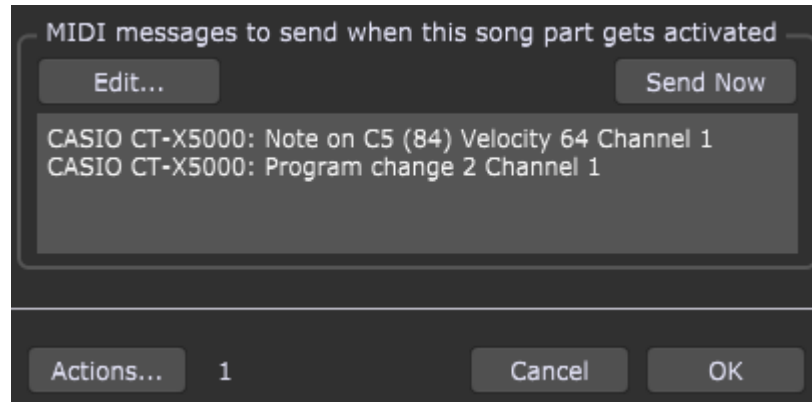
the **MIDI Helper Tool** and click on the **Send Now** button to send the selected MIDI messages to the selected physical MIDI out devices.

The MIDI Helper tool allows you to specify an arbitrary sequence of MIDI events to be sent out to MIDI Out devices:



1. **Add New MIDI Event** - clicking on this button adds a new MIDI event placeholder.
2. **Close** - clicking on this button simply closes this window.
3. **MIDI Output device** - click on this drop-down list to choose a MIDI Out device.
4. **MIDI event** - click on this drop-down list to choose the desired MIDI event. Check [this chapter](#) for a list of events.
5. **Event parameters** - depending on the type of the event, there are different parameters (or no parameters at all) for an event.
6. **Navigation arrows** - click the up/down arrows to move the current event placeholder up/down.
7. **Placeholder number** - simply displays a number for the current MIDI event.
8. **Remove event** - clicking this button removes the current MIDI event.
9. **Duplicate event** - clicking this button duplicates the current MIDI event.

When you finish adding your MIDI events, click on the close button; selected events can be seen in the list:



**Actions...** - clicking this button opens the Action Management window:

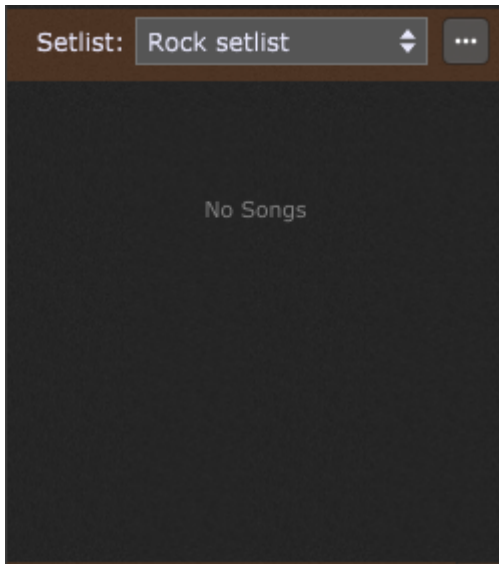


This window allows you to define actions that will be executed when this song part is activated. Refer to [this chapter](#) to learn more about the Action Management window and its elements.

**Note:** if you add the **Set Global BPM** action, it will override the **Override global tempo** setting described above in this chapter.

### 7.3 How to create a setlist?

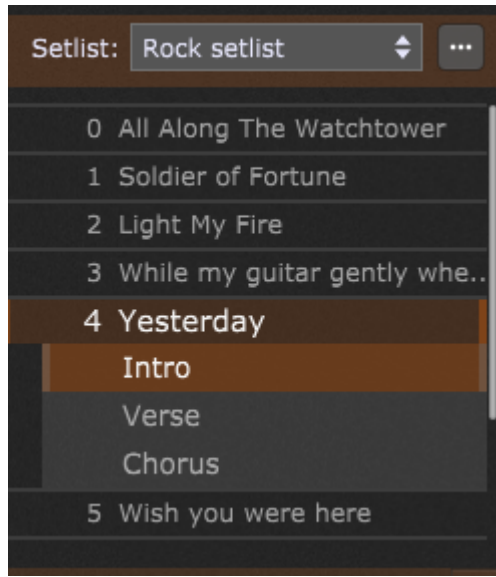
When you [create a song](#), it is automatically added to the All Songs list. To create a setlist, choose the Add new setlist... command in the [Setlists contextual menu](#), or New Setlist... from the Edit menu. Type in a name in the Setlist Name dialog that appears (i.e. Rock setlist) and press Enter:



The newly created setlist is automatically selected but with no songs. To add songs to the setlist, open the **Add songs to setlist** window either by pressing **Ctrl+Shift+F** (or **Cmd+Shift+F** on Mac), by choosing the **Add Song(s) to Setlist...** command in the Setlists contextual menu or by choosing the **Add Song(s) to Setlist...** command in the Edit menu:

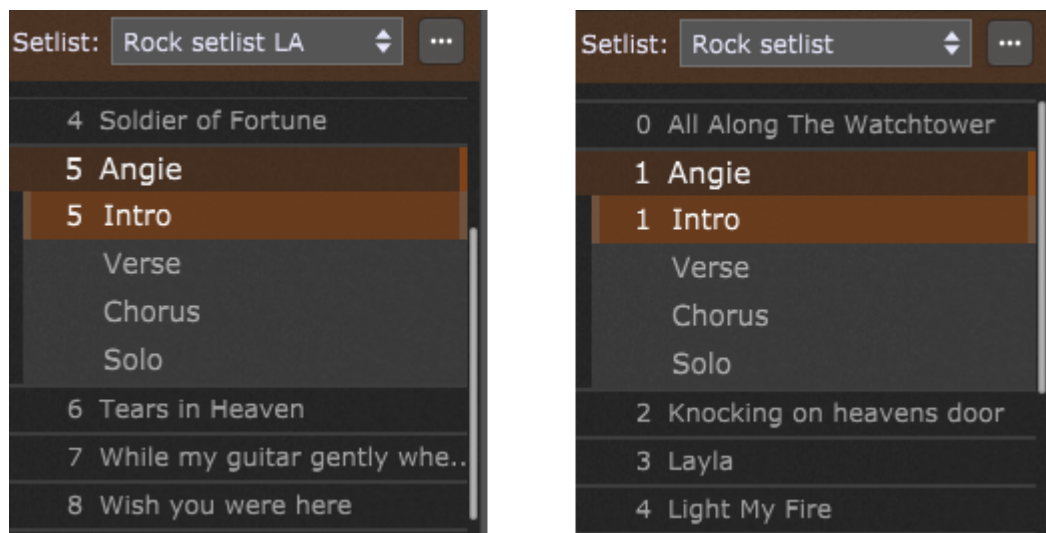


This window lists all available songs. To add a song from this window to the setlist, select it and press the Enter key (or double-click it). Songs that are added to a setlist are automatically grayed out. You can also find a song by entering search terms in this field to filter the view. When you are finished adding songs, press the Escape key to close the window:



You can easily re-arrange songs by dragging them and placing in the desired position. Furthermore, you can sort them by name or by their program change numbers - click on **Edit -> Sort Songs by Name** or **Edit -> Sort Songs by PC Numbers**, respectively.

Setlists can be duplicated and modified for gigs that vary only slightly from a usual song order, or a custom setlist can be created for a specific gig. For example, if tomorrow's setlist is just a little different than this, the quickest thing is to duplicate this setlist and then modify it. Setlist operations (i.e. rename, duplicate, or remove) are easiest to select in the [setlist contextual menu](#); to duplicate a setlist, choose the Duplicate Current Setlist... menu item, type in a name (i.e. Rock Setlist LA) and press Enter:



As seen from the screenshot above, the program change assignment for a song part is specific to the setlist; the same song can have a different program change assignment in

each setlist. This lets you have consistency even when the setlist changes from gig to gig. Refer to the [Song Part Properties](#) chapter for more information on how to assign a program change number to a song part.

Another way to add a current song to a setlist is to right-click on a song, expand **Add This Song To Setlist** and choose one of your defined setlists.

## 7.4 How to capture variation edits?

If you are in a song part and move your widgets (knobs, sliders, etc.), changing their values, and then switch away from that song part - all the changes in that song part will be lost and reset to the values of the underlying variation. For example, look at the screenshots below:



Let's assume that **(A)** shows widget values when you first select a song part. If you want to increase a delay a bit more **(B)** and select another view (i.e. Panels view), the delay value will be reset to **(A)**.

However, Gig Performer provides a mechanism to capture these changes and save them to the currently active song part. So, if you want to retain the amount of delay as shown in **(B)**, click on the **Capture variation** icon (indicated with the red arrow - refer to the [Setlists view](#) chapter for more information):





Clicking on this button allows you to save edits you have made to a variation into the currently selected song part only. **Note:** the changes are not saved to the underlying variation in the rackspace itself.

Notice that additional buttons are available. When the LED is red, it indicates that the song part has saved values different from the underlying variation. Click and hold the COMPARE momentary button to display the original widget values in the corresponding variation; release to see captured changes in the song part (click and hold will display widget positions as shown in (A), and widget positions indicated with (B) will be displayed when you release the mouse).

Click on the **Revert to original variation** button (indicated with the *trash can* icon) to remove captured changes and match the variation values.

If you like how your song part sounds with the increased delay (as shown in (B)), you can apply this change to the underlying variation so that all song parts associated with that variation have this amount of delay. From the [Setlist contextual menu](#), choose the **Push Song Part Changes to Rackspace** menu item. Notice that the LED indicator is no longer red, and the Compare and Revert buttons disappear. Switching to the Panels view shows that the widget's value (i.e. the amount of delay) has been updated and now has the value indicated with (B). All song parts attached to this variation will have a higher amount of delay (except for those with their own captured changes).

**Note:** If you edit values on the fly and like the changes, this is how you update the original variation with the changes. If you want to keep the original variation, duplicate it before pushing song part changes.

**Tip:** The [System Actions](#) plugin provides you with the parameters *TakeSnapshot*, *CancelSnapshot*, and *PushSnapshot* allowing quicker control of managing variation edits.

Check the System Actions chapter to learn more.

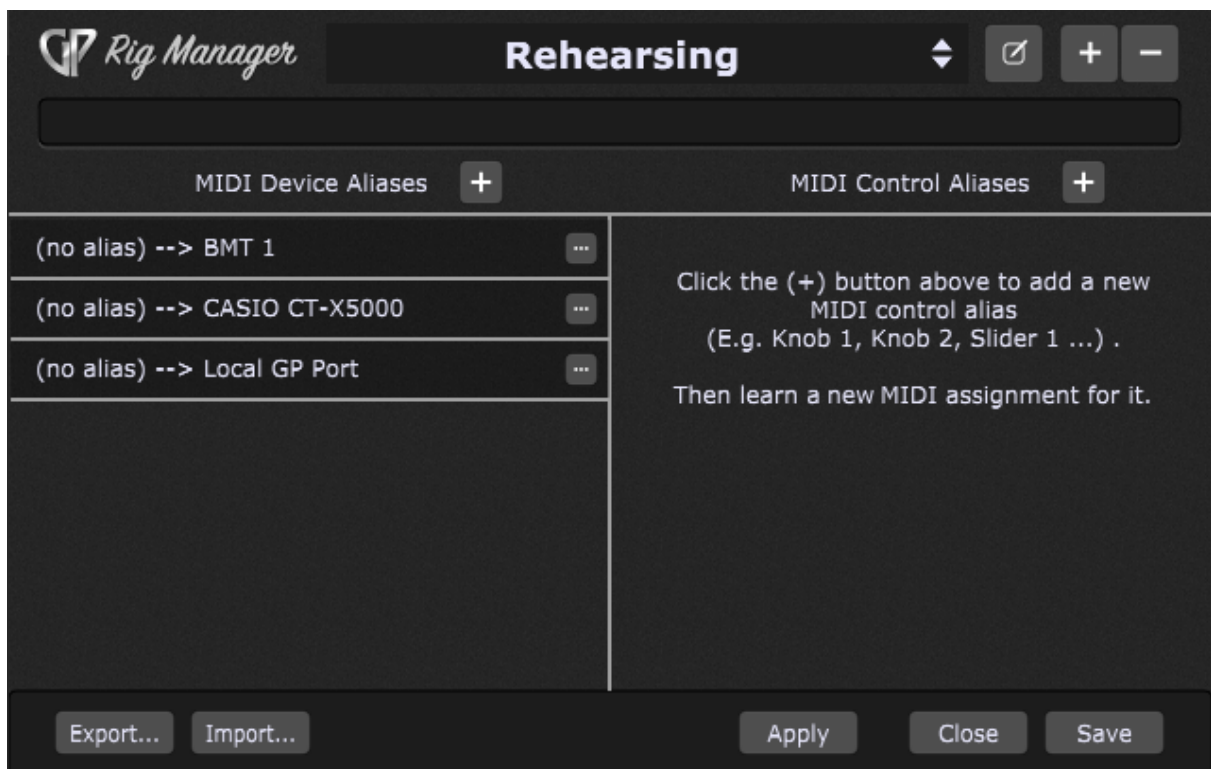
## 8 Rig Manager

Basic Rig Manager concepts are covered in the [Gig Performer user interface and concepts](#) section. In this chapter, you will learn more about the Rig Manager, how to create rigs, define aliases and associate them with your MIDI controllers.

**Tip:** Read [this blog article](#) if you are dealing with back-lined or rented keyboards.

### 8.1 How to create an alias for a MIDI device and a MIDI control?

Let's assume that you start with a clean slate. Open the Rig Manager and rename the default rig to a custom name, i.e. Rehearsing (refer to [this chapter](#) for more information about the Rig Manager window):



The main area of the window is divided into two columns. The MIDI Device Aliases are on the left - the list of aliases for physical control devices like keyboards, pad controllers, and control surfaces. In the screenshot above, none of the connected MIDI devices shown have been given aliases.

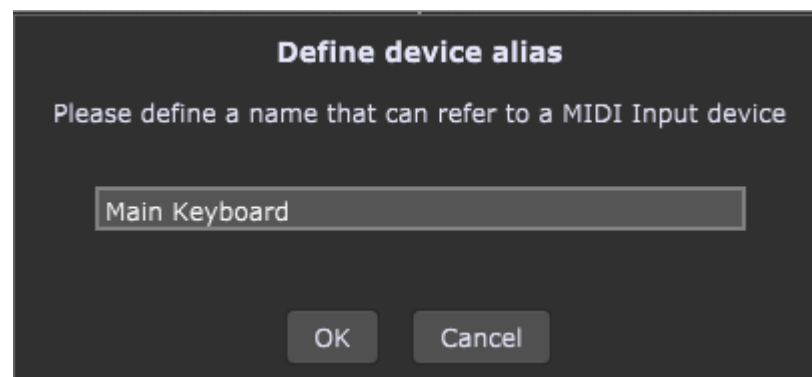
An alias is simply a name you can designate to represent a physical device (on the left) or a control (a knob, slider, button, etc.) associated with some device. If you use aliases for your MIDI In blocks and your widgets in your racksaces instead of the actual devices,

then you can easily switch your entire gig file to use a different set of physical controllers without having to update individual rackspaces.

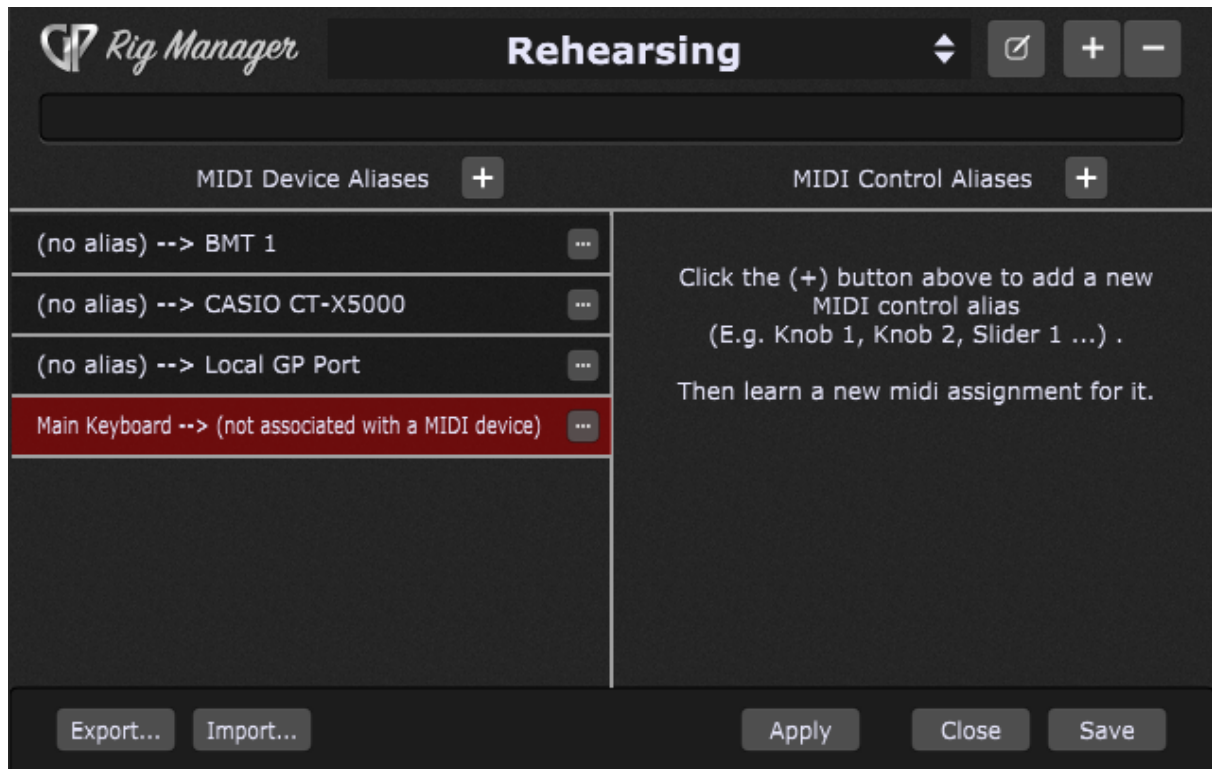
**Note:** the Local GP Port is an advanced feature in Gig Performer. Refer to [this chapter](#) for more information on how to use it.

The column on the right is MIDI Control Aliases - the list of aliases for controls, i.e. sliders, buttons, faders, and knobs. In the screenshot above, no MIDI Control aliases have been defined yet.

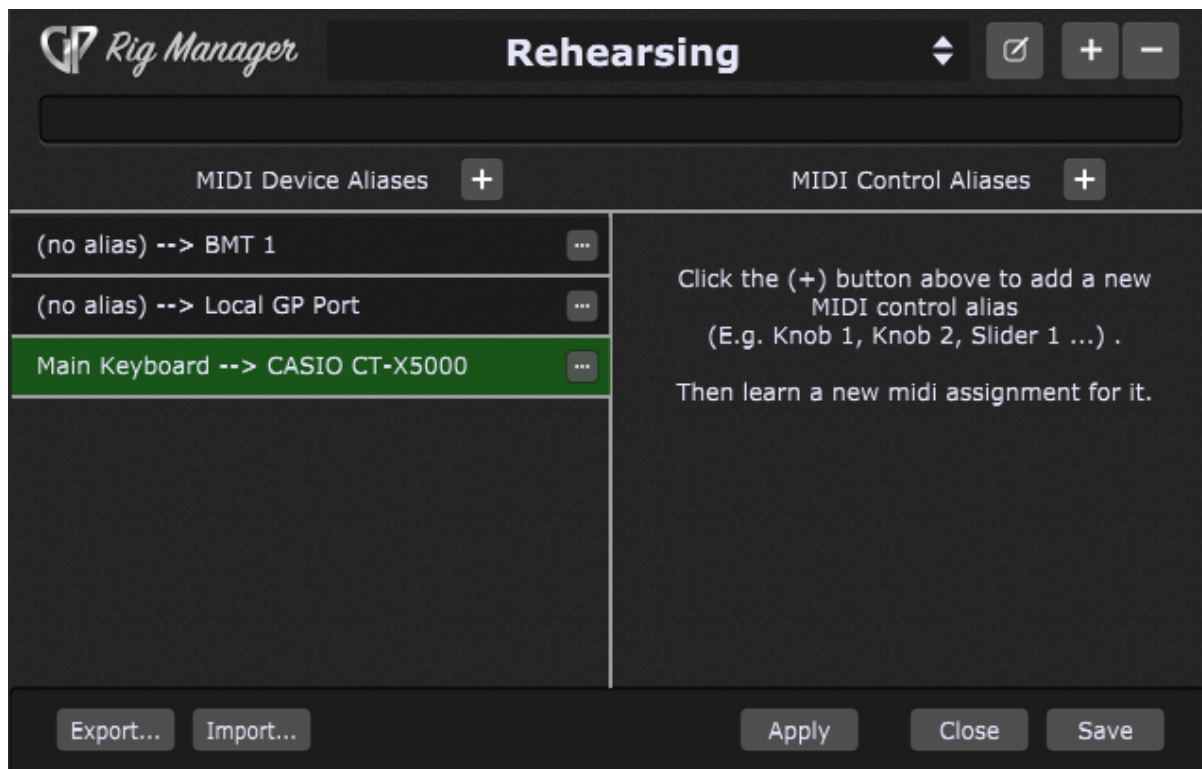
To create an alias for a connected MIDI device, click the plus button at the top of the MIDI Device Aliases list. In the dialog that appears, give the desired MIDI device an alias (i.e. Main Keyboard):



The newly created alias needs to be linked to a physical MIDI device:

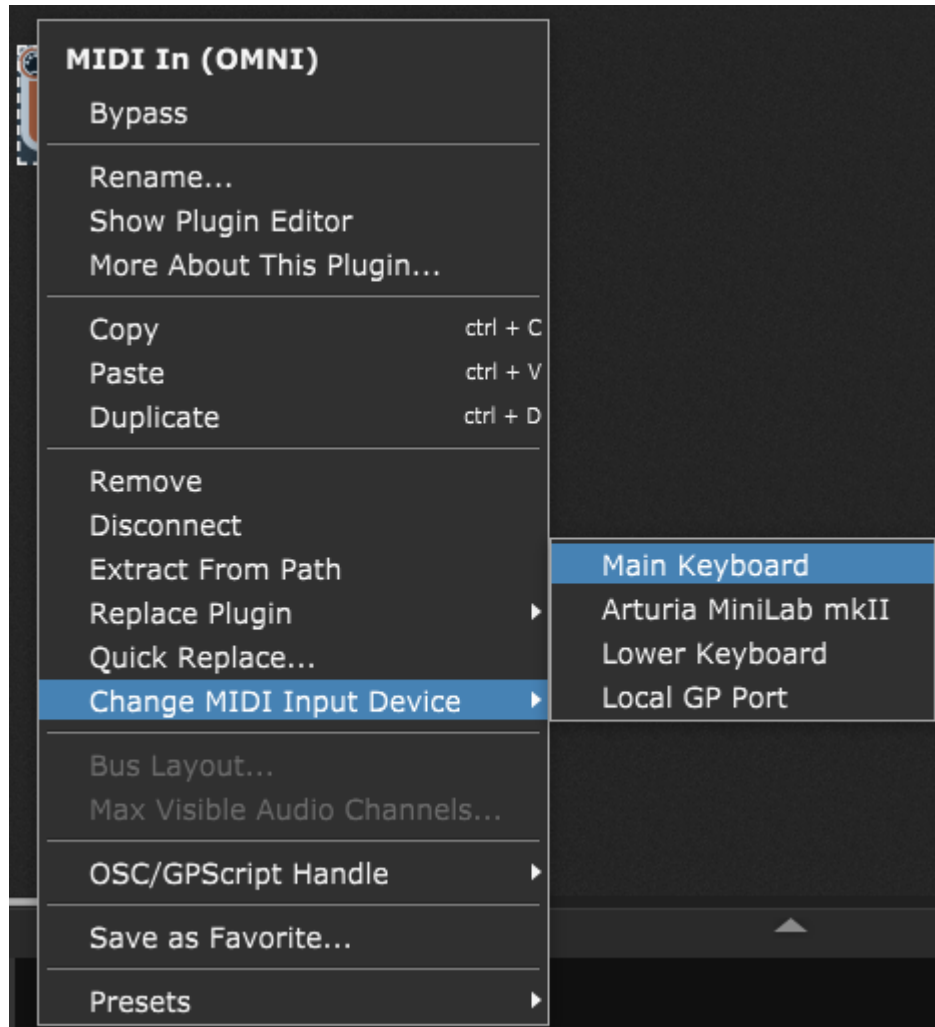


Click on the "three dots" button or right-click on the alias and choose a MIDI device from the Associate MIDI Device sub-menu; alternatively, you can click on the three dots button, then click on the **Learn MIDI device** menu item and use any control on your MIDI device to learn the desired MIDI device (instead of selecting it from the list). After you successfully associate an alias with a MIDI device, it will be highlighted in green:

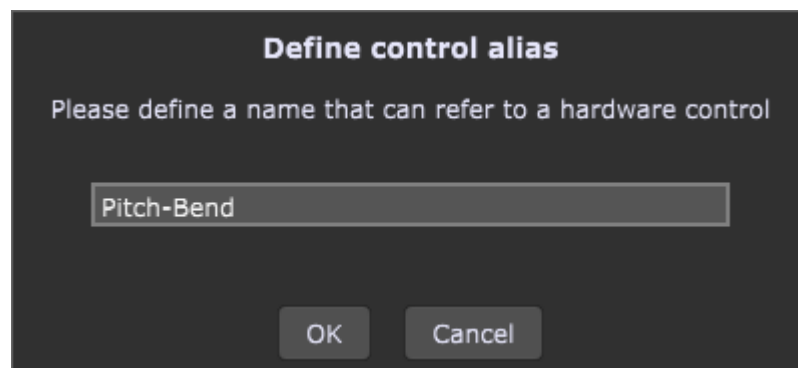


Another way to assign an alias to the specific MIDI device is to click on the "three dots" button for the desired MIDI device and select the **Define alias for this MIDI device...** menu item or right-click the MIDI device and select the **Define alias for this MIDI device...** menu item.

Defined MIDI device aliases are listed in the plugin contextual menu for a MIDI In block:



To create an alias for the desired MIDI control, click the plus button at the top of the MIDI Control Aliases list. In the dialog that appears, give the desired MIDI control an alias (i.e. Pitch-Bend):

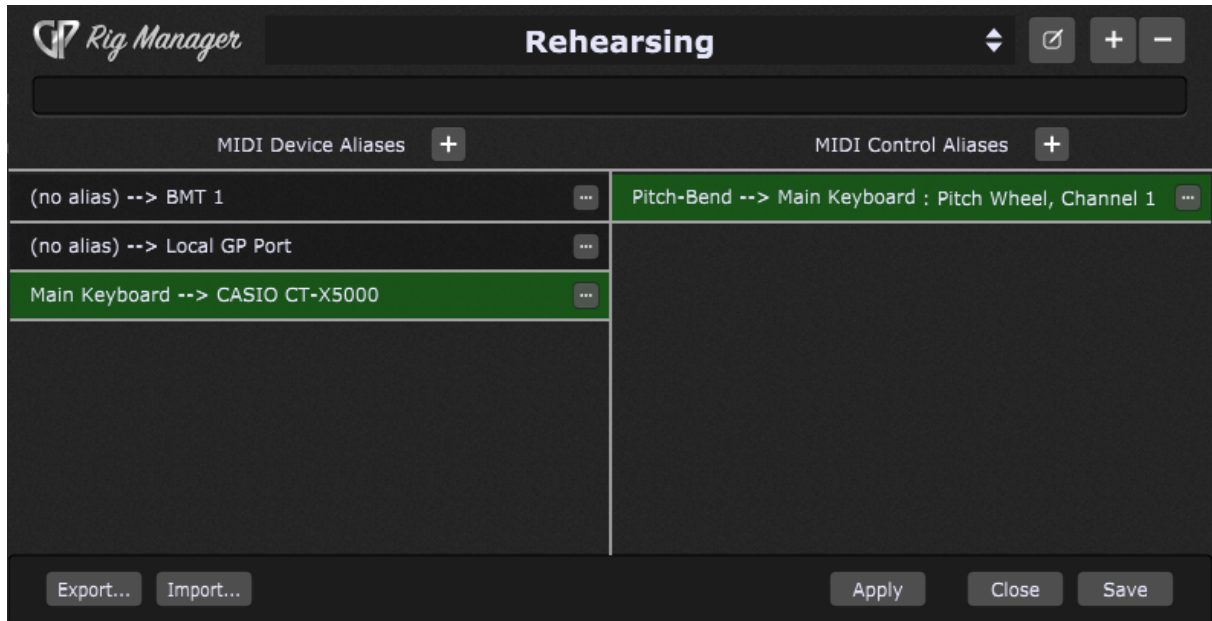


When you create a MIDI Control alias, it needs to be associated with a MIDI control:



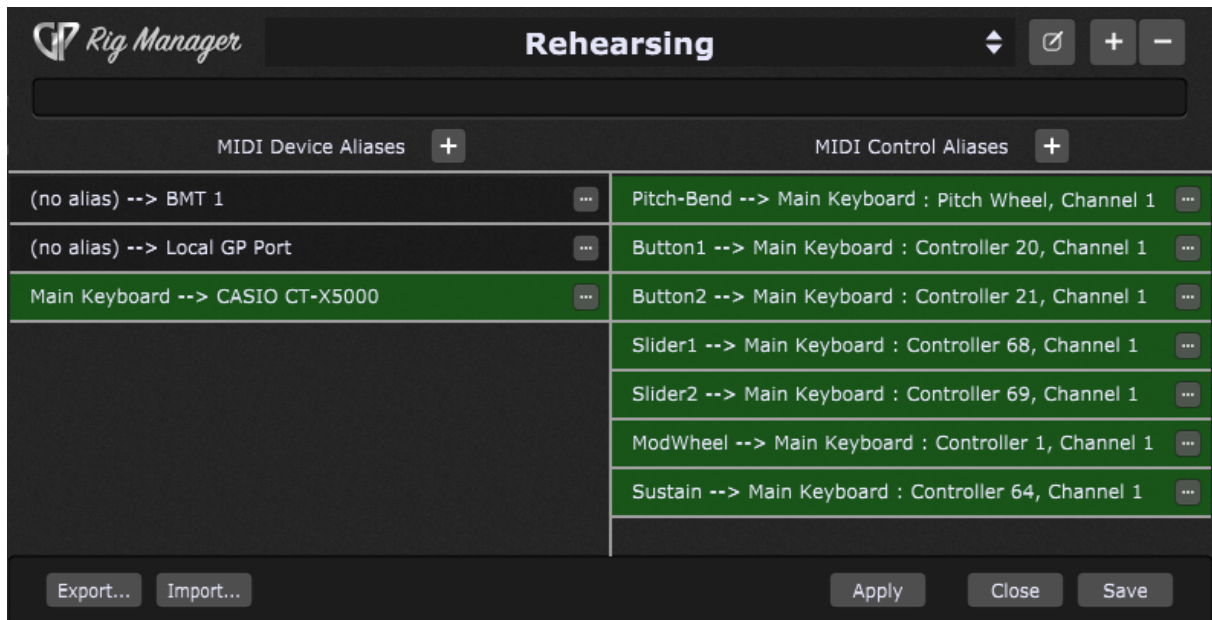
To map the newly created alias (virtual controller) to a physical control, double-click this alias to enter the Learn mode, or right-click it and choose **Learn MIDI control** from the popup menu. Move a knob, fader, pitch-bend or other physical control, and it will be mapped to the alias:





**Note:** you can create a MIDI Control alias before creating a MIDI Device alias; in that case, the full MIDI device name will be displayed in the MIDI control alias entry. The corresponding MIDI Control alias entry will be updated when you assign an alias to that particular MIDI device.

Create other MIDI Control aliases and repeat the Learn MIDI control process for other controls:



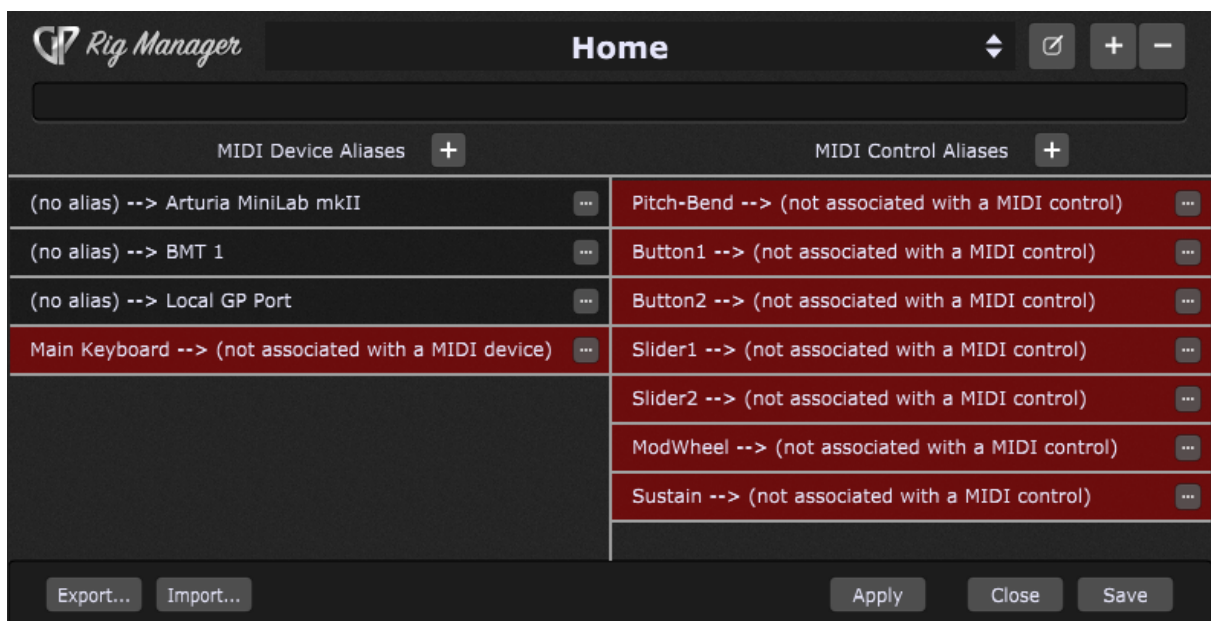
Don't forget to save your rig by clicking on the **Save** button. The best practice is also to export your Rig Manager configuration once you're done editing. Click the **Export...** button to export your configuration to a rigsetup file.

**Note:** the defined aliases in the Rig Manager don't apply to the MIDI outputs, only to MIDI inputs.

## 8.2 How to remap MIDI controls for a new MIDI device?

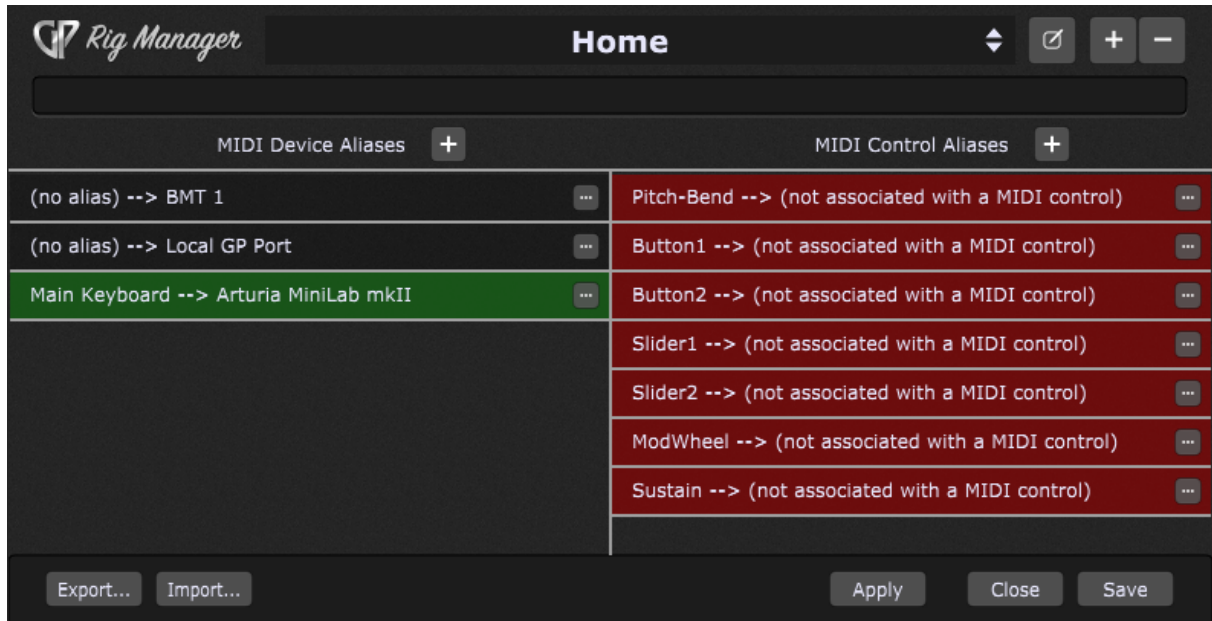
If you use different MIDI devices at home and while performing live, it's easy to associate your aliases to a different MIDI device and its controls. **Note:** this example builds on the [previous chapter](#).

Open the Rig Manager, click on the "plus" button and click on the **Add new Rig...** menu item. Give it a custom name, i.e. Home:

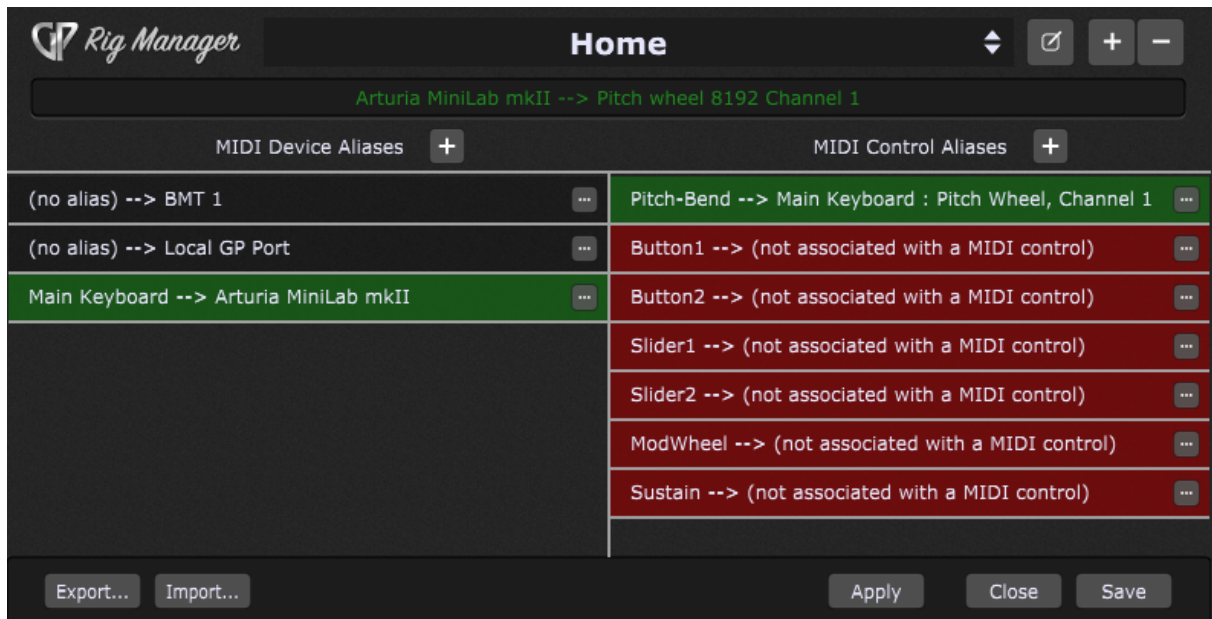


Note that previously defined aliases are preserved while creating new rigs. You only need to remap them, and you are all set.

First, right-click on the Main Keyboard alias, then expand the Associate MIDI Device sub-menu and select the desired MIDI device:

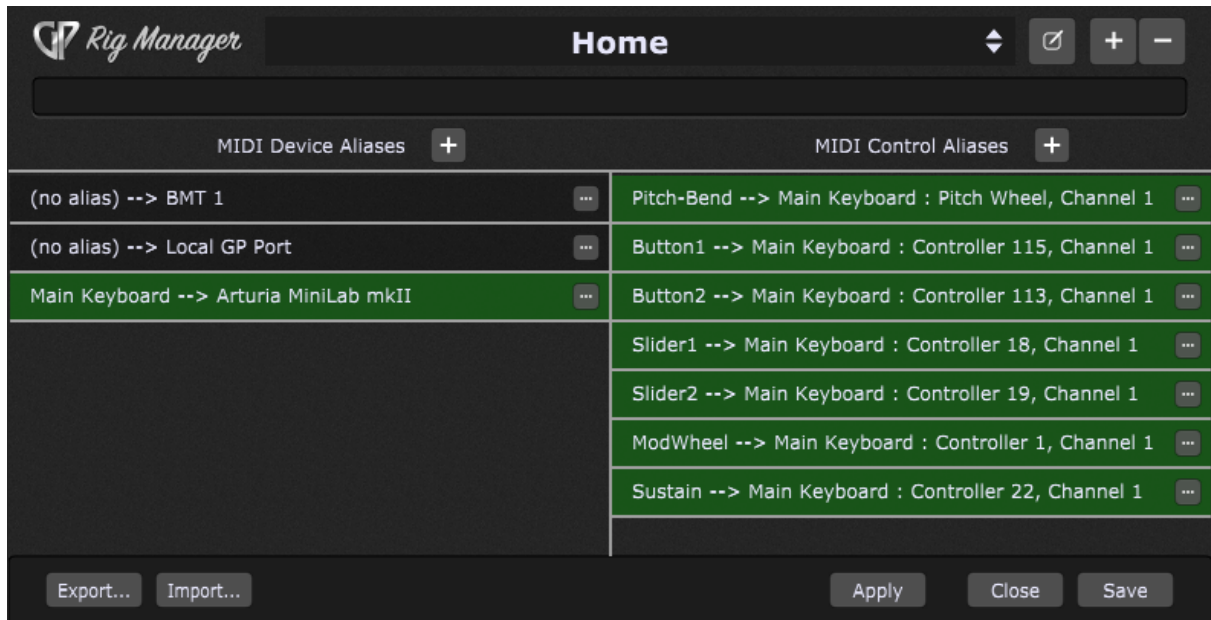


Each MIDI Control alias must be trained for a new physical controller. Double-click on the first MIDI control alias and then move the desired physical control to learn it (note: you can also enter the Learn mode for a control by clicking the "three dots" button to the right of a control and choosing Learn MIDI control):

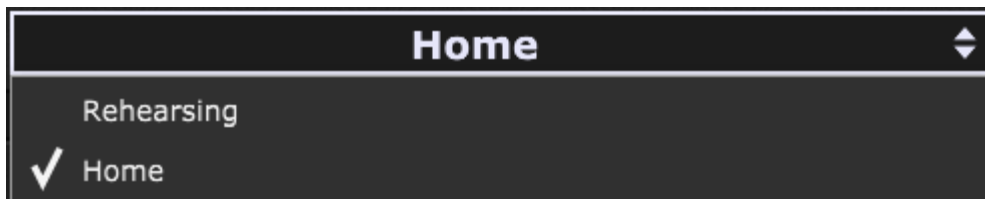


When MIDI events from the physical control are detected, the item lights up in the MIDI Device Aliases and MIDI Control Aliases lists, and the events that came in will be displayed at the top of the window.

Repeat this step for the remaining aliases in the list to associate them with the desired controls:



You can switch easily between your rigs by clicking on the rig drop-down menu:



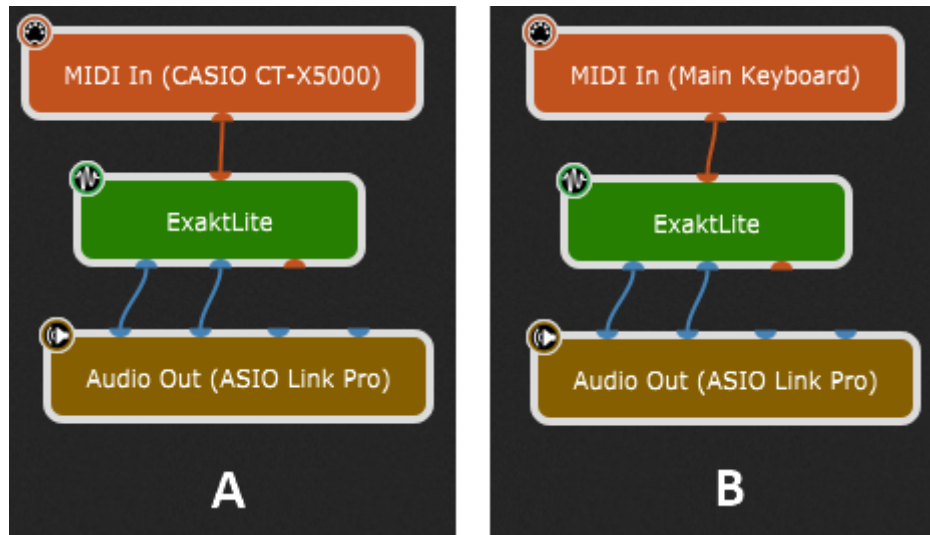
You can create new rigs for any environment you like i.e. Performing, Studio, or Local Gig and easily switch between them.

If you're sharing instruments with another performer and so have to use someone else's keyboard controller for your performance, simply remap the defined aliases as shown above, and all of your widget controls and MIDI In blocks are reestablished (refer to the [How to use aliases](#) chapter for more information). That can be accomplished in a few minutes, even during the sound-check if necessary.

Save your work when you finish with your rig configuration.

### 8.3 How to use aliases?

If you didn't create a MIDI device alias, then your MIDI device (MIDI In block) will be displayed with the full name in the Wiring View (indicated with **A**):

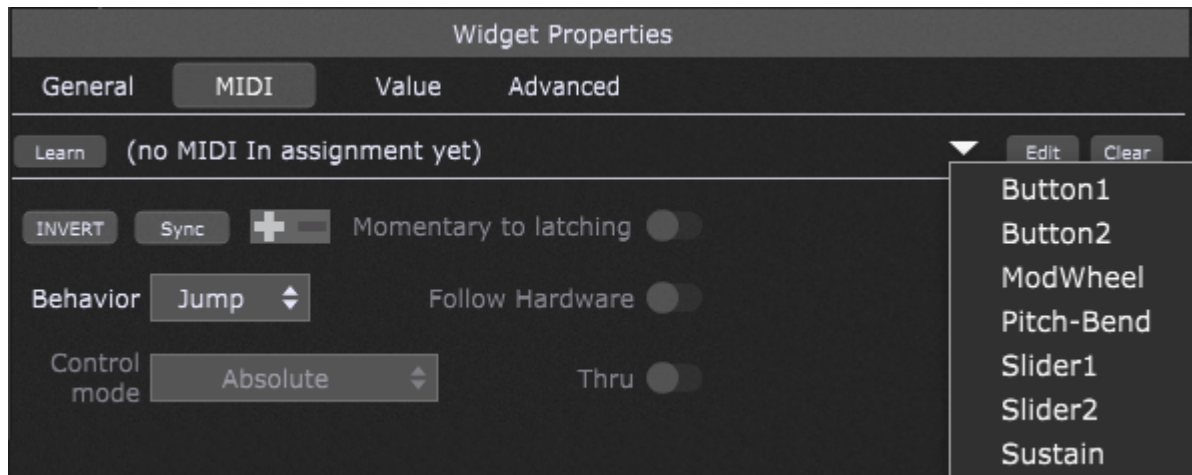


When you create an alias for this MIDI device (as shown in [this chapter](#)) in the Rig Manager, then the alias will be displayed instead (as indicated with **B**).

The benefit of using aliases for your MIDI devices (e.g. Main Keyboard, Controls Left, etc.) is that you don't need to rewire or replace your MIDI blocks to adapt to different MIDI devices, provided that you created rigs for them. Simply select the desired rig, and you're all set. The Rig Manager will automatically arrange the corresponding alias <--> MIDI device associations.

You can use the Rig Manager to help distinguish between identical MIDI controllers. For example, if you use three Roland A800 keyboard controllers, they all have the same ID (as far as USB device names are concerned), and so when you power up, you don't know which keyboard will be associated with which device in Gig Performer. So if you name them (i.e. define MIDI device aliases) "Top", "Middle" and "Bottom" (describing where the three keyboards are located relative to each other), then when you show up at a gig and power up, you can just quickly re-associate each physical controller with the appropriate aliased name.

As for the MIDI control aliases, you can easily use them with widgets in Edit mode of the Panels view; select a widget in Edit mode and click on the MIDI tab in the [Widget Properties inspector](#):



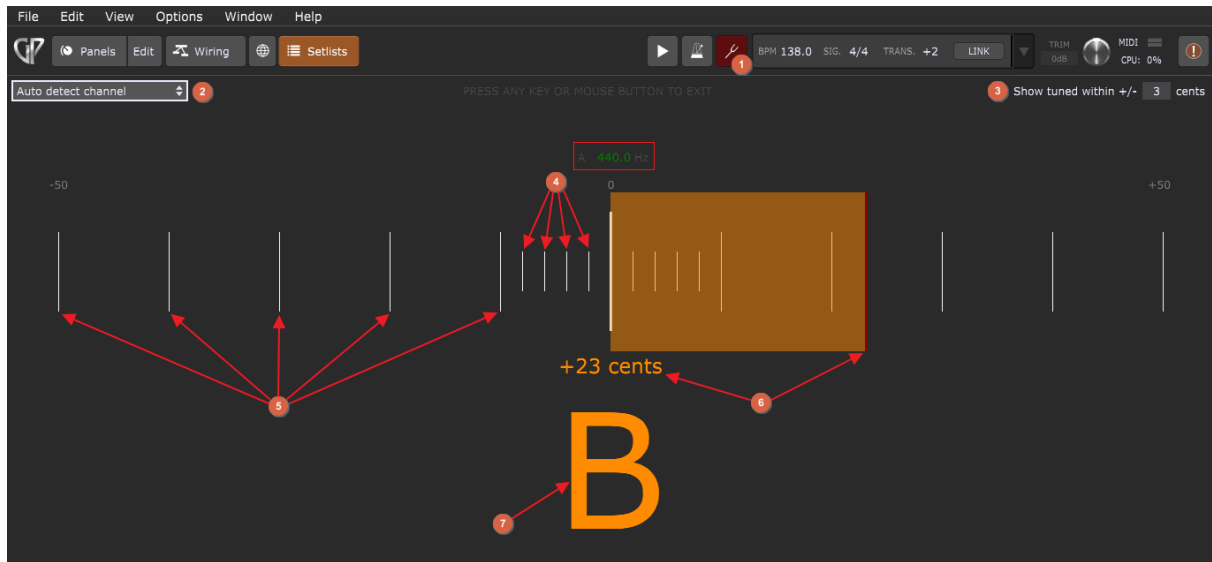
Click on the arrow to display the drop-down menu and choose one of the aliases (see the previous two chapters for more information). The benefit here is when you switch your rigs, this widget remains assigned to the corresponding MIDI control alias (e.g. Button1).

Thus, name your controllers (one per physical MIDI port) and your knobs/sliders, and then when you create new rackspace, you can just select a knob or slider from the drop-down menu in the MIDI tab (Learn area) so you don't have to keep relearning widgets, even if you're just using a single set of controllers - which is a time saver in itself.

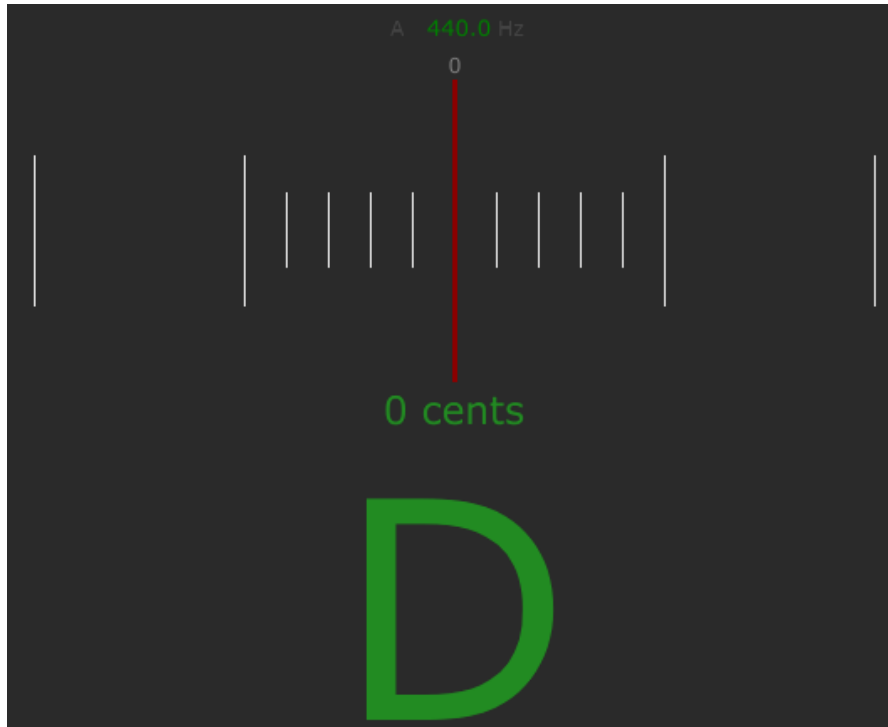
It also makes it easier for you to change your associations, even for the same rig setup.

## 9 The Tuner

Gig Performer includes a built-in tuner:



1. **Tuner button** - click this button to show/hide the tuner. Alternatively, use the keyboard shortcut **Shift + T** or map a button widget to the *Tuner* parameter of the [System Actions](#) plugin.
2. **Channels list** - click here for a drop-down menu to have the Tuner "listen" to a signal coming from a specific input channel of your connected audio interface. The default is to have it listen to signals coming from all channels and auto-detect the input channel.
3. **Show tuned within +/-** - allows you to specify a range within which the main display changes from yellow (out of tune) to green (in tune). Available values are +/- 1 cent to +/- 5 cents; default is +/- 3 cents.
4. **2 cents columns** - these smaller columns represent 2 cents (-2, -4, -6, -8 cents are pointed out with red arrows).
5. **10 cents columns** - larger columns represent 10 cents (-10, -20, -30, -40, -50 cents are pointed out with red arrows).
6. **Tuning indicator** - shows the current inaccuracy displayed in cents.
7. **Input pitch** - a large input pitch value. Yellow color indicates that your current pitch value is out of tune, and green indicates that it is in tune:



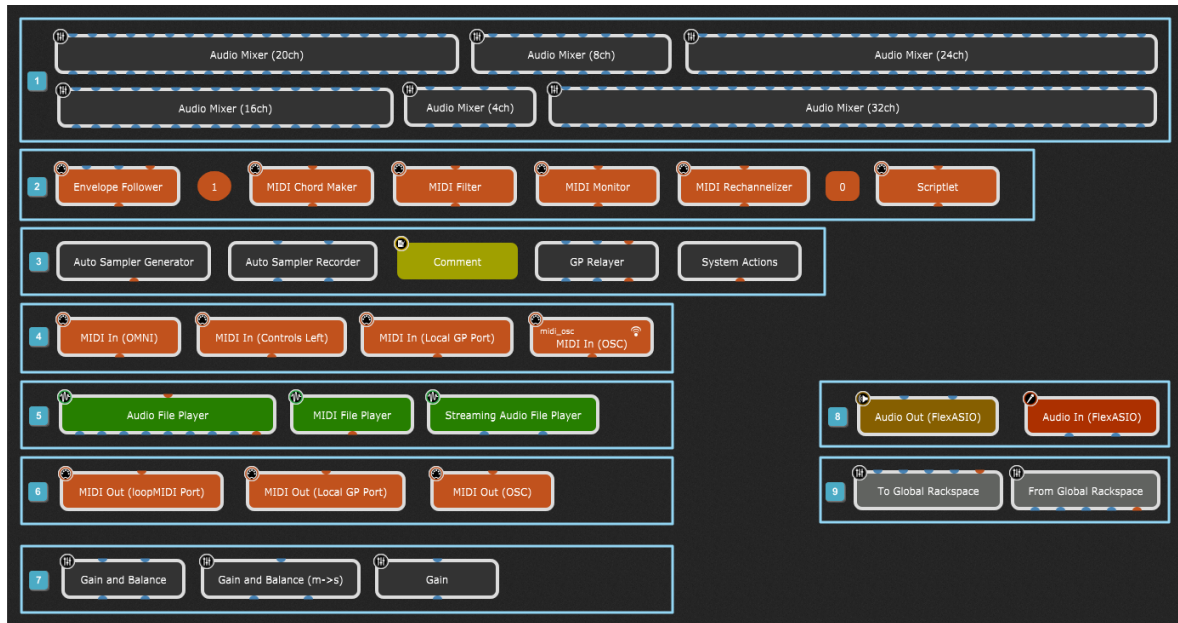
**Note:** you can change the A4 tuning frequency (indicated with the red rectangle) between 400 Hz and 480 Hz. The default value is 440 Hz.

To exit the tuner, click on the Tuner icon or click (or touch) anywhere on the tuner or press any key on the keyboard.



## 10 Built-in Plugins

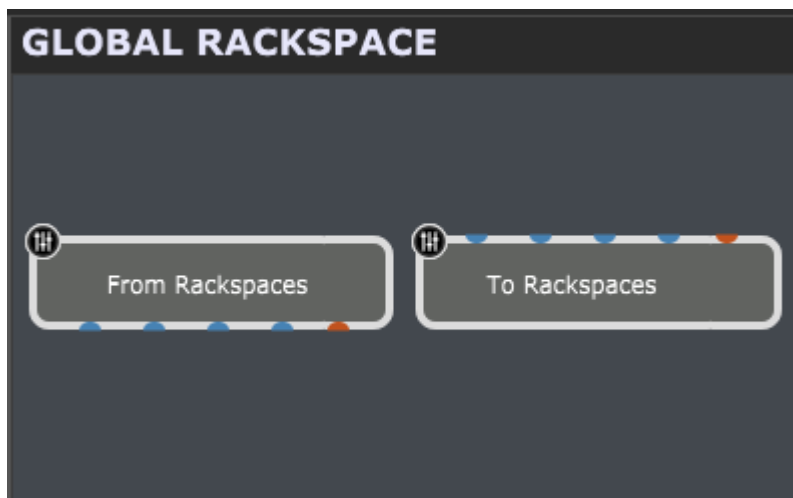
This chapter covers all the plugins that are bundled with Gig Performer. To find out more about each built-in plugin, refer to the corresponding subchapter. All built-in plugins are shown in the screenshot below:



1. **Audio Mixers:** Audio Mixer (4ch), Audio Mixer (8ch), Audio Mixer (16ch), Audio Mixer (20ch), Audio Mixer (24ch), Audio Mixer (32ch).
2. **MIDI Processing and Monitoring:** Envelope Follower, MIDI Channel Constrainer, MIDI Chord Maker, MIDI Filter, MIDI Monitor, MIDI Rechannelizer, MIDI Transposer, Scriptlet.
3. **System:** Audio Sampler Generator, Audio Sampler Recorder, Comment, GP Relay, System Actions.
4. **MIDI Inputs:** MIDI In (OMNI), MIDI In (Local GP Port), MIDI In (OSC), and any other MIDI devices that are attached to your computer will also be available as MIDI inputs.  
 Note: "MIDI In (Controls Left)" indicates that the alias "Controls Left" is created for this MIDI input device in the [Rig Manager](#). When no alias is defined for the connected MIDI input, then the full name of that MIDI device appears (e.g. "CASIO CT-X5000").
5. **Media Players:** Audio File Player, MIDI File Player, Streaming Audio File Player.

6. **MIDI Outputs:** MIDI Out (Local GP Port), MIDI Out (OSC), and any MIDI output devices present on your computer.  
Note: defined aliases in the Rig Manager don't apply to the MIDI outputs; they only apply to MIDI inputs.
7. **Gain Controls:** Gain and Balance Control, Gain and Balance Control (mono to stereo), Gain Control (mono).
8. **Audio Out** and **Audio In** - audio outputs and inputs of the connected audio interface.
9. **Global Processing:** To Global Rackspace, From Global Rackspace.

Additionally, there are two built-in plugins that are available only in the [Global rackspace](#) - **From Rackspace** and **To Rackspace**:



To add a built-in plugin onto the design surface, first, make sure that you are in the [Wiring View](#), and then either right-click to open the [contextual menu](#) or press **Ctrl + P** (or **Cmd + P** on Mac) to open the [Quick Plugin, Favorite or Preset Finder](#), and then add the desired plugin onto the Wiring view surface.

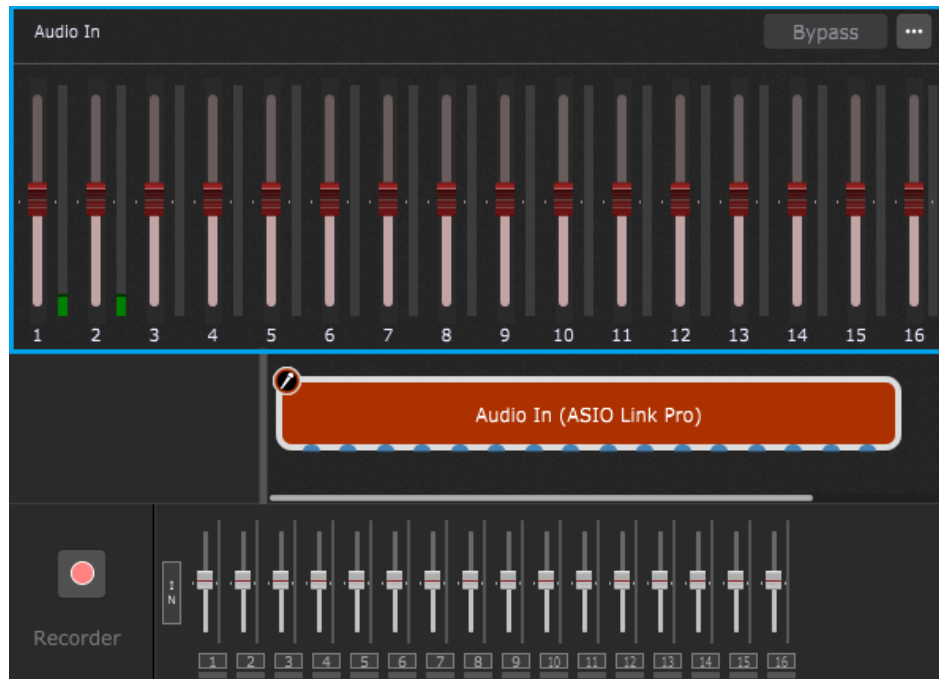
**Note:** not all built-in plugins can be added to the design surface in the Wiring view. Check the subchapters to find out more about each plugin. To learn how to insert a plugin, refer to the [Inserting and Wiring your plugins](#) chapter.

The common parameters for most plugins are **Bypass Plugin** and **Open/Close Plugin Editor**; there are plugins that cannot be bypassed (i.e. Audio In plugin). Check corresponding subchapters to find out more about parameters available for each

supplied plugin.

## 10.1 Audio In

The **Audio In** plugin represents the current connected physical interface used to receive audio or a virtual device, aggregating two or more physical interfaces (depending on your operating system):



Each plugin port represents one audio input. The actual number of these ports is a function of the capabilities of the interface connected to your system. For example, in the screenshot above, 16 ports are shown.

Double-clicking on the Audio In plugin opens its editor window, as indicated by the blue rectangle. You can adjust input levels by moving the corresponding sliders. The number of sliders matches the number of input ports. Moving any of these sliders will automatically update the corresponding slider in the bottom toolbar, and vice-versa.

However, as a matter of course, we encourage you to not adjust your levels through these global controls. It is usually better to insert [Gain Controls](#) or [Audio Mixers](#) into the [Wiring view](#). The global input and output levels are intended for emergency adjustments should you have a problem with FoH audio.

**Note:** when you create a new gig, this plugin is present by default in the Wiring view and cannot be removed from the design surface. However, you can specify different audio

interfaces for Audio In. Navigate to Options -> [Audio Setup](#) to configure your audio interfaces.

**Tip:** if you need to change the selected interface or even the buffer size or sample rate, open or create an empty gig file first and change the audio interface there. This will be much faster as Gig Performer will not have to reset and possibly reload all your plugins. Further, some plugins don't "like" having their buffer size or sample rate changed after being loaded and may consequently crash when receiving such a request.

### Audio In parameters

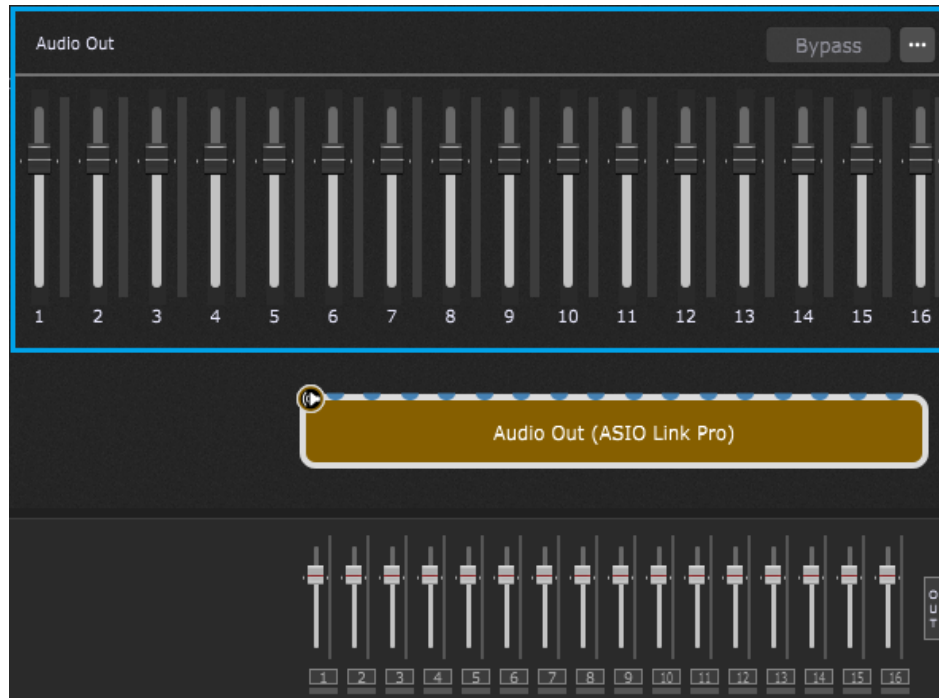
The following parameters are available for the Audio In plugin:

- **Number of channels** - allows you to get the current number of audio input channels.
- **Input Gain (Channel #)** - allows you to control each channel's input gain.
- **Input Level (Channel #)** - allows you to display the input level for each input channel available (use with meter widgets).
- **Overall Input Level** - allows you to display the overall input level for all channels (use with meter widgets).
- **Channel # Muting** - mutes the specific channel (# represents the number of the channel).

**Note:** not all parameters are suitable for the widget control. Parameters such as *Number of channels* are suited for [GScript](#) or [OSC](#) applications. Scripting can determine how many channels the Audio In plugin offers, and then the result can be used for other functions. Using OSC, you can, e.g. create the exact number of VU-meters dynamically (if your OSC application supports this feature).

## 10.2 Audio Out

The **Audio Out** plugin represents the current connected physical interface used to send audio or a virtual device, aggregating two or more physical interfaces:



Each plugin port represents one audio output. The actual number of these ports is a function of the capabilities of the interface connected to your system. For example, in the screenshot above, 16 ports are shown.

Double-clicking on the Audio Out plugin opens its editor window, as indicated by the blue rectangle. You can adjust output levels by moving the corresponding sliders. The number of sliders matches the number of output ports. Moving any of these sliders will automatically update the corresponding slider in the bottom toolbar and vice-versa.

**Note:** when you create a new gig, this plugin is present by default in the Wiring view and cannot be removed from the design surface. However, you can specify different audio interfaces for Audio Out. Navigate to Options -> [Audio Setup](#) to configure your audio interfaces.

### **Audio Out parameters**

The following parameters are available for the Audio Out plugin:

- **Number of channels** - allows you to get the current number of audio output channels.
- **Output Volume (Channel #)** - allows you to control the output volume for each output channel available.

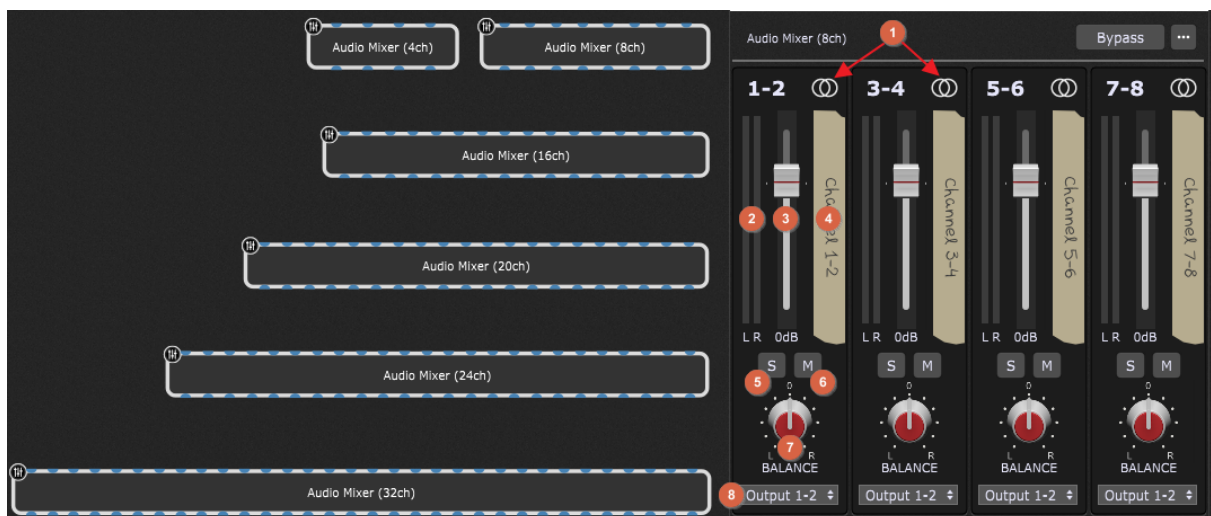
- **Output Level (Channel #)** - allows you to display the output level for each output channel available (use with meter widgets).
- **Overall Output Level** - allows you to display the overall output level for all channels (use with meter widgets).
- **Channel # Muting** - mutes the specific channel (# represents the number of the channel).

**Note:** not all parameters are suitable for the widget control. Parameters such as *Number of channels* are suited for [GScript](#) or [OSC](#) applications. Scripting can determine how many channels the Audio Out plugin offers and then the result can be used for other functions. Using OSC, you can, e.g. create the exact number of VU-meters dynamically (if your OSC application supports this feature).

### 10.3 Audio Mixers

The **Audio Mixer/Router** is a flexible plugin that eliminates the need for a Gain plugin on the output of each source. Functions like level, mute, and output routing can be controlled from widgets so you can, for example, mute and unmute different sources, route channels to various effects devices connected to the outputs, or, of course, mix levels. Each input hosts a stereo channel pair, or can be switched to mono operation.

There are six versions of the plugin, with 2 inputs (4 channels), 4 inputs (8 channels), 8 inputs (16 channels), 10 inputs (20 channels), 12 inputs (24 channels), and 16 inputs (32 channels):



The screenshot above shows different versions of the Audio Mixer plugin and the plugin editor of the Audio Mixer (8ch) plugin; it has 4 stereo inputs and 8 channels. Elements in the plugin window are:

1. **Mono/Stereo channel switch** - click this icon to toggle the channel strip between mono (left channel) and stereo input.
2. **Channel level meter** - displays the post-fader output level from the channel.
3. **Channel level fader** - adjusts level from the channel to the selected output pair. Double-click the fader to return it to the 0 dB position (alternatively, press **Alt** + click on Windows or **Option** + click on macOS). Press **Ctrl** (or **Cmd**) and drag the fader to adjust the level in 0.1 dB steps.
4. **Channel label** - to edit a channel name, double-click the "tape strip".
5. **Channel Solo button** - solos the channel, effectively muting all others.
6. **Channel Mute button** - mutes the channel.
7. **Channel Balance/Pan knob** - for stereo channels, this is a left/right balance control, while on mono channels it is a pan control. Double-click on this control to return it to center (alternatively, press **Alt** + click on Windows or **Option** + click on Mac).
8. **Output channel select** - click this field to choose an output pair for a strip from the drop-down menu.

**Tip:** if you connect multiple plugins' audio outputs to another plugin's input, the signals will be mixed automatically. To create custom blends, insert one of the Audio Mixer/Router plugins and balance as desired. Audio Mixer/Router plugins can also be used to create reverb sends, where the signal is routed to a reverb (or other type of signal processing plugin) from multiple sources. Simply insert an Audio Mixer/Router plugin and route the output of each source to its inputs and a pair of outputs to the reverb plugin.

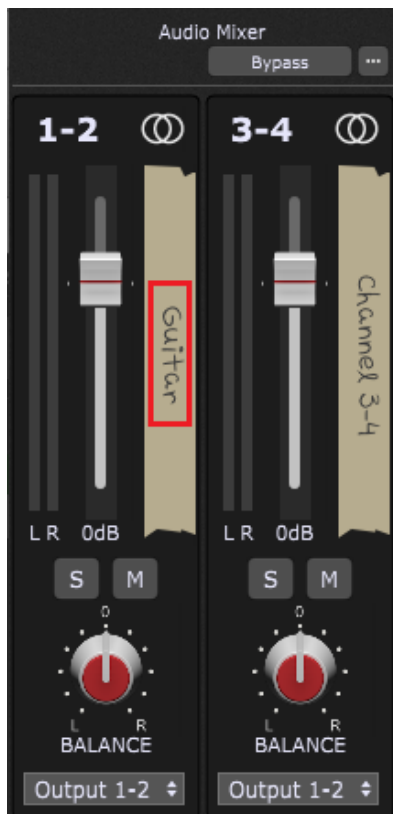
### Parameters

The following parameters are available for Audio Mixers:

- **Channel # volume** - adjusts the channel volume (indicated with **(3)** on the screenshot above).
- **Channel # balance** - adjusts the channel balance (indicated with **(7)** on the screenshot above).

- **Channel # solo** - toggles Soloed/Not soloed modes (indicated with **(5)** on the screenshot above).
- **Channel # mute** - toggles Muted/Not muted modes (indicated with **(6)** on the screenshot above).
- **Channel # output** - changes the output channel (indicated with **(8)** on the screenshot above).
- **Channel # left meter** - displays the left meter (indicated with **(2)** on the screenshot above).
- **Channel # right meter** - displays the right meter (indicated with **(2)** on the screenshot above).

**Note:** names of these parameters may differ depending on the channel label (4); for example, if you rename the channel label from "Channel 1-2" (default) to "Guitar" (indicated with the red rectangle), then corresponding parameters will be named like shown in the screenshot below:

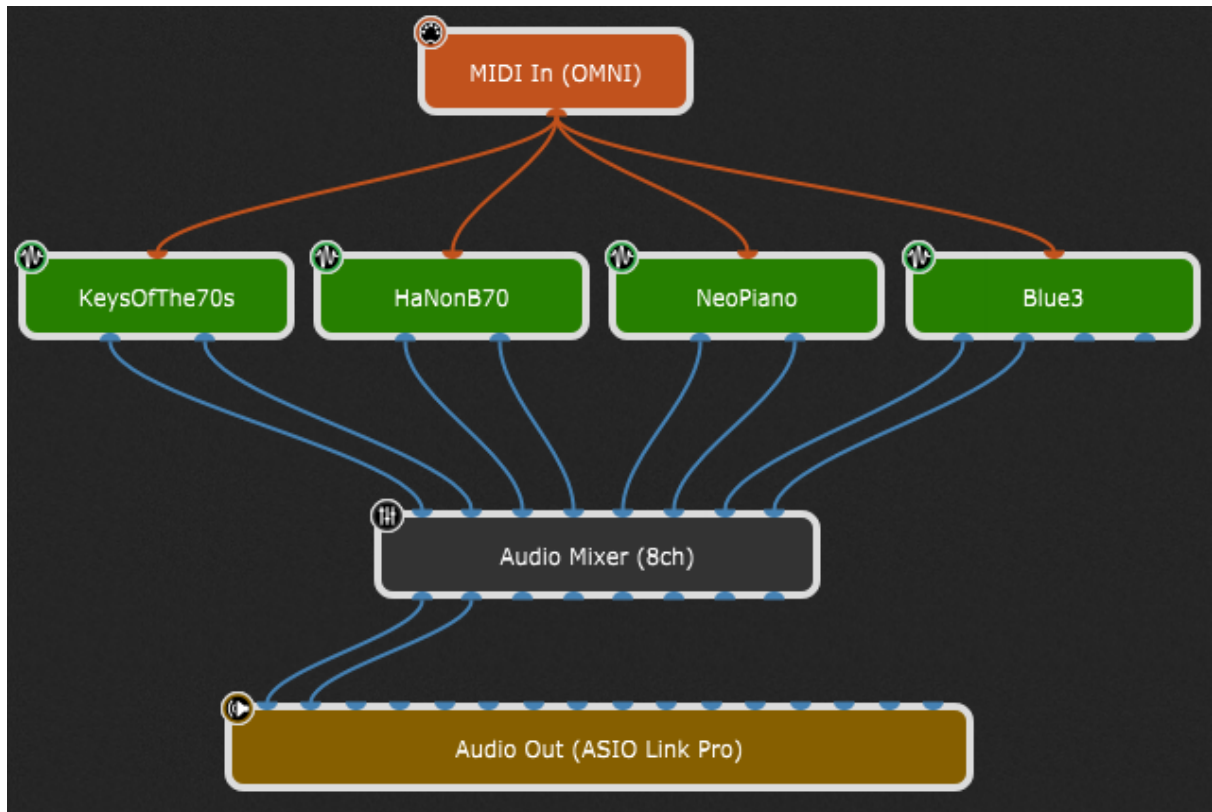




### 10.3.1 Audio Mixer applications

Many tasks can be accomplished with the Audio Mixer plugin. Here are a few examples to get you going.

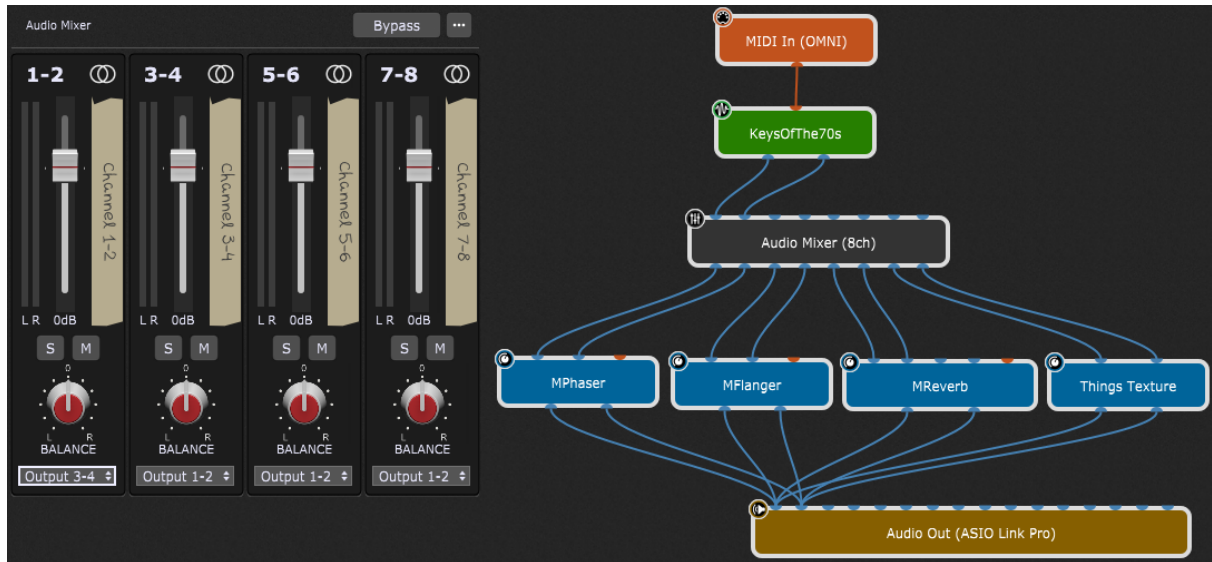
#### Simple Mixer



This example has four synths with stereo pair outputs that feed the audio mixer's eight inputs. Each input has gain and balance controls in the Audio Mixer plugin (so it is unnecessary to add gain and balance plugins on each synth output). All the inputs feeding the audio mixer are then routed to outputs 1 and 2 of the mixer plugin, which is connected to the first stereo pair of audio interface outputs.

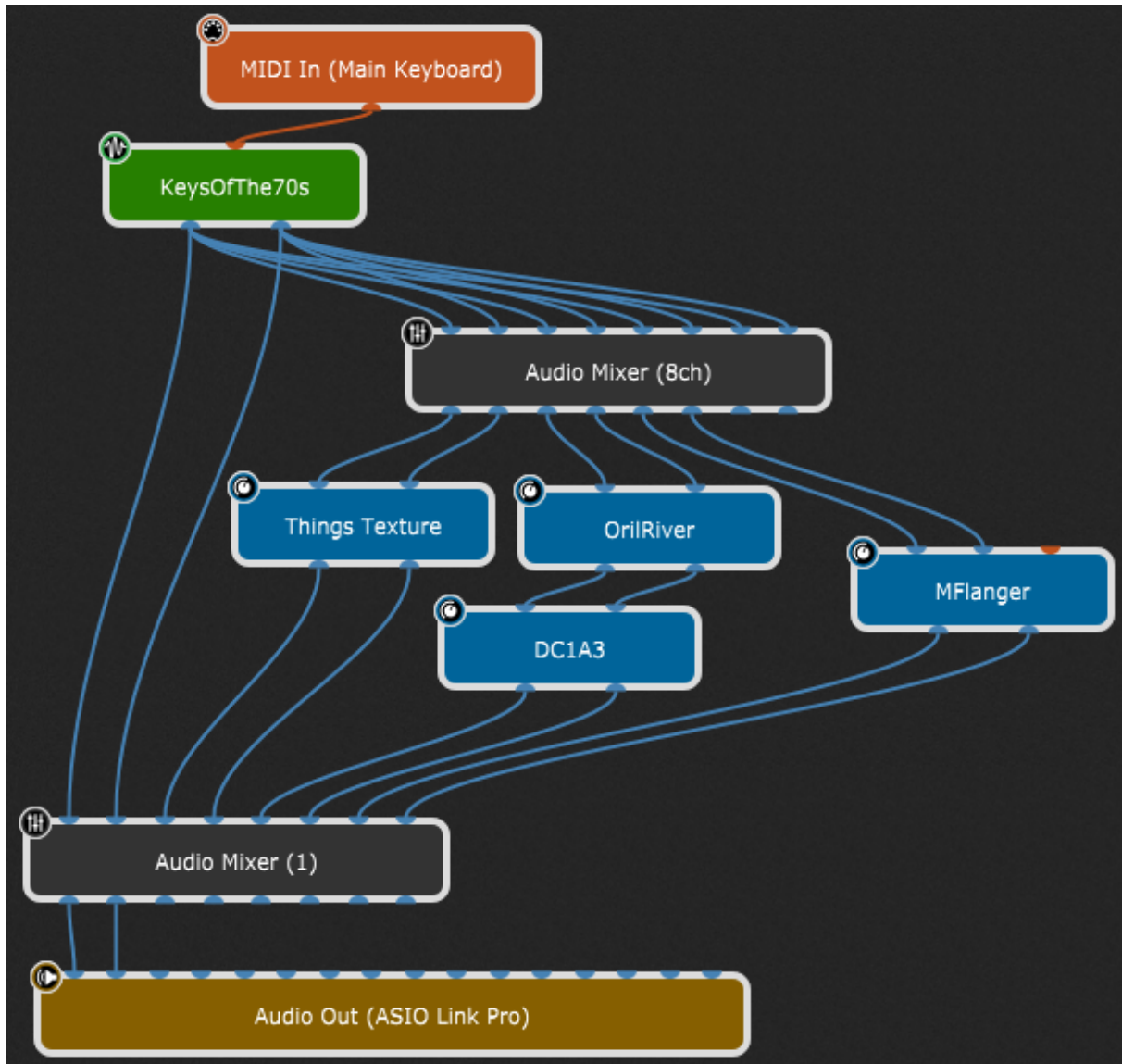
Each channel pair can be soloed or muted as desired, and of course, these can be controlled from front panel widgets (and so from your keyboard controller once you map them to those widgets) as needed.

#### Input router



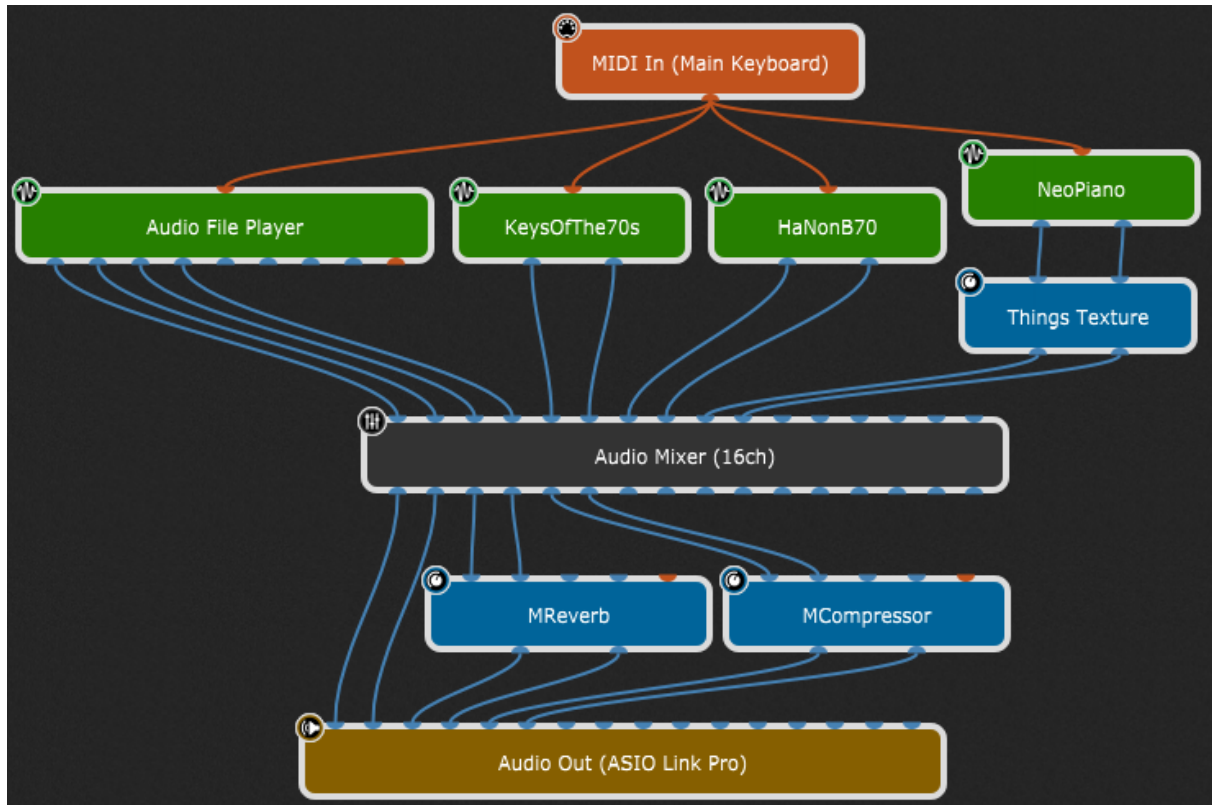
On the right, in the Wiring view of a rackspace, you can see that we have a single synth (the KeysOfThe70s) connected to the Audio Mixer (8ch). The four output pairs of the Audio Mixer are connected to the inputs of four different effects. If you look at the first channel strip in the Audio Mixer editor window (on the left), the KeysOfThe70s is currently being routed to Output 3-4, which feeds the Flanger plugin. By changing the output routing, you can quickly switch from one effect to another. Output routing also can be attached to a slider, knob, or switch widget, letting you control effects switching from your controller.

### Feed multiple effects (aux sends)



Assigning the source synth plugin to all of the inputs turns the mixer into a bank of aux sends. Each channel sends the source signal to one pair of outputs, with a separate level control for each. In this case, the outputs of the effects and the original synth source are all mixed at the end through another eight-channel mixer.

**Multiple submixers in one plugin**

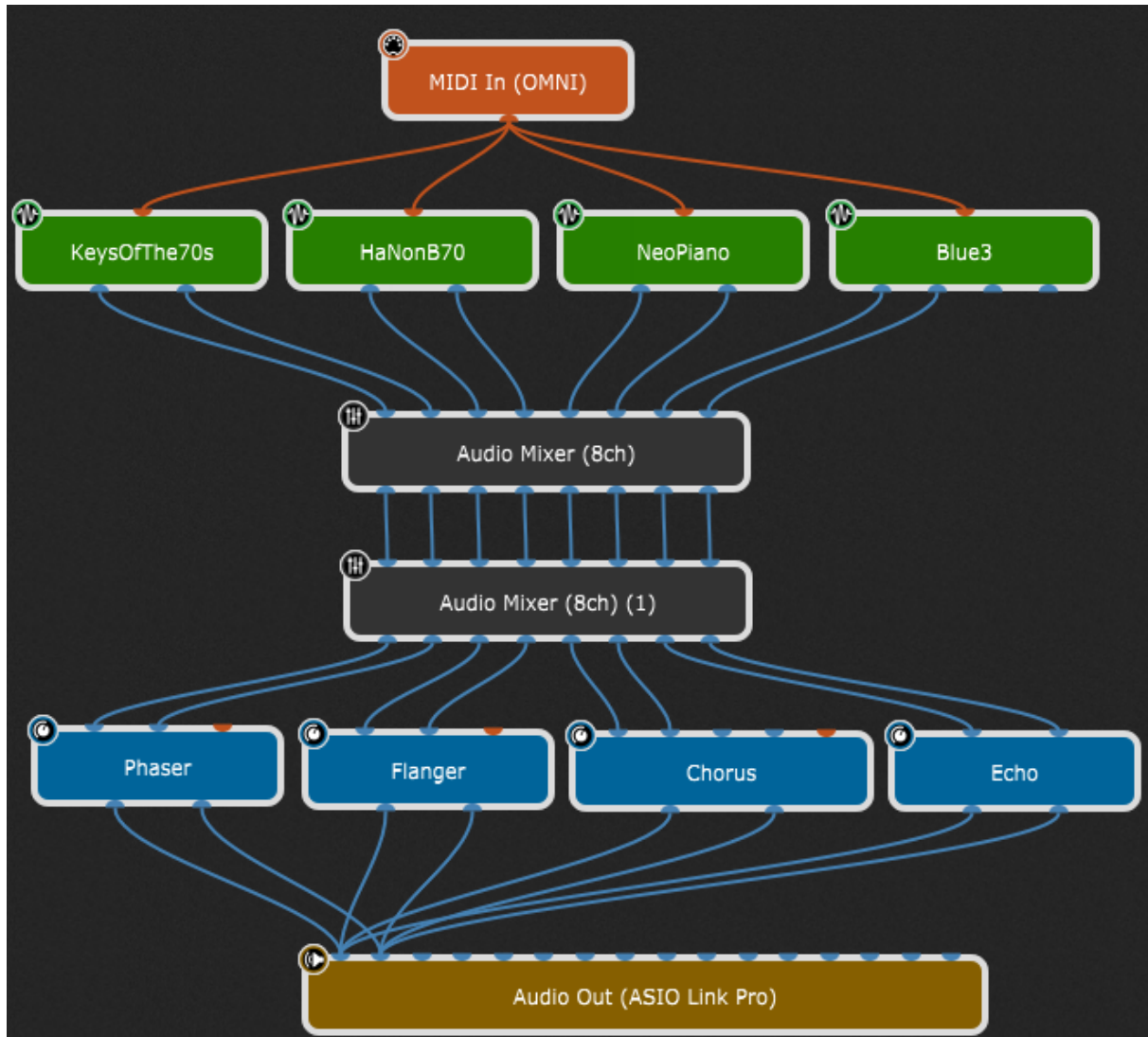


Since inputs can be independently assigned to outputs, a single 16-channel mixer can be used as if it were a few small submixers in one package.

The first and third channel strips (channels 1-2 and 5-6) feed one mix; the second and fourth strips (channels 3-4 and 7-8) feed another mix. The fifth strip (channels 9-10) is routed to the third output pair that feeds an effect:



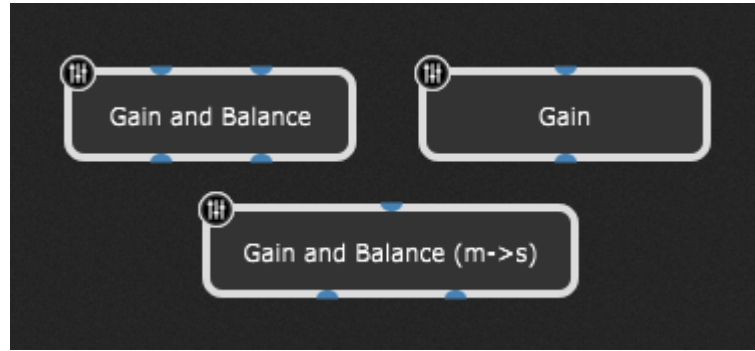
## Mixer and Router Together



In this example, we can instantly route combinations of inputs to combinations of outputs so you can route KeysOfThe70s to the phaser and chorus while routing Blue3 to chorus and echo. By using rackspace variations in Gig Performer, you can instantly switch among arbitrary combinations, using a controller to switch, if you like.

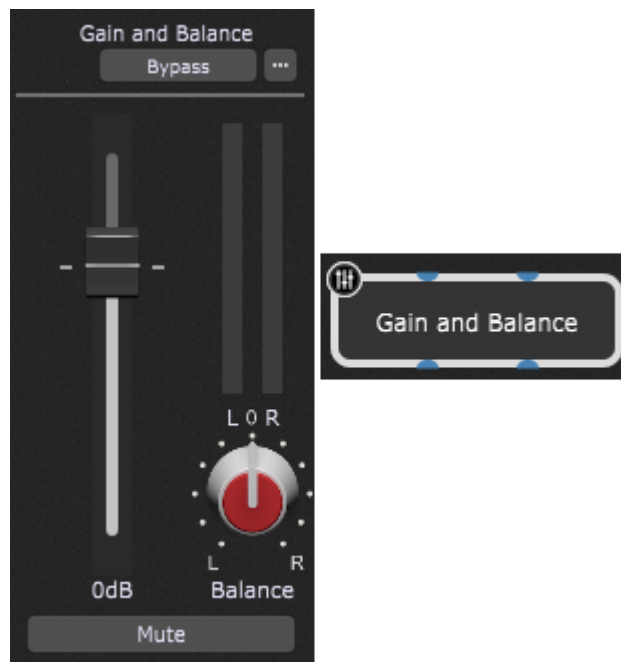
## 10.4 Gain Controls

The gain controls plugins include a Gain and Balance Control, a Gain Control (mono) and a Gain and Balance Control (mono to stereo).



### 10.4.1 Gain and Balance Control

This plugin provides stereo audio input pins and stereo audio output pins:



It allows the gain and Left/Right balance of a stereo audio signal to be adjusted. Clicking on the **Mute** button or setting the **Volume** slider to 0 causes the plugin to be muted (not bypassed). Click [here](#) for more information on the difference between muting and bypassing a plugin.

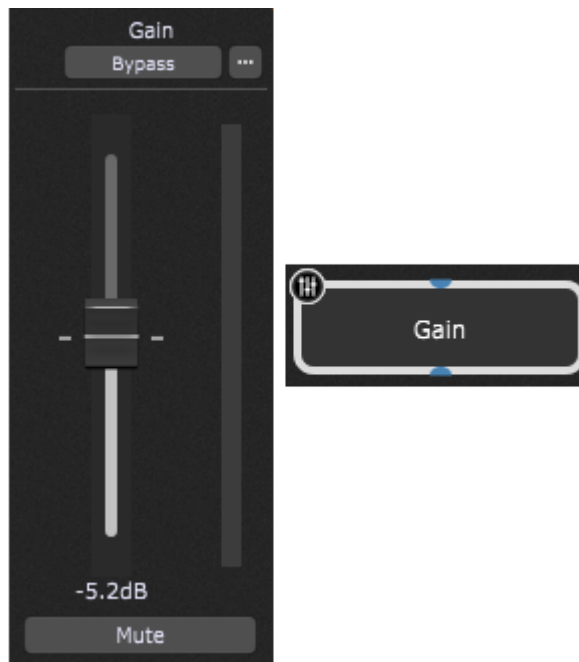
#### Gain and Balance Control parameters

Besides the common parameters **Bypass Plugin** and **Open/Close Plugin Editor**, the following parameters are provided for the Gain and Balance Control plugin:

- **Gain** - allows you to control the gain.
- **Balance** - allows you to control balance (left-right).
- **Channel # Magnitude** - magnitude parameters are used for level metering. Metering can be very helpful in verifying proper gain structure and ensuring that a signal is not overloaded and distorted.
- **Muted** - mutes or unmutes this plugin block.

#### 10.4.2 Gain Control (mono)

This plugin provides a single audio input pin and a single audio output pin:



It allows the gain of a mono audio signal to be adjusted. Clicking on the **Mute** button or setting the **Volume** slider to 0 causes the plugin to be muted (not bypassed). Click [here](#) for more information about the difference between muting and bypassing a plugin.

#### Gain Control parameters

Besides the common parameters **Bypass Plugin** and **Open/Close Plugin Editor**, the following parameters for the Gain Control plugin are available:

- **Gain** - allows you to control the gain.

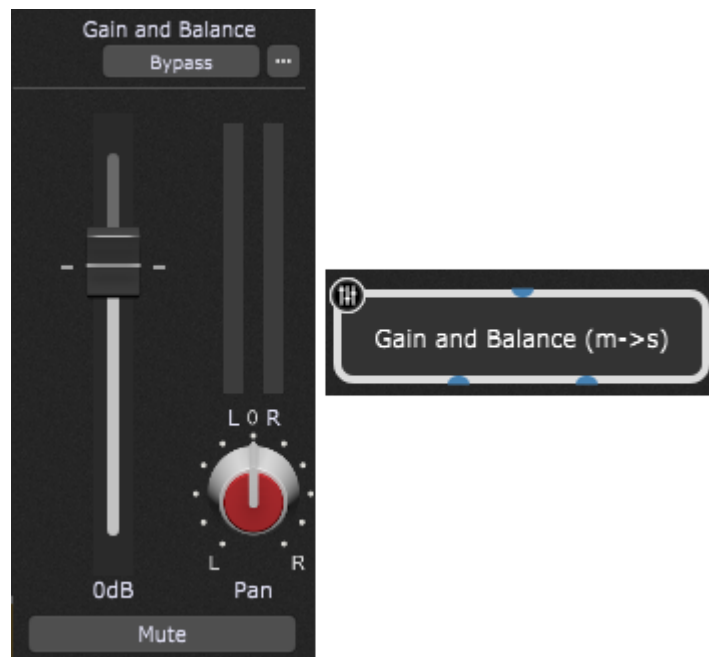


- **Muted** - mutes or unmutes this plugin block.
- **Magnitude** - the magnitude parameter is used for level metering. Metering can be very helpful in verifying proper gain structure and ensure that a signal is not overloaded and distorted.

The meter widgets provided by Gig Performer (available in both vertical and horizontal configurations) allow you to visually monitor levels at any point in the signal chain.

### 10.4.3 Gain and Balance Control (mono to stereo)

This plugin provides a single audio input pin and stereo audio output pins:



It converts a mono audio signal to stereo and allows its gain to be adjusted. Clicking on the **Mute** button or setting the **Volume** slider to 0 causes the plugin to be muted (not bypassed). Click [here](#) for more information about the difference between muting and bypassing a plugin. A balance control enables the incoming mono source to be panned across Left/Right channels.

#### Gain and Balance (m->s) parameters

Besides the common parameters **Bypass Plugin** and **Open/Close Plugin Editor**, the following parameters for the Gain and Balance (m->s) plugin are available:

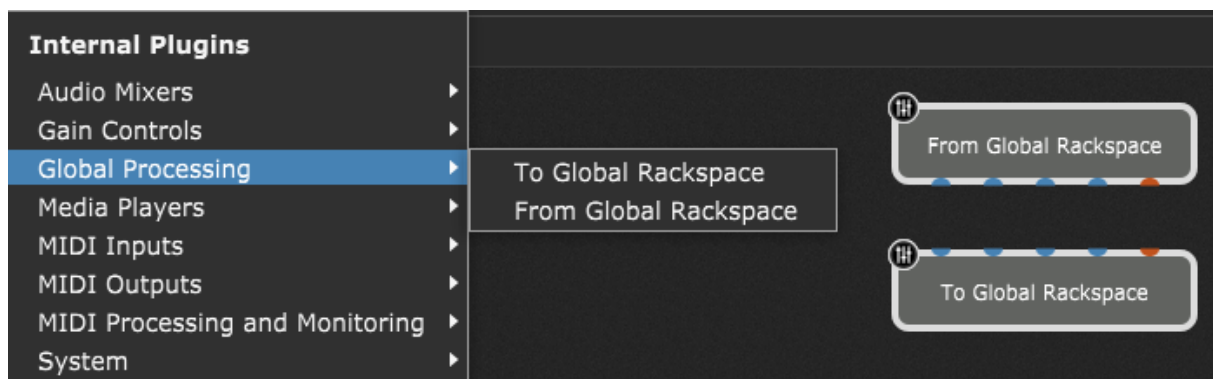
- **Gain** - allows you to control the gain.

- **Balance** - allows you to control balance (left-right).
- **Channel # Magnitude** - magnitude parameters are used for level metering. Metering can be very helpful in verifying proper gain structure and ensuring that a signal is not being overloaded and distorted.
- **Muted** - mutes or unmutes this plugin block.

The meter widgets provided by Gig Performer (available in both vertical and horizontal configurations) allow you to visually monitor levels at any point in the signal chain.

## 10.5 Global Processing

Global Processing plugins encompass **To Global Rackspace** and **From Global Rackspace** plugins:

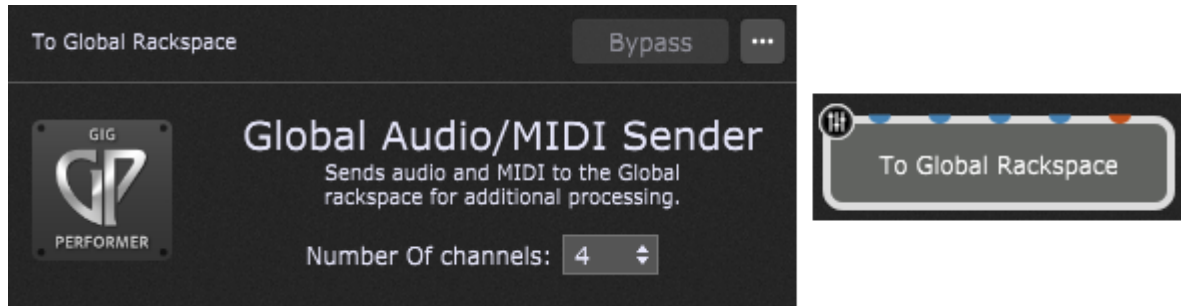


**To Global Rackspace** is used to send audio and MIDI from regular rackspaces to the Global rackspace. **From Global Rackspace** is used to receive audio and MIDI from the Global Rackspace to rackspaces.

Refer to the [Global Rackspace](#) chapter for more information.

### 10.5.1 To Global Rackspace

**To Global Rackspace** is used to send audio and MIDI from regular rackspaces to the Global rackspace for additional processing. By default, this plugin provides four audio input pins and one MIDI input pin:



Double-clicking this plugin opens its editor window, where the number of channels can be selected. You can select up to 64 channels for this plugin.

**Important note:** changing the number of channels will update all the global plugin blocks in all rackspaces. Affected plugins are:

- To Global Rackspace
- From Global Rackspace
- From Rackspaces
- To Rackspaces

From Rackspaces and To Rackspaces plugins are present in the Global rackspace only, and will be addressed separately since they are not the part of the Global Processing plugins' group.

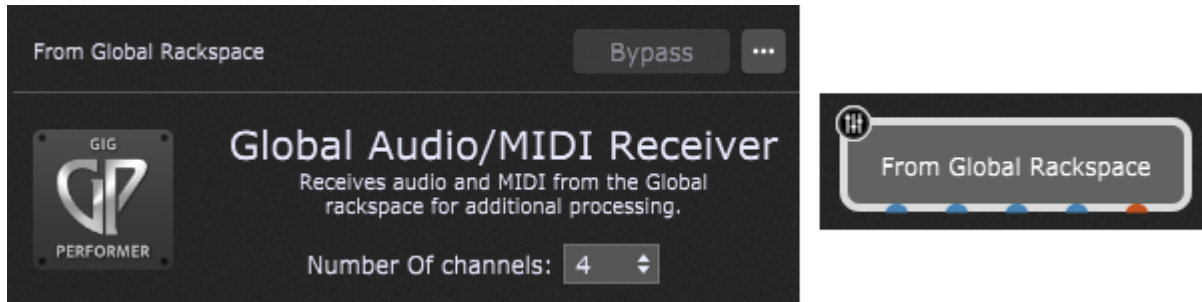
See more information on how to connect **To Global Rackspace** plugin in the [Global Rackspace](#) chapter.

### **Parameters**

The To Global Rackspace plugin provides 256 available and initially empty slots; to occupy a specific slot, you have to set the Global Parameter Assignment in the [Widget Properties inspector](#) for a global widget. Refer to [this chapter](#) to learn more.

## **10.5.2 From Global Rackspace**

**From Global Rackspace** is used to receive audio and MIDI from the Global rackspace to regular rackspaces for additional processing. By default, this plugin provides four audio output pins and one MIDI output pin:



Double-clicking this plugin opens its editor window, where the number of channels can be selected. You can select up to 64 channels for this plugin.

**Important note:** changing the number of channels will update all the global plugin blocks in all rackspaces. Affected plugins are:

- To Global Rackspace
- From Global Rackspace
- From Rackspaces
- To Rackspaces

**From Rackspaces** and **To Rackspaces** plugins are present in the Global rackspace only, and will be addressed separately since they are not the part of the Global Processing plugins' group.

### Parameters

The From Global Rackspace plugin provides 256 available and initially empty slots; to occupy a specific slot, you have to set the Global Parameter Assignment in the [Widget Properties inspector](#) for a global widget. Refer to [this chapter](#) to learn more.

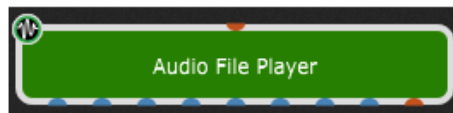
## 10.6 Media Players

In this chapter find out more about the integrated media player plugins in Gig Performer - Audio File Player, MIDI File Player, and Streaming Audio File Player.

**Tip:** Read [this blog article](#) to learn about the Gig Performer features that make it a powerful tool for backing track users.

## 10.6.1 Audio File Player

The **Audio File Player** plugin provides a MIDI input pin, eight audio output pins, and a MIDI output pin:



It allows you to load and play up to eight audio files (in AIF, FLAC, MP3, OGG or WAV formats), either one at a time or simultaneously. You can use this plugin as a backing track player or for looping short samples, switching seamlessly from one to another while running. Up to eight audio output channels are available, singly or in pairs.

The Audio File Player plugin provides eight "lanes" for playback. You can think of these as roughly equivalent to tracks on a DAW although each lane could have an entire song on it. Its editor provides controls that enable the following operations:

1. **Single/Multi** - allows you to switch between playing back one audio file at a time versus multiple files simultaneously.
2. **Loop** - turns looping on or off for all lanes.
3. **Go To Beginning** - allows you to instantly go to the beginning of all audio files.
4. **Time/Tracks display** - displays the current time, in hours/minutes/seconds/hundredths of a second, as well as the number of tracks (channels) loaded with audio files.
5. **Play/Pause** - allows you to start or pause file playback. The "pause" function does not return the file to the beginning - it simply pauses it wherever it is. Clicking on the button a second time resumes playback from the point at which it was stopped.
6. **Sync** - when turned on (lit green), Gig Performer will play and pause the Audio File Player automatically when its button is engaged. This allows you to begin playback of audio files at the same time you start a drum machine emulation plugin, for example.
7. **Load Full Directory** - loads a full folder of audio files at once, up to the number of available (empty) lanes. If, for example, you have six empty lanes and you try loading eight files using this button, only the first six files will get loaded, into lanes 3 - 8. With this feature, you can create a directory with all of the files you will need, then load the entire player with them at the gig by invoking only a single command.
8. **Metronome** - clicking on this icon provides a number of additional controls that allow you to create a MIDI click track in sync with the tempo of your Master track. See the [MIDI metronome](#) chapter for more details.

In addition, the Audio File Player editor offers the following lane-specific controls:

9. **File name** - displays the name of the file currently loaded in the lane.
10. **Play/Silence this lane** - enables or mutes playback of the file loaded into the lane.
11. **Load New Audio File** (file cabinet icon) - click this to load an audio file into the lane.
12. **Remove Audio File** (trashcan icon) - click this to delete the audio file from the lane.
13. **Master indicator** - when lit, indicates that the file in the lane is being used as the Master, determining the actual playback length of all other files. Note that this is not a parameter that can be changed; instead, Gig Performer automatically designates the lane loaded with the longest audio file as the Master. If you load

other lanes with new files, the Master designation may change accordingly (if the new files are longer than the one in the currently designated Master lane).

14. **Output assignment** - allows you to set the output for that lane's playback. A stereo file should ideally be set to a pair of outputs (i.e. Output 1&2, Output 3&4, etc.), but if it is sent to a mono output, the two channels will automatically be mixed to mono. Conversely, sending a mono file to a pair of outputs will result in the same signal being routed to both outputs.
15. **Level indicators** - shows the current playback level for that lane.
16. **Volume control** - allows you to adjust the volume of the lane.
17. **Oneshot on/off** - allows you to turn looping off for that lane, even if the master Loop button is on (lit). Clicking on the Oneshot button (causing it to light) during playback will cause the file in that lane to play to the end and then stop. If you then turn Oneshot for that lane off again, the file will resume playback when the file loaded into the Master lane ends.
18. **File duration** - displays the duration in seconds and samples of the file currently loaded in the lane.

#### To load audio files, do any of these:

- Click the Load New Audio File button in a lane and navigate to the desired file.
- Drag and drop the file from the macOS Finder or Windows Explorer onto the lane into which you want it loaded.
- Click the Load full directory button and navigate to a source folder of audio files to be loaded in the Audio File Player.

#### To play audio files:

- a) to play a single audio file, be sure the Single/Multi button is set to single and that the lane is play-enabled, then click the Play/Pause button to start playback.
- b) To play multiple files simultaneously:
  - i. Load all of the files you want to play.

- ii. Set the Single/Multi button to Multi. This allows you to enable or mute multiple lanes.
- iii. Set muting for tracks you don't want to hear.
- iv. Click the Play/Pause button to start playback.

#### **Things to note about the Audio File Player:**

- The Audio File Player designates the longest file loaded as the Master, whether or not it is play-enabled. The duration of the Master track becomes the loop length when loop playback is enabled.
- The [MIDI metronome](#) is generated based on the length of the Master track and information **you** supply about the number of bars in the loop and the number of beats in each bar.
- Lanes can be muted and enabled in real-time while the player is in playback. As this is muting, not stopping playback, all files always stay in sync with the master track's location, so if a lane is muted during playback and then enabled 10 seconds later, it resumes playing 10 seconds later than the point at which it was muted.
- When a file is muted, you will still be able to see "ghost" meter activity for it.
- Beneath the file duration on the right of each lane is a progress bar that turns blue as the file plays. The length of the bar represents the full duration of the file in that lane and the progress through it of the play cursor. So, while working in Multi mode, in 10 seconds, the blue bar in a lane with a 20 second-long file is filled to one-half, where as the blue bar for a two-minute loop running at the same time in another lane is only filled 1/12th of the way across, just a small chunk.
- To move the playhead to different locations in the Master file other than the start, create a knob or slider widget and map it to the Playhead position parameter of the Audio File Player.
- The Audio File Player allows files of different sample rates to be loaded into separate lanes and played simultaneously.

#### **Audio File Player parameters**



The following parameters are available for the Audio File Player plugin:

- **Play/Pause** - allows you to start or pause playback.
- **Play From Beginning** - causes playback to start from the beginning in response to a widget movement.
- **Play From Marker** - causes playback to start from the marker position (see the "Marker position" below) in response to a widget movement.
- **LOOP On/Off** - allows you to turn looping on/off.
- **Playhead position** - allows you to change the playback start point from a widget. Using a continuous widget (such as a knob or a slider) for this purpose enables you to seek particular points or even to "scrub" one or more tracks while they are playing.
- **Marker Position** - allows you to change the marker position from a widget.
- **Set Marker now** - allows you to set the marker position "on the fly" from a widget.
- **Click Generation On/Off** - allows you to use a widget to turn the MIDI click track on or off.
- **Play Mode (single/multi)** - allows you to use a widget to switch between Single and Multi playback.
- **Lane 1-8 Playing** - allows you to use a widget to enable playback from the specified lane (1-8).
- **Lane 1-8 One Shot On/Off** - allows you to use a widget to turn the Oneshot mode on or off for the specified lane.
- **Lane 1-8 Volume** - allows you to use a widget to adjust the volume of the specified lane.
- **Lane 1-8 Output Assignment** - allows you to use a widget to change the output assignment for the specified lane.
- **Lane 1-8 Filename** - displays the name of the song (file name without path or extension).
- **Lane 1-8 Length** - displays the lane length in seconds (the underlying parameter - always between 0.0 and 1.0 - represents a percentage of 10 hours).

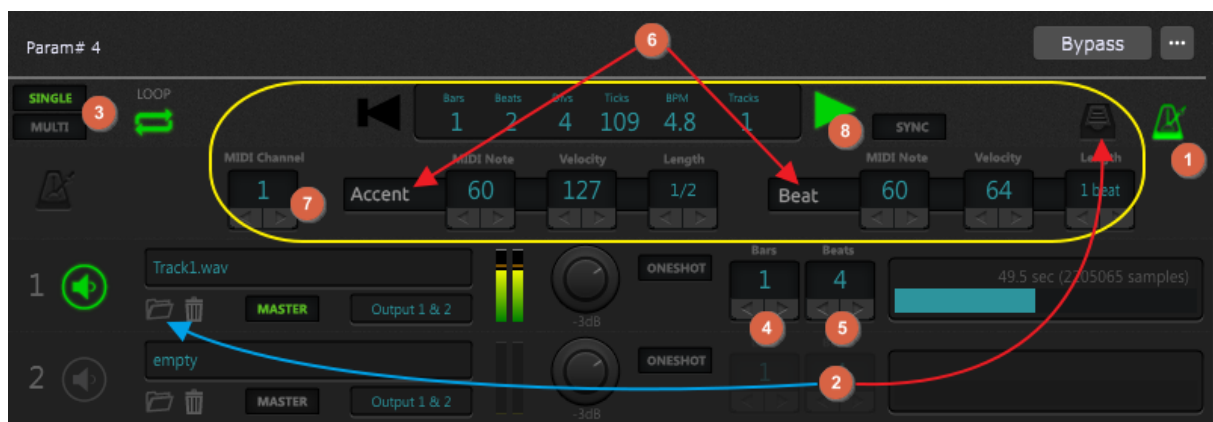
- **Largest Lane Length** - displays the length of the largest lane in the Audio File Player (the underlying parameter represents a percentage of 10 hours).

**Note:** we recommend the use of on-off widgets, such as one of the LED buttons, for real-time control over most Audio File Player parameters. However, **Playhead position**, **Marker position**, **Lane 1-8 Volume**, and **Lane 1-8 Output Assignment** are better controlled with continuous widgets such as knobs and sliders.

### 10.6.1.1 MIDI metronome

Included in the Audio File Player plugin is a MIDI metronome optimized for working with short loops. Tell the MIDI metronome the musical length of a loop and it will generate MIDI notes for a metronome. The MIDI metronome does not do tempo detection and is unrelated to the audio metronome.

MIDI metronome settings can be seen on the screenshot below (indicated with the yellow rounded rectangle):



Follow these steps to use the MIDI metronome:

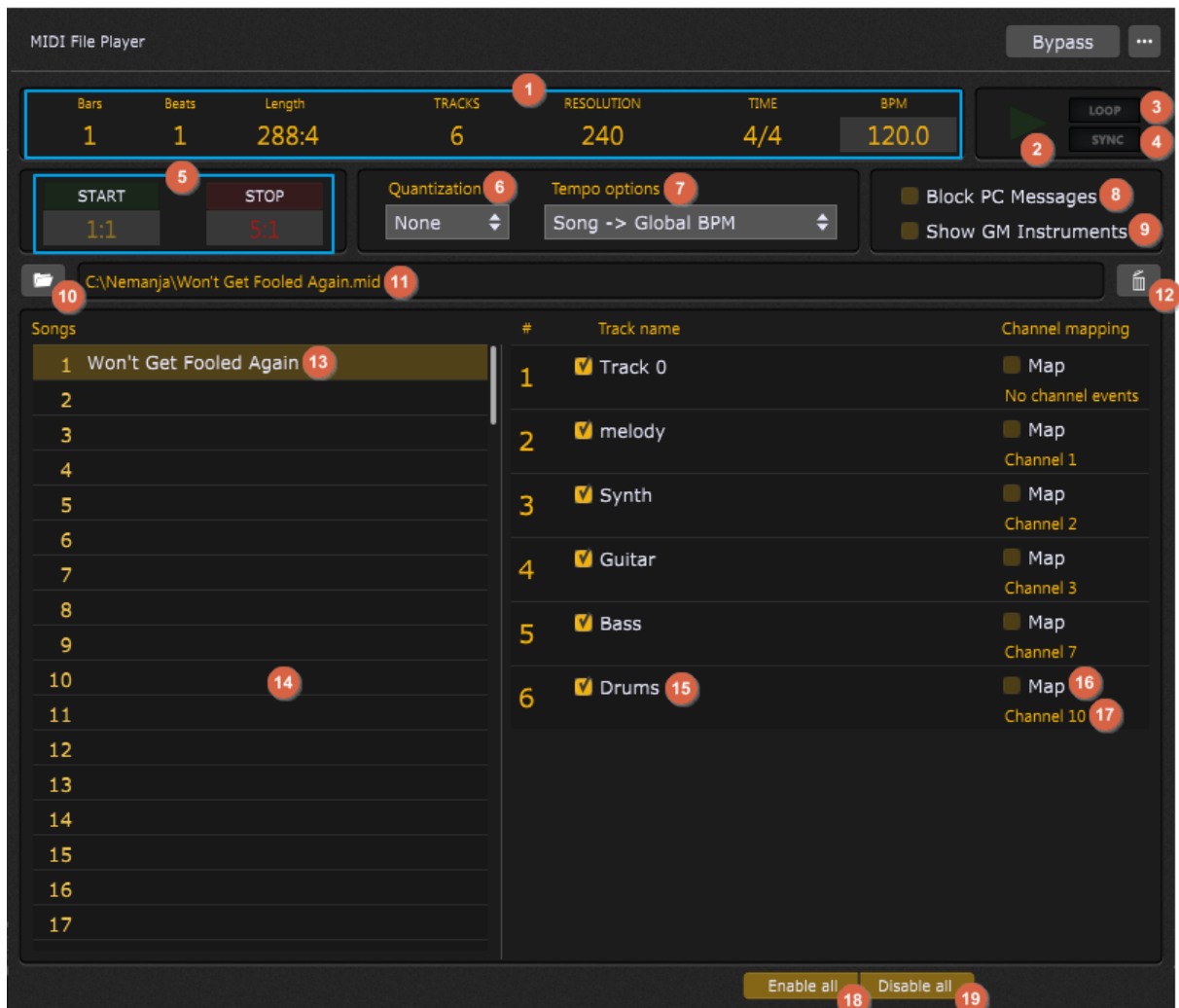
1. Click on the **Metronome** button **(1)** to show MIDI channel and note settings for the metronome (indicated with the yellow rounded rectangle).
2. Load a sound file into the audio player **(2)**. You can either load a single sound file (indicated with the blue arrow) or load a full directory of audio files into sample lanes (indicated with the red arrow). **Note:** if you want to play multiple sample lanes in parallel, click on the **Multi** button **(3)**. In Multi mode, the longest file will be designated as the Master and used to calculate the metronome.

3. Enter the duration of the loop in bars into the **Bars** field **(4)** on the lane of the Master track, by typing it in directly or using the +/- arrows beneath the field.
4. Enter the number of beats per bar in the loop into the **Beats** field **(5)** on the lane of the Master track by typing it in directly or using the +/- arrows beneath the field.
5. Choose or add a plugin as the sound source for the metronome. Choose the sound you want to use for the metronome and set a MIDI channel for it. Don't forget to make an audio connection from the sound source to your interface outputs.
6. In the Wiring view, connect the MIDI Out port of the Audio File Player to the MIDI In port of the plugin you want to use to generate the metronome sound.
7. Define the *MIDI note*, *velocity*, and *duration* (Length) for both the **Accent** and **Beat** metronome sounds **(6)**.
8. Set the **MIDI Channel** field to the channel number of the metronome sound **(7)**.
9. Click the **Play/Pause** button and the metronome will play, along with the audio files **(8)**.

**Note:** The MIDI metronome assumes the loop contains an exact number of bars, with the beginning of the file being bar 1, beat 1. If this is not the case, the metronome may not sync correctly with the Master track.

### 10.6.2 MIDI File Player

The **MIDI File Player** provides one MIDI output pin:



The MIDI File Player plugin can load up to 128 songs. For each song you can easily select which tracks to play and optionally map them to specific channels. A looping feature is also supported, where you can further specify both start and stop positions.

Double-clicking this plugin opens its editor window, which provides controls that enable the following operations:

1. **Information display** - displays the information about the selected song (and its corresponding MIDI file):
  - a. **Bars** - displays the current bar in the song.
  - b. **Beats** - displays the current beat within a bar in the song.
  - c. **Length** - displays the total length of a song, expressed in bars and beats (e.g. 288:4 implies 288 bars and 4 beats).

- d. **Tracks** - displays the number of tracks in the selected song.
  - e. **Resolution** - displays the resolution of the current song, expressed in the number of ticks.
  - f. **Time** - displays the time signature of the current song.
  - g. **BPM** - displays the initial beats per minute of the current song. This field can be edited with a custom value (up to 640 BPM).
2. **Play MIDI file** - clicking this button plays the current song. Click again to stop playing and go to the beginning of the song.
  3. **Loop** - clicking on this button enables you to loop the MIDI file continuously; for this feature to work, you need to specify the loop length - clicking on the Stop button and specifying the stop position is mandatory (see **(4)**).
  4. **Sync** - clicking this button synchronizes playback with the global playback in the main toolbar. Use this feature if you use multiple MIDI and/or audio file players and you want them to play or stop at the same time. Find out more about the global play button in [this chapter](#).
  5. **Start/Stop** - this section enables start and stop options for the song. If the Loop option is not enabled, specifying the Stop position will stop the MIDI File Player when that position is reached. Start and stop positions are expressed in bars and beats (5:1 implies 5 bars and 1 beat). If you specify the stop position, e.g. at 5:1, the playing stops at the last "tick" before 5:1 (i.e. 4:4). Note that you only need to specify the end position for the loop feature to work. In that case the loaded MIDI file will be played from the beginning of the file until the specified End position.
  6. **Quantization** - clicking on this drop-down list allows you to choose the desired quantization (None, 1, 1/2, 1/4, 1/8, 1/16). If you need a more sophisticated quantization function such as triplets or a certain amount of "humanize", then you should perform such operations on the MIDI file using your DAW and set the Quantization in this plugin to None.
  7. **Tempo options** - allows you to select the desired tempo option:
    - a. **Song Tempo** - allows the MIDI file to be played at its own tempo without impacting anything else.
    - b. **Song -> Global BPM** - clicking on this option sets the global tempo from the song tempo.

- c. **Global BPM -> Song** - selecting this option sets the song tempo from the global tempo.
  - d. **Manual BPM** - selecting this option allows you to set a custom BPM value for a song.
  - e. **Manual BPM -> Global BPM** - selecting this option sets the global tempo to the manually defined tempo.
8. **Block PC messages** - checking this option blocks program change events found in tracks from being sent out.
  9. **Show GM Instruments** - a General MIDI file typically has 16 tracks for various instruments, and there is a convention about the channel numbers. For example, drums are always on the channel 10. This option assumes the loaded MIDI file is a GM file and will display GM instrument names based on the program changes found in the tracks.
  10. **Load a new MIDI file button** - loads a new MIDI file (both .midi and .mid file extensions are supported).
  11. **Location display** - displays a location of the loaded MIDI file for the currently selected song.
  12. **Unload selected file** - clicking on this button unloads the selected MIDI file.
  13. **Song name** - displays the name of the selected song.
  14. **Songs list** - the MIDI File Player plugin supports loading up to 128 songs.
  15. **Track name** - shows the track name in the current song. By default, all tracks are enabled in the loaded MIDI file. Click on the corresponding checkbox to disable/enable a track.
  16. **Map channel** - checking this option allows you to map all MIDI events to the specified channel (by default, channel 1).
  17. **Channel information** - shows the assigned channel per the loaded MIDI file. You can remap this channel by checking the Map channel check box **(16)**.
  18. **Enable all tracks** - clicking this button enables all tracks in the selected song (by default, all tracks are enabled).
  19. **Disable all tracks** - clicking this button disabled all tracks in the selected song. Use this feature if you want to select only a few tracks to play.

### MIDI File Player parameters

The following parameters are available for the MIDI File Player plugin:

- **SongNumber** - allows you to select one of the available songs in slots 1-128.
- **SongDisplayMode** - allows you to select how the song name will be represented. You can choose between "Song name" (e.g. Won't be fooled again), "Song index" (e.g. 1) and "Both" (e.g. 1 - Won't be fooled again).
- **Play** - allows you to start or pause file playback.
- **PrevSongName** - displays the name of the previous song.
- **NextSongName** - displays the name of the next song.
- **GotoPrevSong** - goes to the previous song.
- **GotoNextSong** - goes to the next song.
- **Loop** - toggles the loop mode on/off.
- **MidiTempo** - allows you to select the desired tempo option (see the bullet item **(7)**).
- **Quantization** - allows you to choose the desired quantization (see the bullet item **(6)**).
- **StartStopSync** - starts or stops the Sync feature (see the bullet item **(4)**).
- **LoopStart** - toggles the Start setting (disabled/enabled).
- **LoopEnd** - toggles the End setting (disabled/enabled).
- *Reserved* - these parameter slots are not currently used.
- **BlockProgramChanges** - toggles blocking program change messages (Yes/No).
- **GeneralMidiMode** - toggles the General MIDI mode (Yes/No).
- **BarBeatDisplay** - displays bars and beats for a song.
- **CurrentPosition** - displays the current position (expressed in bars and beats).
- **TrackEnable\_#** - allows you to enable or disable the specified track (# represents a number between 1-128).

- **TrackName\_#** - displays the name of the specified track (# represents a number between 1-128).
- **MappingEnable\_#** - allows you to enable or disable the channel mapping (# represents a number between 1-128).
- **MapChannelNumber\_#** - allows you to specify the channel number (between 1 and 16) if the mapping is enabled (# represents a number between 1-128).
- **StepForward** - allows you to step forward a MIDI file (so-called the Finger Tap feature). [Click on this video](#) to see this feature in action.
- **StepReset** - resets the stepping position.
- **ManualBPM** - allows you to set the BPM value in the MIDI File Player when the tempo options are set to **Manual** or **Manual BPM** -> **Global BPM**.
- **CurrSongName** - displays the name of the current song.

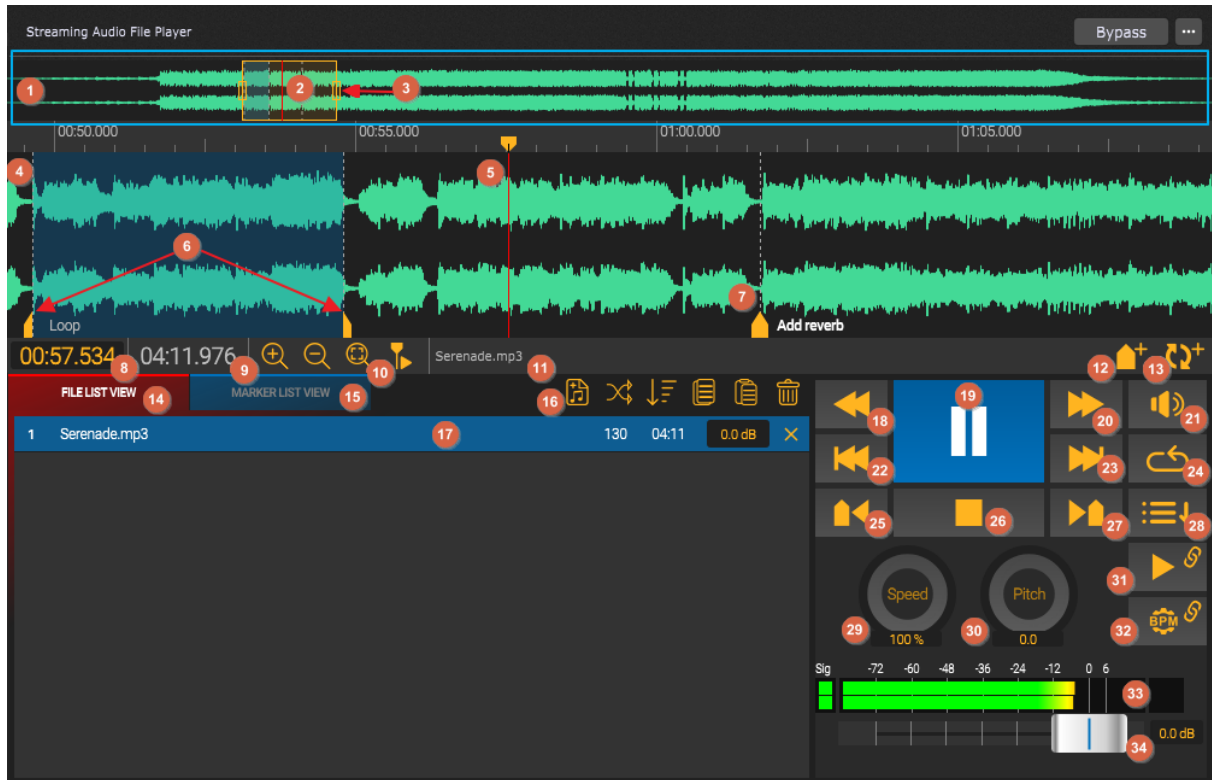
Use parameters such as PrevSongName, TrackName, or BarBeatDisplay with label widgets; insert a label widget into a rackspace (or the Global rackspace) and map it with these parameters to see the name displayed.

**Tip:** Read [this blog article](#) to learn how to improve the sound quality of instruments in your MIDI files.

### 10.6.3 Streaming Audio File Player

The **Streaming Audio File Player** streams audio files from disk, allowing you to have hundreds of songs in the player with minimum computer resources. Supported audio files are: MP3, WAV, OGG, FLAC, BWF, AIFF, AIF, ASF, WMV, WM, and WMA. The Streaming Audio File Player plugin provides two audio output pins. Double-clicking the plugin block opens its editor window, which provides controls that enable the following operations:

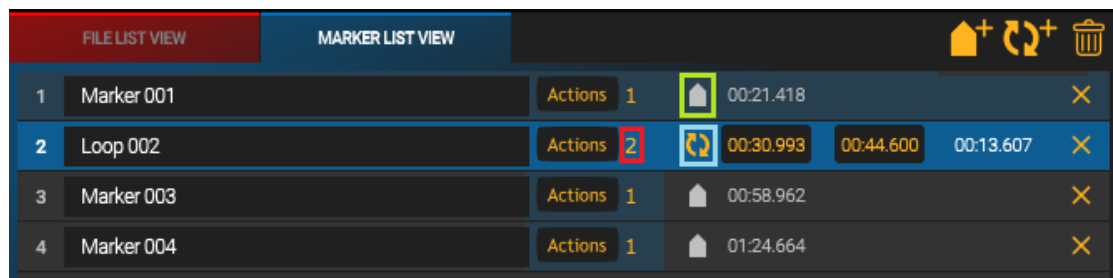




1. **Mini display** - represents a small display of the Timeline, along with loops, markers, and the playhead position.
2. **Current section** - represents the currently visible portion of the Timeline indicated with (4). Scrolling up or down with your mouse will move this selection to the right or left, respectively. If you hold down the **Ctrl** key (or the Command key on Mac) while scrolling up or down, then the selection will be zoomed in or out, respectively. Alternatively, you can set the zoom level using the buttons indicated with (10). Click on any part of the Mini Display (1) or simply drag it to move the current selection.
3. **Handles** - by clicking on the right or left handle and dragging right or left, you can adjust the displayed portion of the Timeline indicated with (4).
4. **Visible portion of the timeline** - in this area you can create markers and loop points. The size of this portion depends on the current selection, indicated in (2).
5. **Playhead** - indicates the playhead of the Streaming Audio File Player. The Playhead can be moved across the timeline when the track is not playing. The Playhead position is indicated with (8). To change the playhead position, you can:

- a. click anywhere on the Timeline; right-clicking on the timeline will have the same effect and also open a menu allowing you to add a marker or a loop point at the playhead's position.
  - b. drag the playhead across the timeline.
  - c. enter the playhead position manually (see **(8)**).
6. **Loop selection** - this area represents the loop selection and is defined by two loop points (indicated with two red arrows). You can change the loop selection by dragging the loop points. Alternatively, you can edit the loop point values manually in the Marker List View. Clicking on a loop point switches the view to the Marker List View **(15)**.
7. **Marker** - indicates a marker, a point in the timeline that can contain multiple [Actions](#), triggered when the playhead reaches the marker. Clicking on a marker switches the view to the Marker List View **(15)**. Double-clicking a marker opens the [Action management](#) window, which allows you to define the desired actions for that marker. To change the marker's position, you can drag the marker across the timeline or edit the marker's position manually in the Marker List View. Double-clicking a marker while holding down the **Shift** key allows you to execute all actions defined for that marker. Right-clicking the marker opens the contextual menu that allows you to *Copy marker actions*, *Paste marker actions*, and *Duplicate marker*.
8. **Playhead position** - displays the current position of the playhead **(5)**. You can click on this display and change the playhead position in the *minutes:seconds.milliseconds* format.
9. **Total track length** - displays the total track length.
10. **Zooming and following the playhead** - this section contains four buttons that allow you to:
  - a. **Zoom In** - zooms in the Timeline display.
  - b. **Zoom Out** - zooms out the Timeline display.
  - c. **Zoom 100%** - zooms the Timeline so the entire Timeline is displayed on the screen.
  - d. **Follow playhead position** - allows you to automatically scroll the Timeline window by following the playhead when a track is playing.
11. **Track name display** - displays the name of the currently loaded track.

12. **Add marker at playhead** - clicking this button adds a new marker at the playhead's position. **Note:** markers at the same location are not allowed.
13. **Add loop point at playhead** - clicking this button adds a new loop point at the playhead's position. Currently, it is only possible to define one loop area (i.e. to create two loop points). Simply click on this button to add the first loop point; click on this button again to add the second loop point. You can manually edit the time of loop points in the Marker List view (see **(15)**).
14. **File List View** - this view allows you to see the list of all audio files added to the player. To add files you can drag and drop audio files onto this window section or click on the *Add new audio files* button (see **16**). You can drag added tracks to re-order them. Double-click a track to load it.
15. **Marker List View** - Each track has its own list of markers. This view allows you to see the list of all markers and loop points associated with the currently selected track. The marker symbol is indicated with the light green rectangle and the loop symbol is indicated with the light blue rectangle.



Click on a marker or a loop to select it (alternatively, click on the marker's name on the timeline). Clicking on an already selected marker or loop will allow you to rename it. Double-clicking a marker or a loop will position the playhead at the marker's/loop's position. You can drag a marker/loop to change the order of markers. Clicking on the **Actions** button opens the [Action Management](#) window allowing you to define each marker's desired actions. The number of defined actions for each marker is indicated with the red rectangle. You can manually type in the position of each marker, and the loop points for the defined loop (the third value for the loop displays the loop duration). Clicking on the **X** button removes a marker/loop.

16. **View-specific buttons** - the button layout in this area depends on the current view.

If the [File List](#) view is active, then the following buttons are available:

- **Add new audio files** - click this button to add new audio files to the player.

- **Shuffle track list** - allows you to randomly rearrange the track list.
- **Sort by name** - allows you to sort the track list by name in ascending or descending order.
- **Copy all markers to clipboard** - the main use for this feature is to copy all markers from one track to another or to copy the markers, delete the track and replace it with an updated version of the same track and then put the markers back.
- **Paste all markers from clipboard** - pastes all markers from the clipboard.
- **Remove all tracks** - click this button to remove all tracks from the player.

If the Marker List view is active, then the following buttons are available:

- **Add marker at playhead** - allows you to add a new marker at the playhead's position.
  - **Add loop point at playhead** - allows you to add a new loop point at the playhead's position.
  - **Remove all markers** - click this button to remove all markers from the list.
17. **Track** - the currently loaded track. You can see the BPM of this track (e.g. 130 BPM in the screenshot above), the duration for this track, edit the dB (volume) value or remove the track from this player by clicking the **X** button.
  18. **Rewind** - clicking this button allows you to rewind the track by 10 seconds. Hold down the **Shift** key to rewind to the beginning.
  19. **Play/Pause** - clicking this button allows you to start or pause playing the currently loaded audio track. This button is disabled when the **Sync with host** feature is enabled.
  20. **Fast Forward** - allows you to fast forward the track by 10 seconds.
  21. **Mute** - click this button to mute the current track.
  22. **Previous Track** - click this button to load the previous track in the player. When the first track is reached, clicking this button will load the last track in the player.
  23. **Next Track** - click this button to load the next track in the player. When the last track is reached, clicking this button will load the first track in the player.
  24. **Play Loop** - when this button is turned on, clicking the Play button will start playing the defined loop.
  25. **Previous Marker** - clicking this button allows you to position the playhead to the previous marker.

26. **Stop** - click this button to stop playing an audio file.
27. **Next Marker** - clicking this button allows you to position the playhead to the next marker.
28. **Auto Play Next Track** - click this button if you want the next track to play automatically when the current track ends.
29. **Speed** - move this knob to change the speed of the loaded audio file (from 33% to 150%). Double-clicking this knob resets the value to the default (100%).
30. **Pitch** - move this knob to change the pitch of the loaded audio file (from -12 to +12). Double-clicking this knob resets the value to the default (0).
31. **Sync with host** - clicking on this button synchronizes the playhead with Gig Performer's Global playhead. When this button is enabled, the player's playhead is disabled, and the Global playhead controls play/pause operations.
32. **Sync BPM with host** - this feature allows you to detect a song's BPM and sync that tempo with the Global BPM.
33. **Level meters** - allows you to see the audio levels in real-time.
34. **Volume slider** - use this slider to set the desired volume value in dB. Double click this slider to reset to 0 dB.

### Streaming Audio File Player parameters

The following parameters are available for the Streaming Audio File Player plugin:

- **Mute** - allows you to mute and unmute the audio playback.
- **Loop** - allows you to play the loop specified by selected loop points.
- **Auto Play Next** - allows you to automatically play the next track when the current track finishes.
- **Play** - starts playing an audio file or pauses playing an audio file.
- **Next Track** - goes to the next track. When the last track is reached, it goes to the first track in the player.
- **Previous Track** - goes to the previous track. When the first track is reached, it goes to the last track in the player.

- **Rewind** - rewinds the audio file by 10 seconds.
- **Rewind To The Beginning** - rewinds the audio file to the beginning.
- **Fast Forward** - goes forward by 10 seconds.
- **Volume** - controls the volume.
- **Notify Playhead Position** - this parameter is implemented for cases where you don't need to track the the playhead's position. If you don't need the player to send out the position on every process block call (for example, every 2.9 milliseconds if you're running at 44.1 kHz with a 128 sample buffer size), you can turn it off.
- **Position** - allows you to control the position of the player's playhead. If you need to track the position explicitly (i.e. so it is synced with the playhead), make sure that the Notify Position parameter is On.
- **Track Number** - allows you to select the desired track in the player, up to 1000 tracks; the widget value 0 corresponds to the first track, 0.1 to the second track, and the maximum value 100 corresponds to the 1000th track.
- **Track Name** - shows the name of a currently loaded track. Attach it to a label widget.
- **Number Of Tracks** - displays the current number of loaded tracks in the player, up to 1000 tracks.
- **Sync to Host** - allows you to sync the player's playhead to the Gig Performer's Global playhead.
- **Next Marker** - goes to the next marker on the timeline relative to the current playhead position.
- **Previous Marker** - goes to the previous marker on the timeline relative to the current playhead position.
- **Triggered Marker Number** - displays the number of a currently triggered marker.
- **Go To Marker** - allows you to go to the desired marker, up to 1000 markers; the widget value 0 corresponds to the first marker, 0.1 to the second marker, and the maximum value 100 corresponds to the 1000th marker.
- **Trigger Marker Actions** - allows you to manually execute the actions defined for the desired marker number. The widget value 0 corresponds to the first marker, 0.1 to the second marker, and the maximum value 100 corresponds to the 1000th marker.

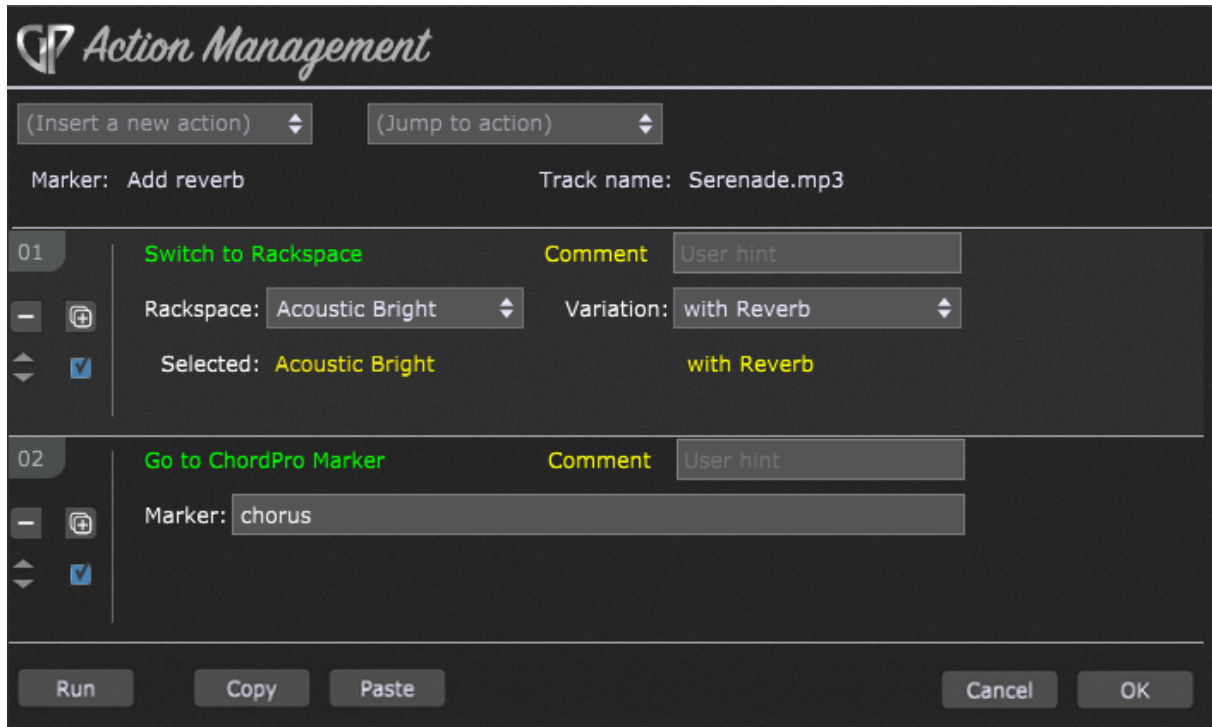
Even if you are not playing an audio track, this is a convenient way to trigger an arbitrary sequence of actions.

- **Number Of Markers** - displays the current number of markers for a track, up to 1000 tracks.
- **Stop** - stops playing an audio file.
- **Follow BPM** - allows you to synchronize BPM with host
- **Player Speed** - allows you to change the speed of a loaded audio track.
- **Player Pitch** - allows you to change the pitch of a loaded audio track.

Use parameters such as Mute, Loop, Auto Play Next, Play and Notify Position with button widgets. Parameters such as Volume and Position should typically be mapped to knobs or sliders. Finally, parameters such as Number Of Tracks and Number Of Markers should typically be mapped to label widgets.

#### 10.6.3.1 Action Management

You can add markers at various points in a track in the [Streaming Audio File Player](#) to cause one or more actions to be triggered automatically as the playhead passes those markers. Double-clicking a marker symbol on the timeline or clicking on the Actions button opens the Action Management window:



This window allows you to manage actions for each marker. Actions allow you, for example, to switch rackspaces, switch songs, set widget values and captions, start or stop the Metronome, go to a specific position in your lyrics file, send MIDI events, or simply display a message in the main Gig Performer window.

**Note:** the [Song Part Properties](#) window also allows you to trigger actions when that song part is activated.

There are 21 action types supported, and they can be combined.

You can click on the **Run** button to trigger all defined actions before saving them. Further, you can click **Copy** to copy all actions to the Clipboard and **Paste** to paste all actions previously copied to the Clipboard.

All actions include the Comment field, where you can optionally include a hint for the desired action. This hint is shown in the 'Jump to action' combo box (indicated with the red rectangle).



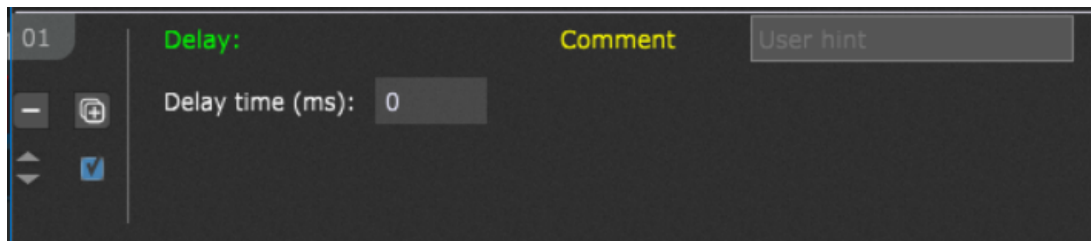


**Note:** a number of actions relies on widget and plugin handles. To learn about handles in Gig Performer, see the blog article [What is a handle and how to use it](#).

All these action types will be described below.

### 1. Delay

The Delay action allows you to add a delay of up to 1000 milliseconds.

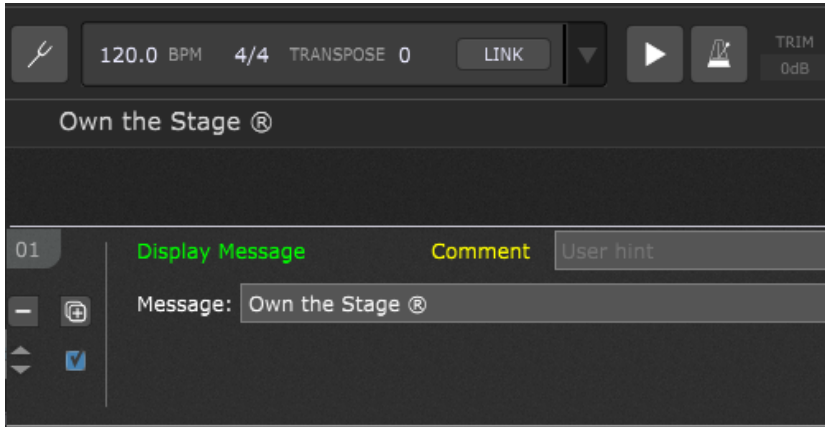


The intent of this action is to interact with external synthesizers that can't handle receiving multiple MIDI messages (such as bank select followed by program change or a program change followed by a CC message to adjust the volume) that arrive too quickly.

Note: this action is synchronous; all subsequent actions processing will be delayed as well.

### 2. Display Message

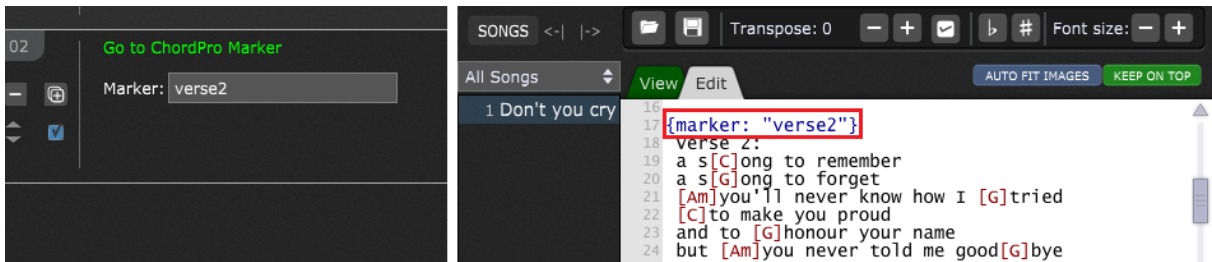
The Display Message action simply flashes the specified message in the main Gig Performer's window:



If there are multiple messages on a single marker, then only the last message will be displayed.

### 3. Go to ChordPro Marker

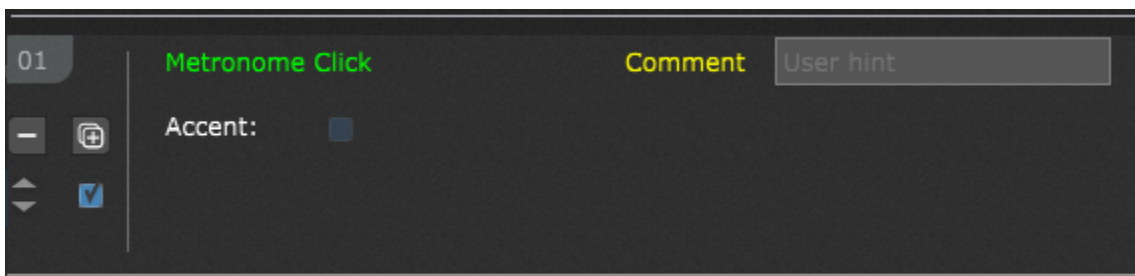
This action assumes that you have the *marker* directive defined in your ChordPro file, e.g. `{marker: "verse2"}`, as indicated with the red rectangle in the screenshot below:



In the Go to ChordPro Marker action you need to enter the marker name without quotes. When the playhead reaches this marker, it will move the ChordPro file to the specified position (in this example, to show Verse 2).

### 4. Metronome Click

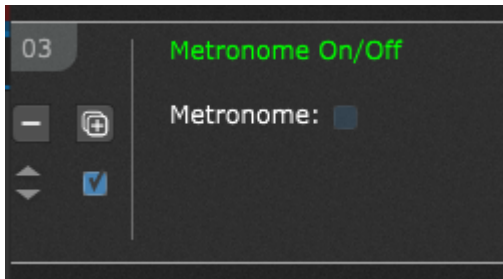
This action allows you send a single metronome click.



Tick the Accent checkbox if you need to trigger an accented beat.

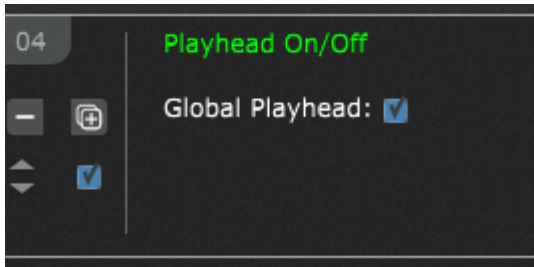
## 5. Metronome On/Off

This action starts the Gig Performer Metronome feature when the check box is ticked, or stops the Metronome when the checkbox is not ticked:



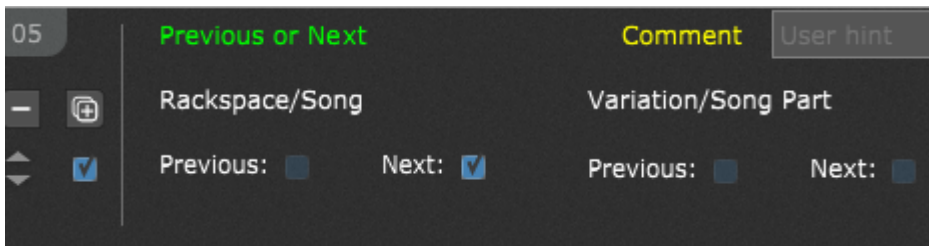
## 6. Playhead On/Off

When the check box is ticked, this action starts the Gig Performer Global playhead. When the checkbox is not ticked, the Global playhead stops playing:



## 7. Previous or Next

While the Panels view is displayed, this action allows you to go to the previous or next rackspace or variation.



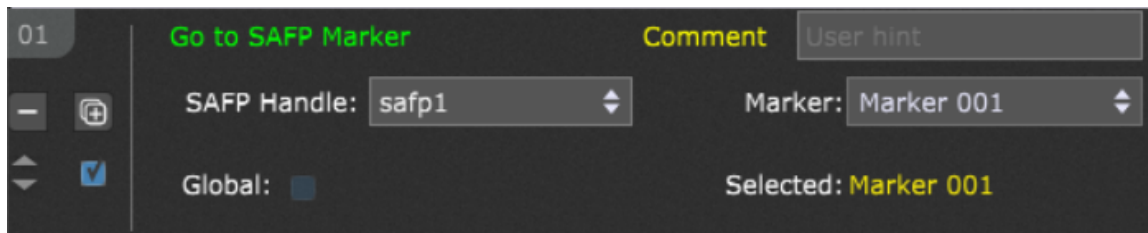
If the Setlist view is displayed, this action allows you to go to the previous or next song or song part. Note that the check boxes are mutually exclusive, i.e. you cannot go to the

next and previous rackspace simultaneously, and you cannot go to the next rackspace and the next variation simultaneously.

Note: this action is not available as a song part action.

## 8. SAFP - Go to Marker

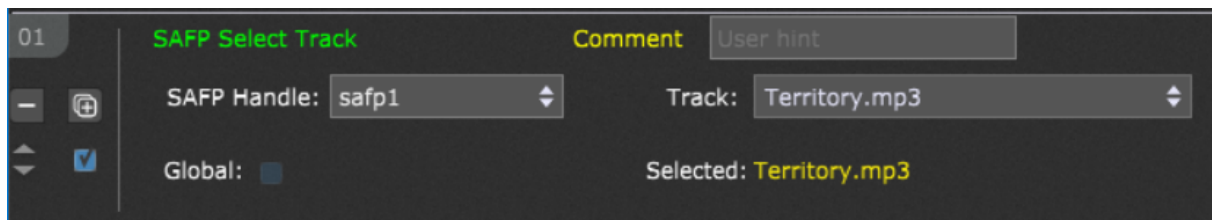
This action allows you to jump to the desired marker that is specified in a Streaming Audio File Player block.



Note: this action is only available as a song part action.

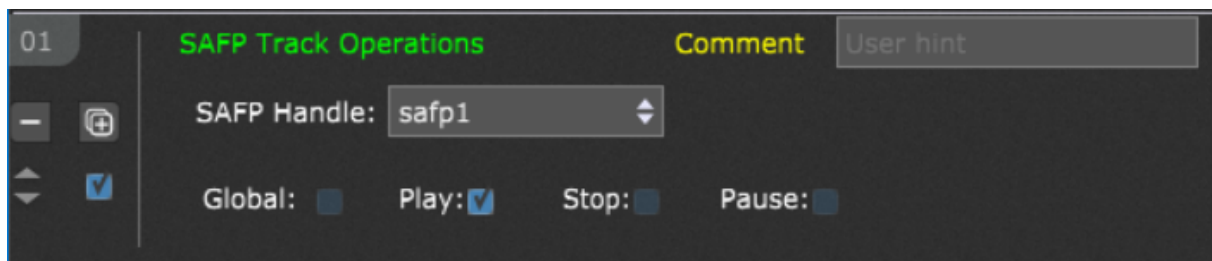
## 9. SAFP - Select Track

This action allows you to select the desired track in the desired Streaming Audio File Player block.



## 10. SAFP - Track Operations

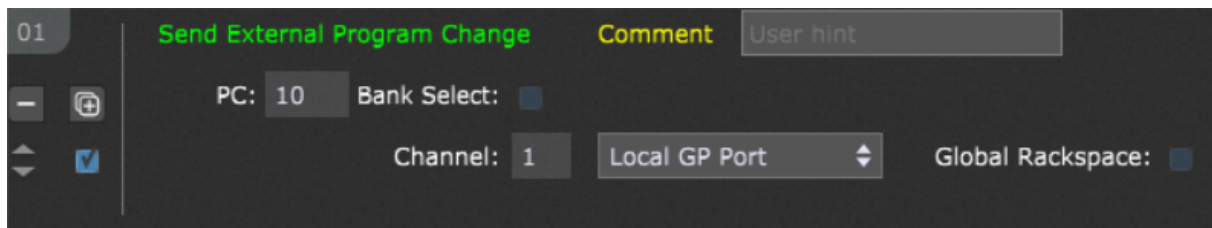
Supported track operations in this actions are Play, Stop, and Pause.



After you select the SAFP handle, simply check the desired operation.

## 11. Send External Program Change

This action allows you to send a program change message with optional bank select messages to external hardware.



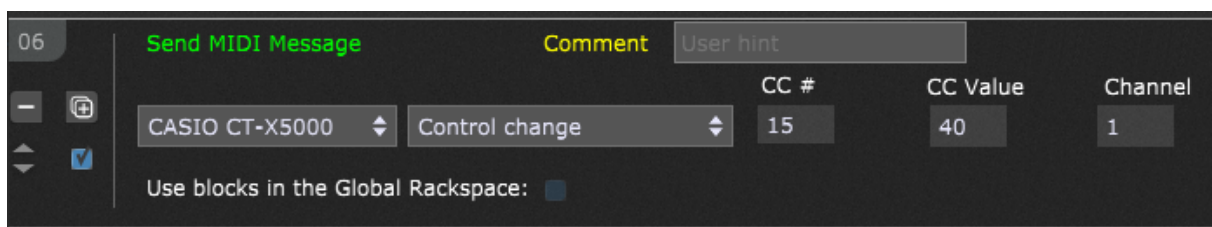
Type in the desired program change number and optionally select bank values (MSB and LSB). Afterward, specify the desired MIDI channel and a MIDI target. Supported targets are: Physical MIDI Out devices, [Local GP Port](#), MIDI Out blocks with handles, MIDI In blocks with handles, Global MIDI Out blocks with handles and Global MIDI In blocks with handles.

If your plugin block is in the Global rackspace, be sure to check the *Global Rackspace* checkbox.

Note: this action is only available as a song part action.

## 12. Send MIDI Message

This action allows you to send a MIDI message to a specific block in a rackspace:



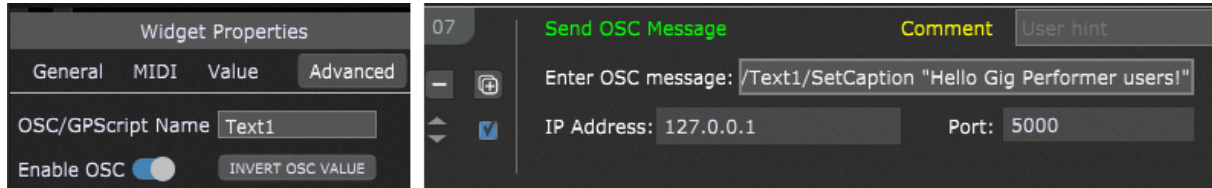
Supported MIDI messages are: Note On, Note Off, Control Change, Program Change, Aftertouch, Pitch wheel, Polytouch, SysEx, Song Position, Song select, Tune request, RT Timing clock, Start, Continue, Stop, and Reset.

Supported blocks are: Physical MIDI Out devices, [Local GP Port](#), MIDI Out blocks with handles, MIDI In blocks with handles, Global MIDI Out blocks with handles and Global MIDI In blocks with handles.

Check the *Use blocks in the Global Rackspace* check box if your MIDI block is located in the Global rackspace.

### 13. Send OSC message

This action allows you to send the desired OSC message to the desired target (defined with the IP address and the port).

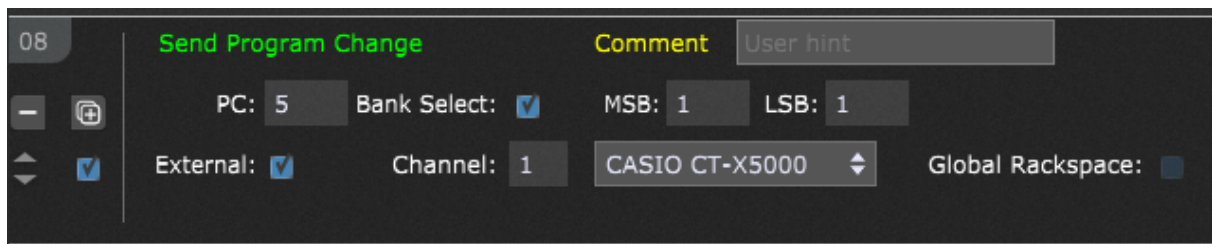


For example, if you add a label widget and give it the OSC name **Text1**, then you can easily change its caption to **Hello Gig Performer users!** using the OSC message in the screenshot above (provided that OSC is enabled in the [OSC Options](#) and that Gig Performer listens at the port 5000) when the playhead reaches the marker that contains this action.

An OSC message can contain an arbitrary number of integer, float or string arguments. You can trigger arbitrary GPScript functions from the timeline by sending OSC messages into Gig Performer scripts. That's why the Actions Management window can easily accommodate the most complex automation use cases.

### 14. Send Program Change

This action allows you to send a program change message to Gig Performer itself or an external device.



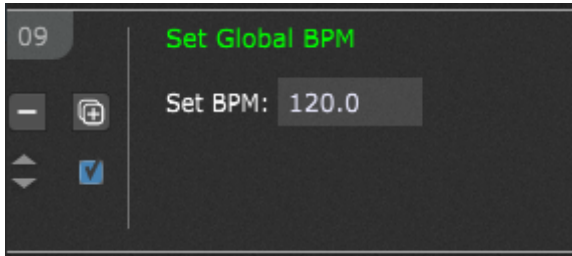
Simply type in the desired program change number and optionally select bank values (MSB and LSB); if you want to send a program change message to an external device, check the External check box and then specify the desired MIDI channel and a target. Supported targets are: Physical MIDI Out devices, [Local GP Port](#), MIDI Out blocks with handles, MIDI In blocks with handles, Global MIDI Out blocks with handles and Global MIDI In blocks with handles.

Further, if you want to send a PC message to a block located in the Global rackspace, check the **Global rackspace** check box.

Note: this action is not available as a song part action.

## 15. Set Global BPM

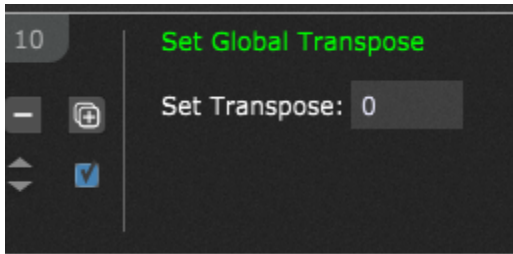
This action allows you to set the desired Global BPM value:



You can select a value between 20 and 640.

## 16. Set Global Transpose

This action allows you to set the desired Global BPM value:



You can select a value between -48 and +48.

## 17. Set Plugin Parameter Value

This action allows you to set a plugin's parameter value directly, i.e. without using widgets.



Select the plugin's handle, select the desired parameter from the list, and then specify the desired value for that parameter.

If you want to bypass the selected plugin, check the Bypass check box.

### 18. Set Widget Caption

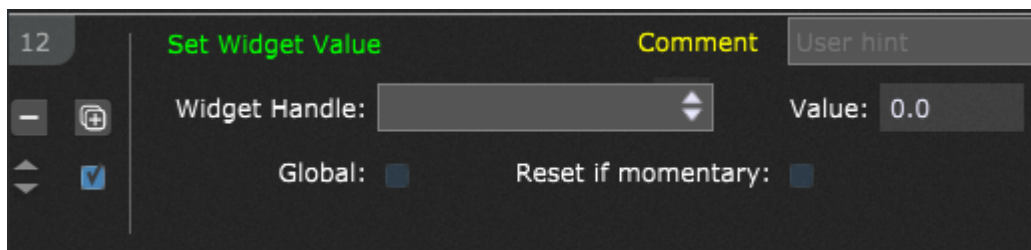
This action allows you to set the caption of any widget:



To set a widget caption, you must first select a widget handle, i.e. the OSC/GPScript name in the Advanced tab from the [Widget Properties Inspector](#). If the widget is located in the Global rackspace, make sure to click on the Global check box.

### 19. Set Widget Value

This action allows you to set the desired widget value:



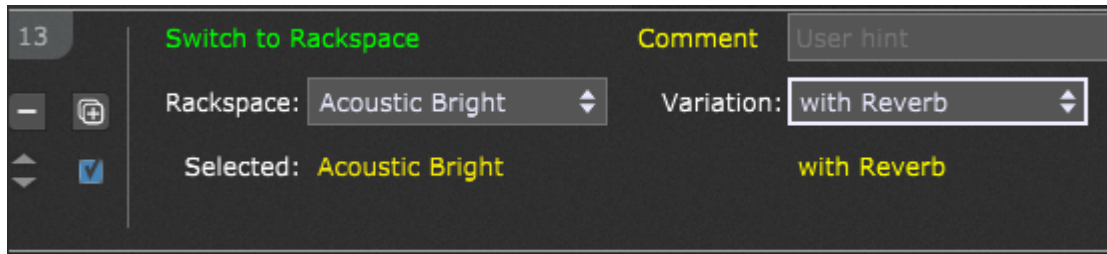
To set a widget value, you must first select a widget handle, i.e. the OSC/GPScript name in the Advanced tab from the [Widget Properties Inspector](#).

If the widget is located in the Global rackspace, click on the Global check box. Tick the *Reset if momentary* check box if your widget is set to be 'Momentary' in the Widget Properties inspector and you want to reset its value.

### 20. Switch to Rackspace

This action allows you to switch to the desired rackspace:





Simply select the desired rackspace and variation from the combo boxes. The selected rackspace and variation will be displayed in yellow color.

Note: this action is not available as a song part action.

## 21. Switch to Song

This action allows you to switch to the desired song:

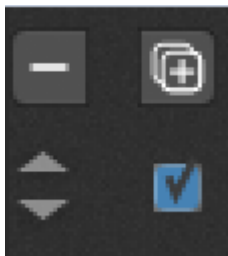


Select the desired song and song part from the combo boxes (the first song part is selected by default).

By default, songs from the current setlist will be displayed. If you want to display all songs (from the All Songs setlist), check the **All songs** checkbox. The selected song and song part will be displayed in yellow color.

Note: this action is not available as a song part action.

Every action has these common controls:

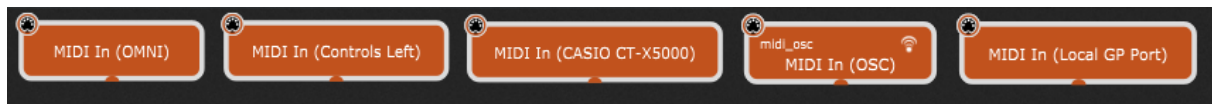


These controls allow you to remove an action, duplicate an action (in case you want to add similar actions, such as sending multiple MIDI messages), move an action up, move an action down, and enable an action. If you want to disable an action, untick the check box.

Note: you can also define action in the Song Part Properties window. Refer to [this chapter](#) to learn more.

## 10.7 MIDI Inputs

These plugins are used to receive incoming MIDI messages and pass them on to other plugins; they provide a single MIDI output pin:

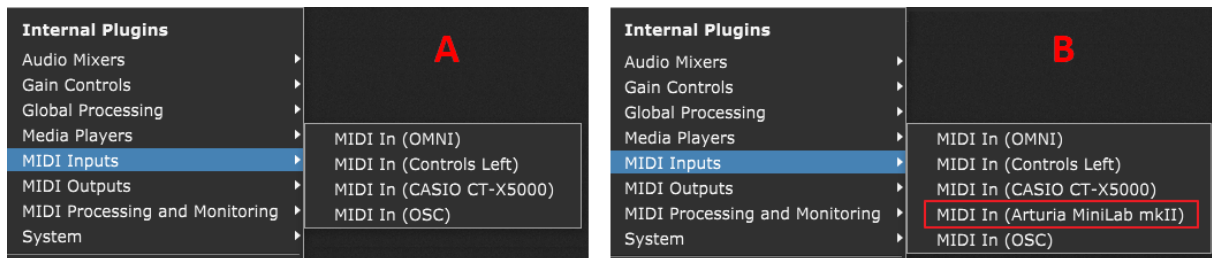


The image above will be different for your setup as it depends on the currently connected MIDI devices and aliases that you might have configured for your currently connected MIDI devices.

1. **MIDI In (OMNI)** - receives MIDI messages coming from any MIDI Input port on your system. The MIDI In (OMNI) is a generic plugin that "listens" to all incoming MIDI messages from all connected controllers, on all channels, and on all ports. It is the default MIDI input block in any new rackspace. You can delete it if you don't need it (for example, if you are using plugins to process incoming audio signals only from a guitar or a microphone) and add it later if needed. You can also duplicate this plugin if needed. This block is designed for simple setups - use this when you're getting started. Once you start working with multiple keyboards and/or control surfaces, you should switch to MIDI In blocks associated specifically with each device to route MIDI appropriately.  
**Note:** if you are using a virtual MIDI Port (such as the built-in IAC port in Mac), you must not use OMNI blocks as you will get feedback ([click here](#) to learn more).
2. **MIDI In (Controls left)** - "Controls left" is a custom alias that this author defined in the Rig Manager for the specific MIDI input device. This alias is tied to the specific MIDI Input port on this author's computer, and this block receives MIDI events only from the that input port. Refer to the [Rig Manager](#) chapter to find out more about aliases.
3. **MIDI In (CASIO CT-X5000)** - for this connected MIDI input device, an alias has not been defined. The text in brackets is the full name of the MIDI device and it depends on the MIDI device's manufacturer. This block receives MIDI events only from the specific input port where this MIDI device is connected.

4. **MIDI in (OSC)** - receives MIDI messages only from OSC sources. For more information on OSC, [click here](#).
5. **Local GP Port** - this block is intended for very advanced setups. Check the [Local GP Port](#) chapter to learn more.

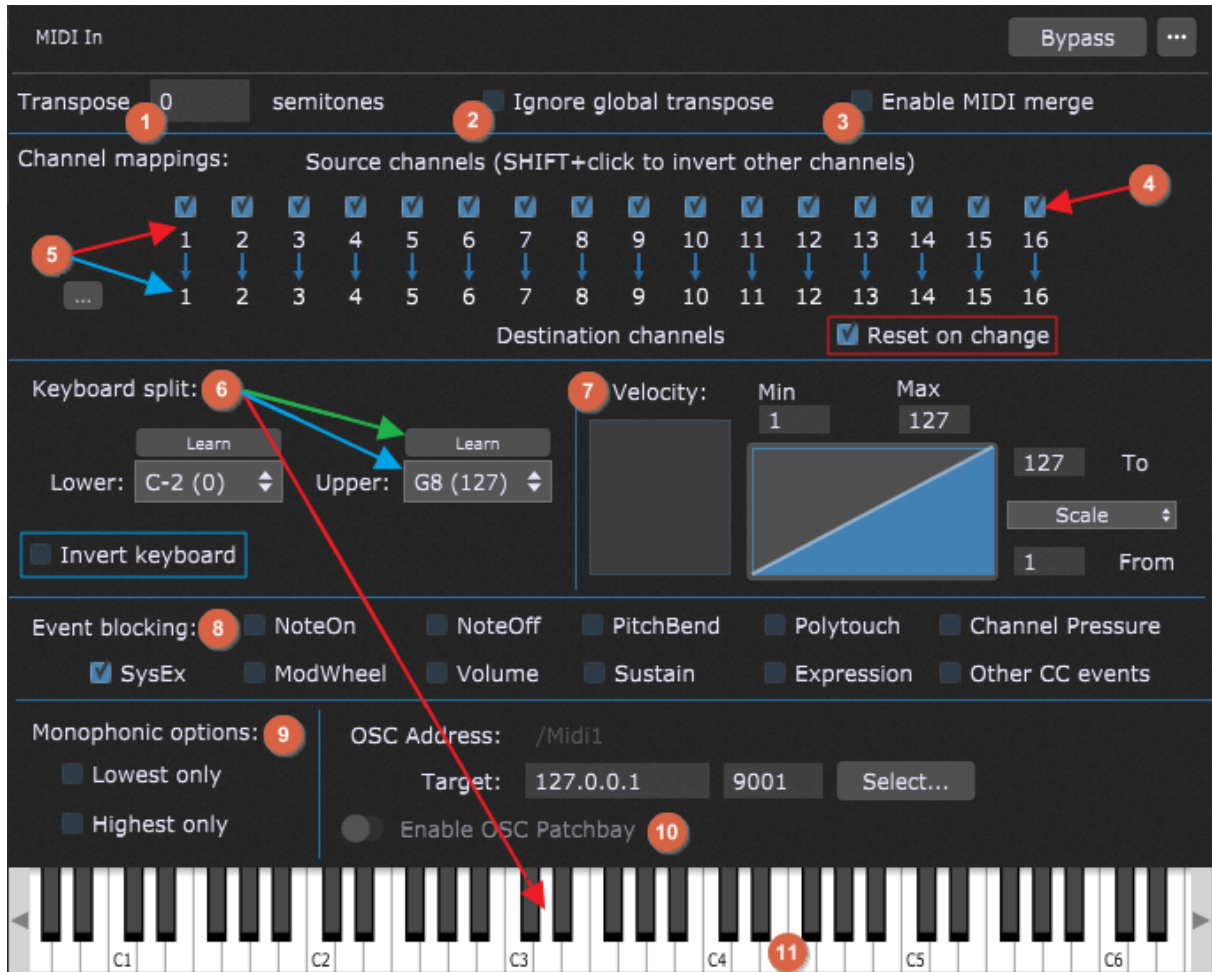
If you connect one or more MIDI controllers to your system, Gig Performer automatically generates additional controller-specific MIDI In blocks; right-click on the empty surface in the [Wiring view](#) to open the [contextual menu](#):



Then expand the MIDI Inputs sub-menu to see the list of all connected MIDI In devices. On the screenshot above note the state before (A) and after (B) connecting your MIDI device. The actual name of your connected MIDI device (indicated with the red rectangle) depends on the device's manufacturer.

These can (and, in most cases, should) be substituted for the generic MIDI In (OMNI) plugin. Because they "listen" to specific MIDI controllers and/or ports, their usage is generally preferable, especially if your rig contains multiple keyboards or controllers. Using specific MIDI In blocks makes it easy for you to route incoming MIDI from each device to different plugins.

Double-clicking a MIDI In plugin opens its editor window:

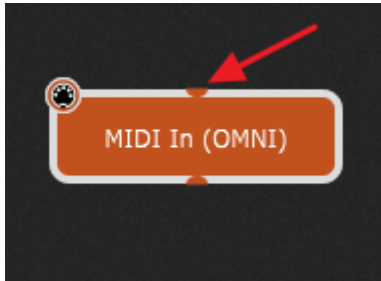


The MIDI In plugin editors provide the following functionality:

1. **Key transposition** - transpose incoming note messages over a range of +/- 127 semitones. Each MIDI In block can set its own transpose value. This is particularly useful when you use keyboard splits where you want to position (say) a piano at the bottom two octaves of your keyboard but obviously you want the actual notes to be pitched at higher octaves. Normally the MIDI In block's transpose value is added to the global transpose since you generally want to transpose all your instruments by the same amount.
2. **Ignore global transpose** - checking this option ignores the global transpose that is set in the upper toolbar of the main Gig Performer window (refer to [The main Gig Performer window](#) chapter, bullet item 8 for more information). For example, suppose you are using a particular MIDI In block to control percussion or sound effects in a sampler. In that case, you generally do not want the transpose to be changed, so you can configure individual MIDI In blocks to ignore the global transpose value.

3. **Enable MIDI merge** - this option has a very specialized purpose as it lets you feed in MIDI events from other plugins that produce MIDI (arpeggiators, MIDI file players, MIDI sequencers etc.). After this point, they can be processed using other parameters available here and by [GPScript](#), for example.

After you enable this option, a MIDI input block (e.g. the MIDI In (OMNI) in the screenshot below) will display the MIDI input pin:

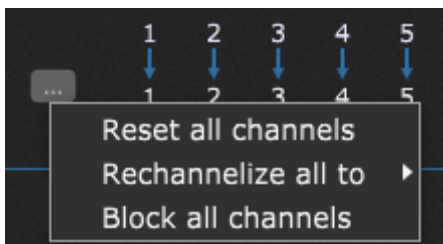


This way you can leverage some of the MIDI In block's functionality. For example, it's an easy way to do channel remapping (instead of using multiple [MIDI Channel Constrainer](#) plugins) and you can use it to block certain kinds of events, etc.

4. **Channel blocking** - specified MIDI channels can be blocked by simply unchecking the box above the desired source channel. Hold the **Shift** key and click on a channel to disable/enable all other channels.

5. **Channel mapping** - a source channel (indicated with the red arrow) can be mapped to a different destination channel (indicated with the blue arrow); click on the destination channel and type in a number to map to the different destination channel. The **Reset on change** checkbox (indicated with the red rectangle) is checked by default, allowing you to stop all pending notes after any channel change.

Also note the "..." button:



It allows you to quickly reset all channels to default, rechannelize all source channels to the desired destination channel or block all channels.

6. **Keyboard split** - enables you to create keyboard splits by specifying the lower and upper notes of the split. Notes below the Lower value and/or above the Upper value are

blocked; only notes between those values are allowed through. The Lower and Upper values can be entered in one of these ways:

- a) click on the Lower note or Upper note field and choose the desired note from the corresponding drop-down menu (indicated with the blue arrow).
- b) click on the Learn button (indicated with the green arrow) above the Lower note or Upper note field and play the desired note on your MIDI controller. Click the Learn button again to complete the setting.
- c) click on the Learn button above the Lower note field and click on the desired low note on the plugin's virtual keyboard (indicated with the red arrow); then click the Learn button above the Upper note field and click the desired high note on the plugin's virtual keyboard. Remember to click the Learn button again after the note is learned to exit Learn mode.

It is easy to see the valid split range because notes below the Lower note and above the Upper note are grayed out. Check out our blog article [How to create keyboard and velocity splits](#) for in-depth guidelines.

**Note:** the keyboard in the MIDI In block plugin shows the physical key being pressed when you play your controller rather than the output value, which might be transposed.

Ticking the **Invert keyboard** checkbox (indicated with the blue rectangle) allows you to produce lower notes when playing higher up the keyboard (and vice-versa). This is a nod to Joe Zawinul who famously inverted the ARP 2600 keyboard when he performed Black Market. It makes it easy to do certain counterpoint as you can use the same fingering on both hands.

7. **Velocity** - the old *Velocity Layering and Scaling* options have been replaced by the *Velocity Curve* which gives you much more flexibility to map your input velocities. You can implement the following features:

- a) **Velocity layering** - velocities below the MIN value and/or above the MAX value are blocked; only velocities between those values are allowed through. If the same value is entered for MIN and MAX, all incoming notes will have their velocity changed to that value. This allows you to create velocity layers by specifying whether notes are heard depending on how hard (or soft) you press a key.
- b) **Velocity scaling** - by selecting "Scale" in the drop-down box and entering "From" and "To" values, the velocity of all incoming note messages is adjusted to reduce their dynamic range (i.e. softer notes will be heard louder, or louder notes will be heard more softly).

c) **Velocity constraining** - by selecting "Constrain" in the drop-down box, any incoming note messages with a velocity below the "From" value are given that value exactly; at the same time, any incoming note messages with a velocity above the "To" value are given that value exactly. For example, if you enter a "From" value of 60, and a "To" value of 80, all incoming note messages with velocities of 60 or less are given a velocity of 60 exactly, while all incoming messages with velocities of 80 or more are given a velocity of 80 exactly; only velocity values from 61 - 79 are preserved. Constraining allows you to create hard velocity splits.

The MIDI In plugin editor's velocity range control is particularly useful if you want to have multiple sounds available but hear only some when you play softly and others when you play harder. If connected to a piano plugin, for example, you can opt to only hear the piano sound when you play the keyboard forcefully.

**Tip:** after you have defined a velocity range, you may want to change the actual velocities that are sent to your sound-generating plugin. For example, you may have created a piano sound that you only want to be heard if the velocity value is greater than 80. However, you may still want the piano to respond as if it was being played quietly. In such a case, set the Min/Max velocity values to 80 and 127 respectively so that you only hear the piano when you hit the keys hard. Then choose Scale and set the From/To values to 1 and 50, respectively. As a result, notes received with a velocity of 80 will produce notes with a velocity of 1, while notes received with a velocity of 127 will produce notes with a velocity of 50. Notes in-between those values will be mapped somewhere in between.

**8. Event Blocking** - sometimes it is desirable to block certain note or continuous controller messages coming from a controller. Although the MIDI Filter plugin is capable of this kind of filtering and much more, MIDI In blocks include an event blocking section that conveniently provides basic blocking capabilities to reduce the number of situations requiring the MIDI Filter plugin.

A common scenario is that your controller has one or more knob, button, or other physical controls that are not being used and so not assigned to a plugin parameter. If this unassigned control accidentally gets nudged, it could unintentionally change a plugin value. Blocking the MIDI messages coming from that control prevents this accidental modification. You might also want to use the Sustain pedal associated with another physical keyboard, but you do not want notes played on that other keyboard to come through (you'll route them to some other synth).

The **Other CC events** box, when checked, blocks all continuous controller messages except for the ones that have their own boxes (i.e. ModWheel (CC1), Volume (CC7), and Sustain

(CC64)). SysEx (System Exclusive) messages are blocked by default. Uncheck this box if you want to pass these messages to the MIDI In block.

9. **Monophonic options** - this area at the bottom of the editor filters the note messages from the controller down to a monophonic stream of notes.

- The **Lowest only** box, when checked, produces a mono part extracted from the lowest note as played on the instrument.

- The **Highest only** box, unsurprisingly, produces a mono part extracted from the highest note as played on the instrument.

These options can be useful for doubling melodies, or extracting a bass part from what might have been written as a piano part. Most often, users choose to check one or the other of these boxes, but you could check both, in which case you would be extracting lines from both the top and bottom of what is being played. Check out [this blog article](#) to see an example using Monophonic options.

10. **Enable OSC Patchbay** (this section shows up when the *Enable MIDI In OSC Patchbay* option is enabled in [OSC Options](#)) - turn on this toggle button if want to send out incoming MIDI messages as OSC messages to another application to be modified. The external application must send back modified messages, as otherwise no sound will be produced from the MIDI In block. Type in the OSC address prefix for this patchbay and IP address and port for the remote device (target) to which you want to send the message (alternatively, click on the **Select...** button to easily select an OSC target from the list that must be previously set in [OSC Options](#)). This is an advanced option and is not required for normal use.

Warning: if you enable the OSC Patchbay and you do not have an external application that is sending back modified messages, no MIDI messages will be sent to connected plugin blocks.

11. **Virtual keyboard** - the virtual keyboard is used to help you set and display the current keyboard split and to let you generate Note messages, allowing instrument plugins to be played even when an external MIDI controller is not connected. You can click on this virtual keyboard to play a key or use the computer keyboard instead. The MIDI channel used for these note messages is determined by the lower field associated with channel 1 in the Channel Mapping section.

### **MIDI In parameters**



The following parameters for MIDI In plugins are available:

- **Channel 1-16 Output** - allows you to change MIDI channel blocking or remapping in real time.
- **MinNote** - allows you to change the keyboard split minimum (lower) note setting.
- **MaxNote** - allows you to change the keyboard split maximum (upper) note setting.
- **Transpose** - allows you to alter the key transposition. Caution: do not change this setting while holding notes or stuck notes may result. If you do get a stuck note, don't panic - just click on the Panic icon to turn all notes off.
- **StopAllNotes** - issues a "turn all notes off" message, along with a sustain OFF message and a Center PitchBend message, same as the Panic button except that it only applies to the specific MIDI In block.
- **StickyNoteChannel** - determines the MIDI channel for the sticky note. Sticky notes allow you to trigger loops or sample playback from a widget. It works by sending Note On events only, as if you played a note but didn't release it. To use sticky notes, create an on-off widget (i.e. a switch or a button) and assign it to the MIDI In plugin and a Note On parameter. Then add a second continuous (knob or slider) widget and assign it to the MIDI In plugin as well, but to the StickyNoteVelocity parameter. When you activate the switch or button widget, a note will play continuously in the connected sound-generating plugin, with the knob widget determining the velocity of the MIDI note message.
- **StickyNoteVelocity** - determines the velocity of the sticky note.
- **Note On C-2 through G8** (MIDI note numbers 0 - 127) - toggles On/Off (SendNoteOn/SendNoteOff); default velocity is 64.
- **MinNoteVelocity** - allows you to pass through only notes with velocities equal to or greater than the minimum note velocity.
- **MaxNoteVelocity** - allows you to pass through only notes with velocities equal to or less than the maximum note velocity.
- **MinScaledNoteVelocity** - the lowest actual velocity will be adjusted to the minimum scaled velocity.
- **MaxScaledNoteVelocity** - the highest actual velocity will be adjusted to the maximum scaled velocity. All notes between are scaled according to the scale curve.

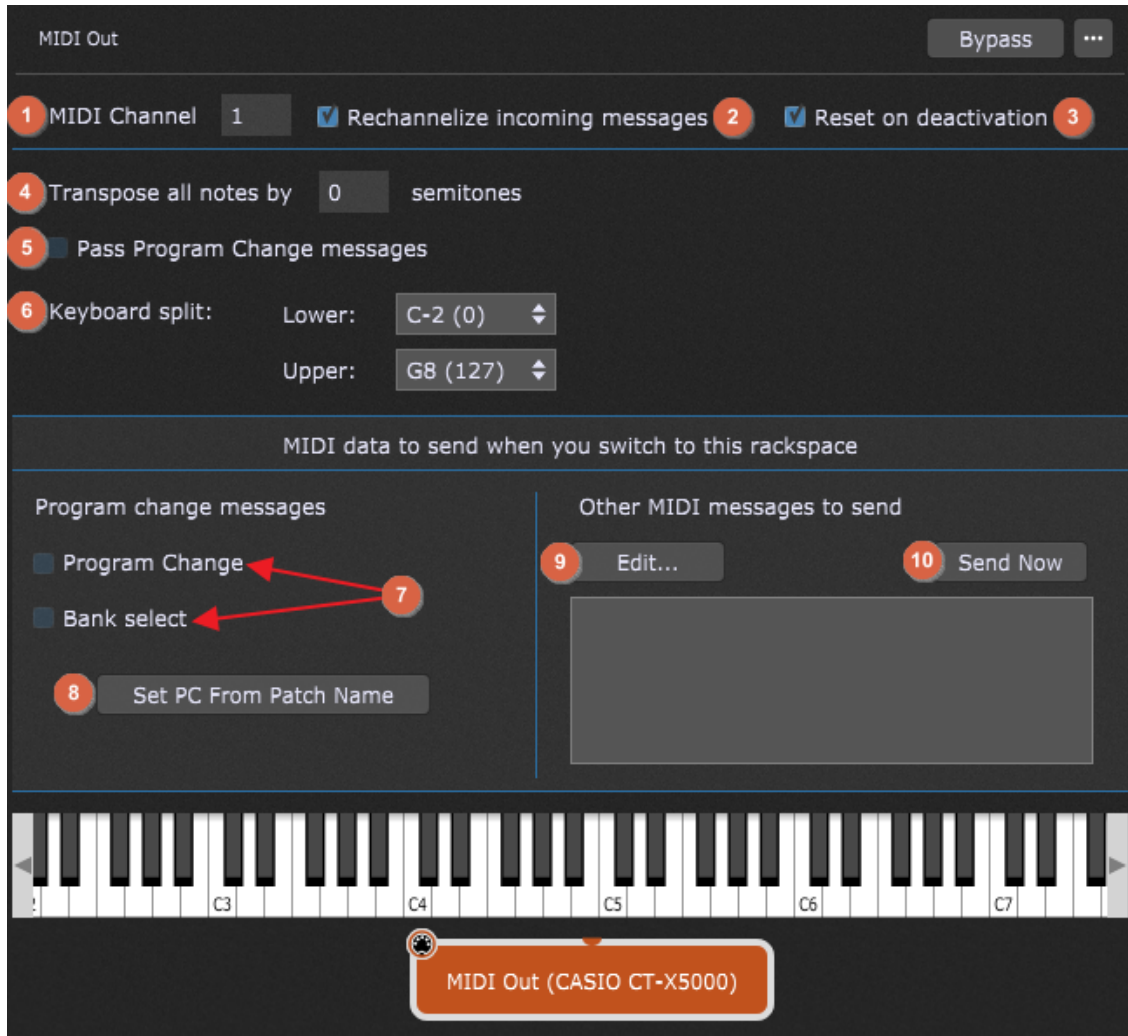
- **OutputVelocityMode** - allows you to change between Scaled and Constrained velocity (to scale or constrain incoming velocity values; the latter puts a hard min/max limit).
- **PC** (Program Change) - allows widget movements to send MIDI program change messages to plugins to call up different presets - note that not all plugins support this and some require initial configuration to respond to PC events.  
**Important note:** sending PC events prevents [Patch Persist](#) from working and can result in stuck notes depending on the plugin implementation. For those reasons, this feature is not recommended for normal use.
- **CC 0-127** - sends out CC (Continuous Controller) message with the value controlled by a widget.
- **Pitchbend** - sends a pitch-wheel message.
- **Aftertouch** - sends an aftertouch (also known as *channel pressure*) message.
- **MonoLowestOnly** - allows only the lowest note of a chord to pass through.
- **MonoHighestOnly** - allows only the highest note of a chord to pass through.
- **BlockNoteOn** - blocks incoming Note On messages.
- **BlockNoteOff** - blocks incoming Note Off messages.
- **BlockPitchbend** - blocks incoming Pitchbend messages.
- **BlockAftertouch** - blocks incoming aftertouch messages.
- **BlockPolytouch** - blocks incoming polytouch messages.
- **BlockAllOtherControllers** - blocks all CC messages not covered by listed parameters.
- **BlockModWheel** - blocks CC1 messages.
- **BlockVolume** - blocks CC7 messages.
- **BlockSustain** - blocks CC64 messages.
- **BlockExpression** - blocks CC11 messages.
- **BlockSysex** - blocks incoming SysEx messages (these messages are blocked by default for compatibility with Gig Performer 3).
- **NoteFollow** - toggles the NoteFollowValue Off/On.

- **NoteFollowValue** - when on, sets the note number to the most recently played note. Attach a widget to this parameter and then group a second widget attached to an arbitrary plugin parameter, which will then follow the note being played.
- **VelocityFollow** - toggles the VelocityFollowValue Off/On.
- **VelocityFollowValue** - when on, sets the velocity to whatever was last played. Use this in a manner similar to NoteFollow.
- **ReleaseFollow** - toggles the ReleaseFollowValue Off/On.
- **ReleaseFollowValue** - when on, sets the release value to whatever was last played. Not all keyboard controllers support Release values.
- **OSCPatchbay** - toggles Off/On the OSC Patch bay (must be enabled in [OSC options](#))
- **IgnoreGlobalTranspose** - enables you to not transpose incoming MIDI notes based on global transpose (useful when playing drums or effects samples).
- **TransposeUp** - increases the transpose by one step.
- **TransposeDown** - decrements the transpose by one step.
- **InvertKeyboard** - inverts the keyboard: higher incoming MIDI notes produce lower outgoing MIDI notes.
- **StopAfterChannelChange** - sends all notes off whenever you change a channel (to prevent stuck notes).

**Note:** the MIDI Controller plugin right-click menu is the same as other plugins, but with the addition of the **Change MIDI Input Device** sub-menu; this menu allows you to reassign the MIDI In or MIDI Out block to the MIDI controller chosen from the menu.

## 10.8 MIDI Outputs

This plugin is created only for connected MIDI controllers that offer one or more MIDI outputs. It provides a single MIDI input pin and is used to send MIDI messages to [external devices](#) such as synthesizers or samplers:

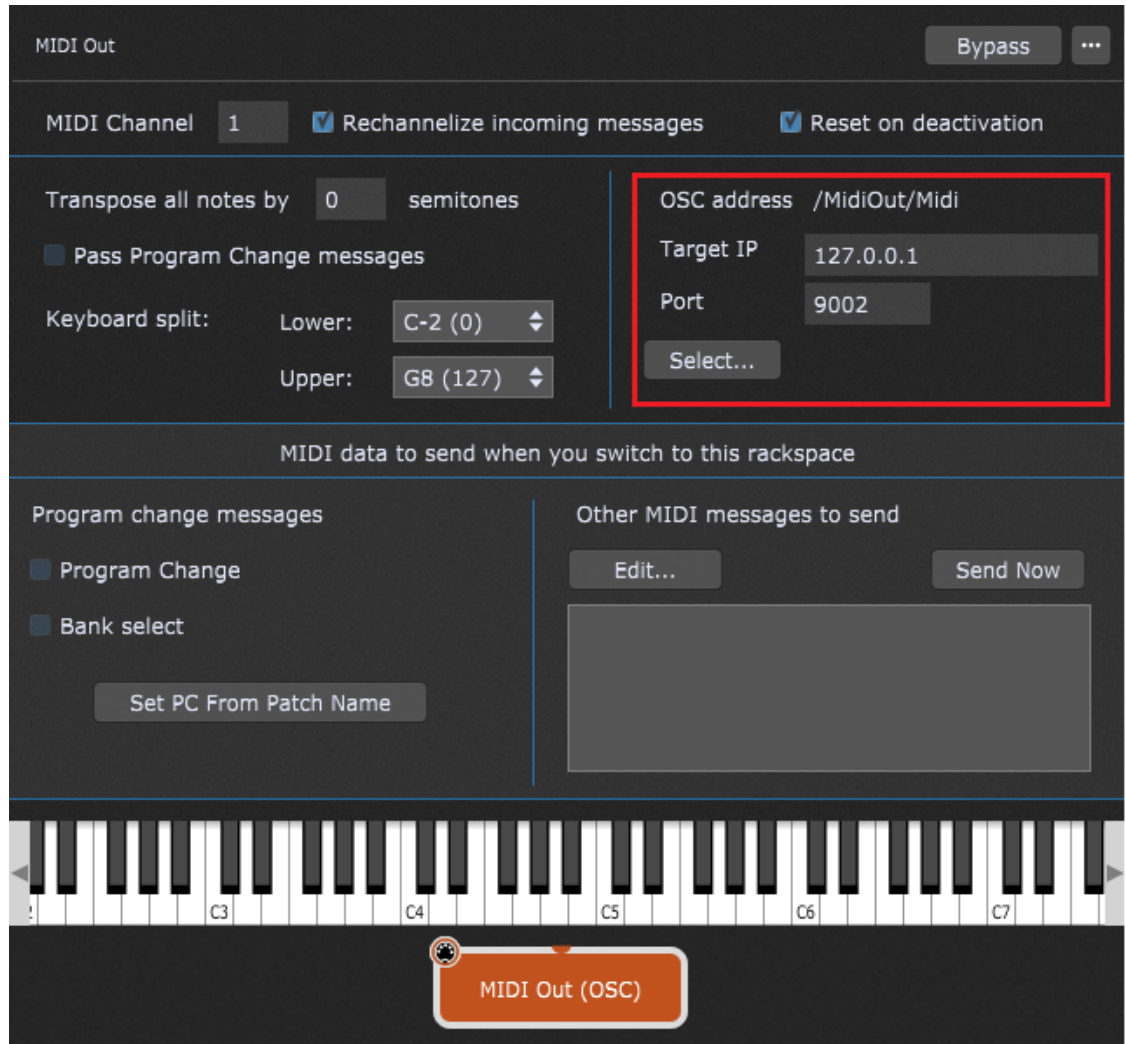


The MIDI Out plugin editors provide the following functionality:

1. **MIDI Channel** - sets the channel of the MIDI output.
2. **Rechanelize incoming messages** - turns on channel mapping so that incoming MIDI channel events have their channel changed to the specified channel. Note that program change transmission (described below) is still associated with the specified MIDI channel, even if this box is checked.
3. **Reset on deactivation** - stops all pending notes when deactivating (switching away from) this rackspace. Note: disabling this option can lead to stuck notes.
4. **Transpose all notes by** - allows you to alter the key transposition of the MIDI output.
5. **Pass Program Change messages** - if this option is checked, then incoming program change messages are passed through to external MIDI devices.

6. **Keyboard split** - allows you to set up keyboard splits for each MIDI Out block. Notes below the Lower value and/or above the Upper value are blocked; only notes between those values are allowed through. **Note:** you can use the virtual keyboard to create a keyboard split; hold the **Shift** key and click on the desired key on the virtual keyboard to set a low note. Hold the **Alt** key (**Option** key on Mac) and click on the desired key on the virtual keyboard to set an upper note.
7. **Program Change** and **Bank Select** - allow Gig Performer to automatically transmit a program change and/or Bank Select message every time a rackspace containing the MIDI Out block is activated so that each of your external MIDI devices (keyboard and rack-mount synths, drum machines, etc.) load with the correct sounds and are instantly ready for use.
8. **Set PC From Patch Name** - rather than just using PC numbers, clicking on this button allows you to select the PC number to be sent out using patch names defined in instrument definition file (INS file) if available for your hardware synth when switching to the current rackspace.
9. **Edit...** - clicking on this button opens the [MIDI Helper Tool](#), which allows you to send arbitrary sequences of MIDI messages (including SysEx) when you switch to this rackspace. Basically, this is the same mechanism that is in the [Song Part Properties](#) dialog that allows you to send an arbitrary sequence of MIDI messages when switching to a song part; in the MIDI Out block the MIDI Helper Tool won't let you change the output device but otherwise works the same way.
10. **Send Now** - clicking on this button allows you to send the selected MIDI messages to this MIDI out device.

The **MIDI Out (OSC)** block includes additional options, as indicated with the red rectangle:

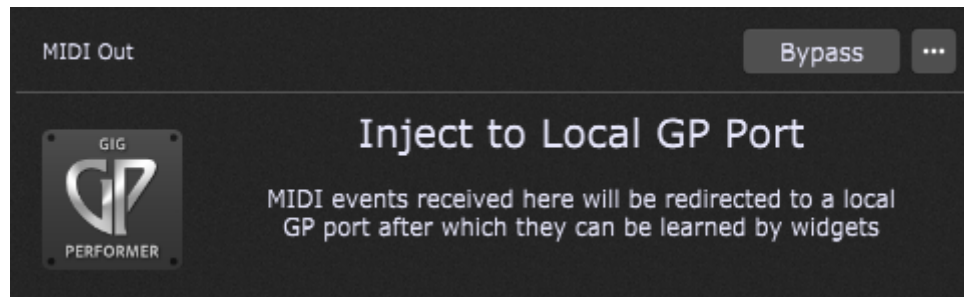


1. **OSC address** - allows you to enter the OSC address for this block.
2. **Target IP** - address of the remote device to which you want to send the message.
3. **Port** - port number of the remote device to which you want to send the message.
4. **Select...** - clicking this button allows you to easily select an OSC target from the list that must be previously set in [OSC Options](#).

The MIDI Out (OSC) block is mainly intended to send OSC-wrapped MIDI messages from Gig Performer to other applications that can respond to such OSC messages. While you can send MIDI from and to the Global rackspace using the Global blocks, in [this blog article](#) you can see an example of how to use MIDI OSC blocks to send MIDI to the [Global rackspace](#).

**Note:** if you use the MIDI Helper Tool in a MIDI Out (OSC) block, the selected MIDI messages will be sent out over OSC. The output device will display OSC rather than MIDI Out.

There is also experimental support for a **MIDI Out (Local GP Port)**. This block is designed to receive messages from plugins such as the [MIDI File Player](#), [Envelope Follower](#), or other third-party plugins such as arpeggiators so that they can be mapped to widgets.



**Important:** do not connect any MIDI In (OMNI) blocks to this MIDI Out block to avoid creating a MIDI feedback loop.

The MIDI Out (Local GP Port) does not have any parameters.

### MIDI Out parameters

The following parameters for the MIDI Out plugin are available:

- **ChannelNumber** - allows you to alter the MIDI output channel number (1-16).
- **MinNote** - allows you to alter the Lower note setting for the keyboard split.
- **MaxNote** - allows you to alter the Upper note setting for the keyboard split.
- **Transpose** - allows you to alter the key transposition. Caution: Do not change this setting while holding notes; stuck notes may result. If you do get a stuck note, don't panic - just click on the Panic icon to turn all notes off.
- **CC 0-127** - allows you to transmit the specified CC (control change) message as the widget is moved.
- **Force MIDI Channel** - causes all MIDI events being transmitted to be mapped to a specific MIDI channel.
- **PC** - allows widget movements to send MIDI program change messages to external MIDI tone generators in order to call up different patches.

- **Pass through incoming PC messages** - toggles the *Pass Program Change messages* setting in the MIDI Out plugin editor.
- **Reset On Switch** - allows stopping all pending notes when switching away from this rackspace.

Refer to this blog article [How to send program change messages out to hardware devices and other applications](#) for in-depth guidelines and examples.

## 10.9 MIDI Processing and Monitoring

In this chapter, find out more about internal plugins that help you monitor, block and redirect MIDI messages. Check this blog article [The most flexible MIDI processing](#) to learn more about various ways to process MIDI in Gig Performer.

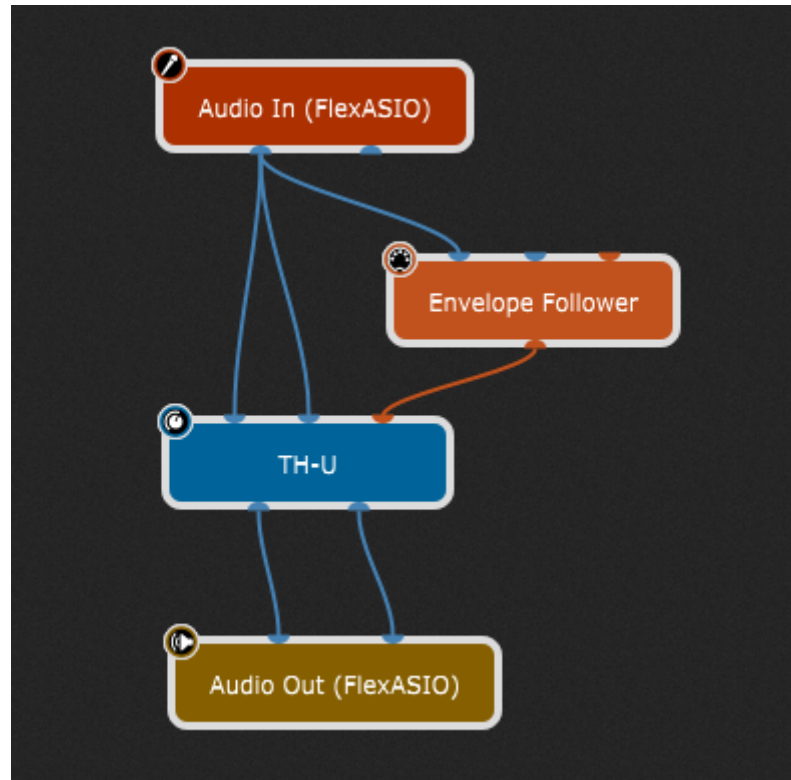
### 10.9.1 Envelope Follower

The Envelope Follower is a plugin that analyzes an incoming audio signal and produces MIDI CC messages with values proportional to the magnitude of that signal.

Uses for electric guitar are, for example, signal-dependent Wah-wah or signal-dependent high cut-off filtering, which would produce almost violin-like effects. Another valuable use is for ducking audio while one is speaking. Read [this blog article](#) to see examples.

The produced MIDI CC messages are sent out through the MIDI output pin of the plugin, which needs to be connected to the MIDI input pin of another plugin, such as an effect plugin.





You can, for example, add an equalizer plugin and let the MIDI CC values control the central frequency of a bandpass filter to create a Wah-wah-like effect.

Double-clicking the Envelope Follower block opens its plugin editor window.



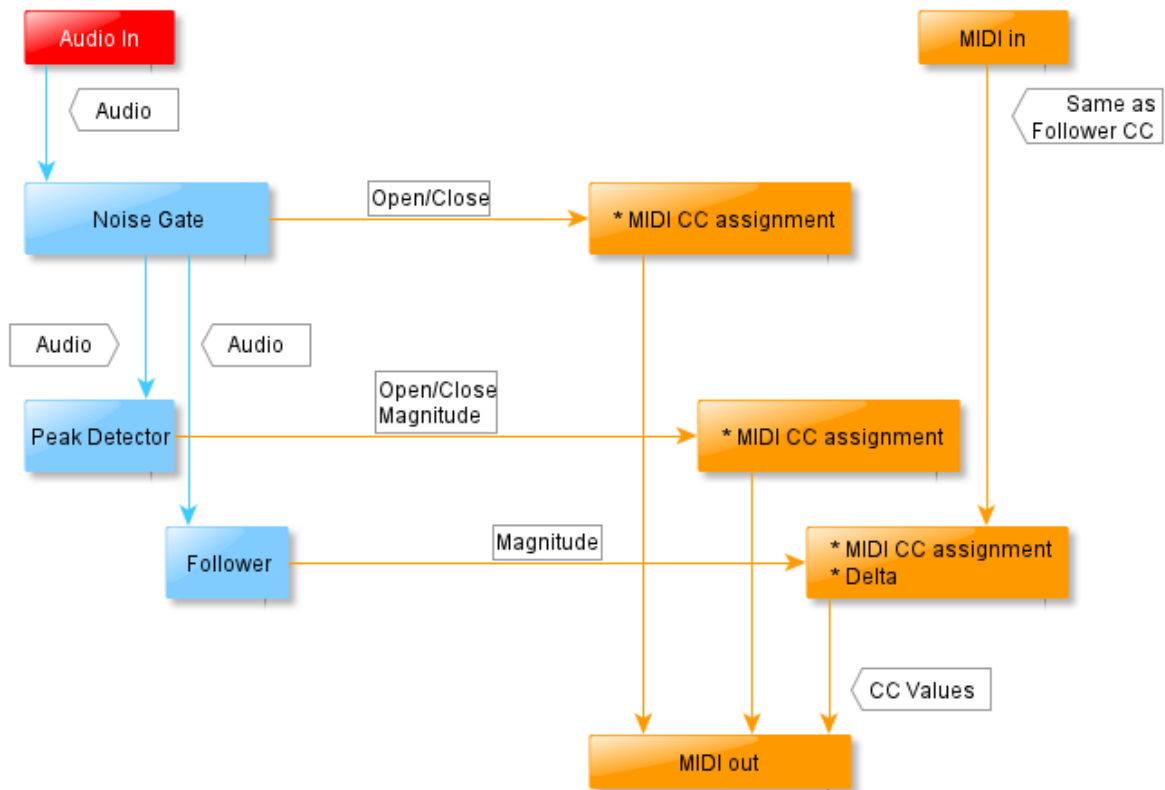
### Features:

- A Noise Gate **(A)** to prevent processing of signals below a certain magnitude. Opening and closing of the gate is reported using MIDI CC messages.
- A Peak Detector **(B)** to detect if the signals rise above a certain threshold. The results depend on the processing type and are reported using MIDI CC messages.
- The envelope follower main controls **(C)**. This component continuously reports the magnitude of the incoming signals. The rising and falling speed of the reported values are independently configurable.
- An option to have an incoming MIDI CC value 'augmented' by the Envelope Follower output **(D)**.
- The MIDI CC numbers are configurable for all three components: Noise Gate, Peak Detector and Follower.
- Outgoing MIDI messages always use channel 1 for the left audio signal and channel 2 for the right audio signal. This applies to all three components.
- The MIDI CC number of the incoming MIDI messages (if used) must be the same as the MIDI CC number used by the follower. MIDI channel 1 is applied to the output of

the left channel and MIDI channel 2 is applied to the output of the right channel. All other messages are discarded.

## Signal path

The signal path through this plugin is described in the following diagram:



The first stage of the plugin is the **Noise Gate** (A). You can set the threshold at which the gate opens and the amount by which the signal has to drop below this threshold to close the gate. The noise gate processes the left and right channels in a 'linked' fashion: If one of the channels crosses the threshold, the gate opens for **both** channels. The gate stays open for both channels until both channels drop below the threshold (minus a delta). This delta makes sure that, when the signal level is just about equal to the threshold, the noise gate really opens or closes instead of going back and forth between those two states (flapping). Please note that the MIDI CC messages for the noise gate are sent for both audio channels (they always come in pairs).

The next stages are the **Peak Detector** (B) and the **Follower** (C). These two stages act independently of each other.

The **Peak Detector** takes the audio signal, and determines if it is above the Threshold (4) or drops below the threshold minus the Threshold Delta. The Peak Processing Control controls the output of the Peak Detector.

The signal of both channels is processed independently.

Except for the Open/Close Peak Processing mode, it is possible to suppress the reporting of CC values of zero, when the peak detector closes.

The **Follower** is the main part of the plugin. The audio signals are analyzed in the following way: First the audio signal is averaged to get rid of very fast fluctuations. The resulting magnitude is then processed by the attack/decay filters. This allows you to have the follower output react either quickly or slowly to audio signal attacks and/or decays. For example, if you set the attack slope to 2,000 ms, it will take 2 seconds to go from 0 to 127 in the output (if the input after gain is near 0 dB). When you use a guitar, you will notice that you will have to play rather loudly for quite some time. However, if you set both the attack and decay to 40 ms, the output will go from 0 to 127 on almost every stroke.

The blue arrows indicate the audio signal path, while the MIDI CC messages follow the orange arrows. The MIDI CC messages for the left audio channel are always reported on MIDI channel 1 and for the right audio channel on MIDI channel 2.

The last stage is **MIDI Output (D)**. Here the MIDI CC numbers for the three components can be configured, as well as the frequency at which changes are reported and one special option for the follower (minimum delta).

## Controls and indicators

The plugin has the following controls and indicators:

### Noise Gate

- **Threshold (1)**. This sets the audio signal level at which the gate should open. The noise gate evaluates the signal level of both audio channels and if one or more of the signals of the audio channels exceeds the threshold, both channels are opened.
- **Threshold Delta (2)**. This sets the amount the audio signal of both channels must drop below the threshold to close the gate. So, when the threshold is set at -60 dB, and the hysteresis is at 5 dB, the level of both audio channels must fall below -65 dB before the gate closes.

- Indicator (3) reports whether the gate has opened (bright red) or not (dark red).

### Peak Detector

- Threshold (4). This sets the audio signal level at which a peak message will be reported. The peak detector independently evaluates the signal level of the audio channels and reports for each audio channel when the signal exceeds the threshold.
- Threshold Delta (5). This sets the amount of audio signal that both channels must drop below the threshold to report a 'no peak' message. So, when the threshold is set at -12 dB and the hysteresis is at 5 dB, the level of an audio channel must fall below -17 dB before processing stops for a particular channel (possibly also sending a MIDI CC value, depending on the Suppress zero values (8) discussed below).
- Peak Scaling (6). This sets the value by which the MIDI CC values are multiplied. It allows the user to increase or decrease the outputted CC values. For example, when this control is set to 0.5 and internally the determined value is 150 (which is higher than the allowed maximum value of 127 for MIDI CC values), the outputted value becomes  $150 * 0.5 \rightarrow 75$ . If this control had been set to 1.0, the outputted value would have been 127, because the plugin keeps the CC values within the allowed range of 0 .. 127. Note that the calculation takes place before the minimum value (0) and maximum value (127) is applied
- Peak Processing (7). This control determines what kind of output is sent. Roughly this output falls into two categories: Simple On/Off (127/0) or the magnitude:
  - *Open/Close*: The output is 127 if the channel input is greater than the threshold or 0 if it drops below the threshold minus delta.
  - *Oneshot*: The output is the magnitude of the audio signal (mapped to a range of 0 – 127) multiplied by the Peak Scaling factor (6). This only occurs once until the signal drops below the threshold minus delta. Then it will be 'rearmed'.
  - *Continuous*: The output is the magnitude of the audio signal (mapped to a range of 0 – 127) multiplied by the Peak Scaling factor (6). All magnitude changes will be sent, until the signal drops below the threshold minus delta.
  - *Continuous Rising*: The output is the magnitude of the audio signal (mapped to a range of 0 – 127) multiplied by the Peak Scaling factor (6). Magnitudes greater than the previously reported magnitude will be sent until the signal drops below the threshold minus delta.

- Suppress Zero Values (8). This control simply prevents the peak detector from sending MIDI CC values of 0 when the magnitude of the input signal falls below the threshold minus delta. Suppressing zeros is not available for the Open/Close processing mode.
- Indicator (9) reports whether a peak level is currently being detected for one or more audio channels (bright red) or not (dark red).
- Indicators (22) report the current output values.

### Follower

- Smoothing Time (10). This sets the size of the window of the first stage of the follower. Please note that the size of this window adds latency to the speed at which the follower responds to attack and decay slopes of the follower MIDI CC messages. Most of the time 20 ms will do nicely.
- Level Adjustment (11). This control increases the level of the signal after the averaging filter before entering the attack and decay filters. You should adjust the gain to have the Level Indicators (12) reach the green level. This way the timings provided by the attack and decay controls are more accurate. Yellow means the level is set too high, red means the level is set too low.
- The next two plugin controls are the attack and decay slope controls. These are the most important controls, because they determine by large the character of the eventually resulting sound, for example a Wah-wah-like effect using an attack and decay somewhere between 40 to 200 ms combined with a steep band-pass filter or a signal-dependent VCF effect using longer times (> 500 ms) combined with a low-pass filter. The timings indicated by these controls are just that: indications. The true timings are strongly dependent on the actual audio input signal levels. These settings depend a lot on the type of audio signal: percussive sounds might benefit from shorter times, while padding-like sounds can be used with longer times. Ultimately, your ears will decide what works best for the effect you want to accomplish.
  - *Attack* (13). This control sets the speed in milliseconds at which the value of the MIDI CC messages produced by the follower should increase from 0 to 127. For example, when this control is set to 1,500 ms and the gain is adjusted the right way, when the input signal level is about 0 dB it will take 1.5 seconds to have the value of

the outputted MIDI messages go from 0 to 127 (provided that this value is 0 at the start of the measurement).

- *Decay* (14). This control sets the speed in milliseconds at which the value of the MIDI CC messages produced by the follower should drop from 127 to 0. For example, when this control is set to 3,000 ms and the gain is adjusted the right way, when the input signal level is about infinite dB it will take 3.0 seconds to have the value of the outputted MIDI messages drop from 127 to 0 (provided that this value is 127 at the start of the measurement).
- Follower Scaling (15). This sets the value by which the MIDI CC values are multiplied. It allows the user to increase or decrease the outputted CC values. For example, when this control is set to 0.5 and internally the determined value is 150 (which is above the allowed maximum of 127 for MIDI CC values), the output value becomes  $150 * 0.5 \rightarrow 75$ . If this control had been set to 1.0, the output value would have been 127, because the plugin keeps the CC values within the allowed range of 0 .. 127. Note that the calculation takes place before the minimum value (0) and maximum value (127) are applied.
- MIDI-In processing (16). This control allows you to have an incoming MIDI CC value added to the output.

Only incoming MIDI messages with a CC number that is the same as that of the follower (see MIDI output below for more details) are processed.

MIDI channel 1 values apply to the left audio channel, whereas MIDI channel 2 values apply to the right audio channel.

There are three types of processing:

- *Ignore MIDI-in*. The incoming MIDI CC messages are ignored.
- *Adding 0..127*. This adds the incoming MIDI CC value to the output after applying the Follower Scaling.
- *Adding -64..0..63*. The incoming MIDI CC values in the range of 0..127 are translated to -64..63. So, an incoming MIDI CC value of 0 results in subtracting -64 of the output and 127 actually adds 63. A MIDI CC value of 64 does not add or subtract anything.
- Indicators (23) report the current output values.

## MIDI Output

- The next three controls determine the MIDI CC number that is used for the various parts of the plugin. As said before, the MIDI channel of the messages always depends on the audio channel: MIDI channel 1 represents the left audio channel, whereas MIDI channel 2 represents the right audio channel. You can assign the same CC number to all messages, but then there is no way to distinguish the messages of the three components, so it is strongly advised that you don't do that.

- *Noise Gate CC (17)*. Sets the CC number of the noise gate reports.

- *Follower CC (18)*. Sets the CC number of the follower reports.

- *Peak CC (20)*. Sets the CC number of the peak detector reports.

Setting a value to 'Off' causes no reports to be sent for that particular component (Noise Gate/Follower/Peak Detector).

- *Follower Minimum Delta (19)*. Setting this value to 2 or higher sets the interval of the MIDI CC values of the follower part. For example, if the most recently outputted MIDI CC value is 12 and this control is set to 4, then the MIDI CC value has to rise to 16 or higher or drop to 8 or lower, before a new report is emitted.

Setting the value to 'Off' causes the normal behavior: All changes are reported.

- *Reporting Frequency (21)*. This control sets the number of times per second the plugin should check for changes to report. Note that there will only be a report of this change if there is a change to a particular MIDI CC value. If the rate is set very low (for example, 2 checks per second), the output will become rather coarse with large jumps. Higher settings result in a much smoother response. Typically, somewhere between 15 – 30 will do nicely, but you can crank it up to 50 checks per second.

## How to use an Envelope Follower

Here is an example of how to use this plugin:

1. Add the Envelope Follower plugin and route the input signal from an electric guitar (say) to the plugin's left audio input.
2. Add an equalizer plugin (for example, the MEqualizer plugin from MeldaProduction).



3. Route the MIDI output of the Envelope Follower plugin to the MIDI input of the equalizer plugin.
4. Modify the equalizer so that MIDI CC number 7 controls the central frequency of a band pass filter.
5. Connect the equalizer plugin somewhere between the signal inputs and signal outputs.
6. Play the instrument and adjust the Level Adjustment (11) of the plugin. If the level indicator (12) reaches the green level (second 'led' from the top) rather often, then the applied gain is right.
7. Adjust the attack (13) and decay (14) controls to your liking.
8. Have fun.

To see examples of how to use this plugin, check out [our blog](#) and [this community thread](#).

## Notes

- You can resize the plugin editor window by grabbing and dragging the lower-right corner.
- The values of the controls can also be directly modified by clicking on the value.
- The controls without a knob can also be modified by dragging the mouse in the value box. You can slow down the speed at which values change by pressing the **Ctrl** key (or the **Cmd** key on a Mac) while you drag the mouse.

## Envelope Follower parameters

The following parameters are available for the Envelope follower plugin:

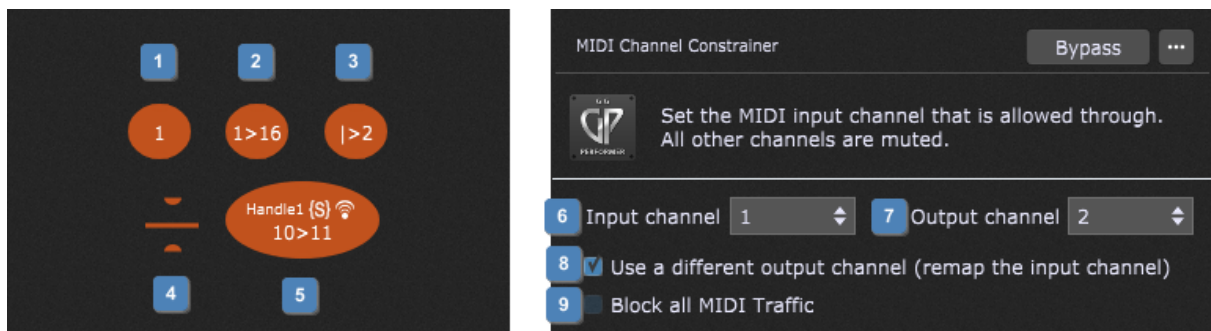
- **Reporting Frequency** - adjusts the reporting frequency in the MIDI Output section, from 1 Hz to 50 Hz.
- **Follower Level Adjustment** - adjusts the level adjustment in the Follower section, 0 dB to 30 dB.
- **Follower Attack** - adjusts the attack in the Follower section, from 10 ms to 3000 ms.

- **Follower Decay** - adjusts the decay in the Follower section, from 10 ms to 3000 ms.
- **Follower Smoothing Time** - adjusts the smoothing time in the Follower section, from 20 ms to 100 ms.
- **Follower Scaling** - adjusts the follower scaling in the Follower section, from 0.10 to 3.00.
- **MIDI-In processing** - sets the type of the MIDI-In processing: Ignore MIDI-in, Adding 0..127, or Adding -64..0..63.
- **Peak Threshold** - adjusts the threshold in the Peak Detector section, from -100 dB to 12 dB.
- **Peak Threshold Delta** - adjusts the threshold delta in the Peak Detector section, from 3 dB to 30 dB.
- **Suppress Peak Off Reports** - controls the Suppress Zero Values toggle button in the Peak Detector section (Off/On).
- **Peak Detector Scaling** - adjusts the peak scaling in the Peak Detector section, from 0.10 to 3.00.
- **Peak Processing** - sets the type of the Peak Processing: Open/Close, Oneshot, Continuous, Continuous Rising.
- **Noise Gate Threshold** - adjusts the threshold in the Noise Gate section, from -100 dB to 0 dB.
- **Noise Gate Threshold Delta** - adjusts the threshold delta in the Noise Gate section, from 3 dB to 30 dB.
- **Noise Gate CC** - adjusts the noise gate CC value in the MIDI Output section, from Off to 127.
- **Peak Detector CC** - adjusts the Peak CC value in the MIDI Output section, from Off to 127.
- **Follower CC** - adjusts the follower CC value in the MIDI Output section, from Off to 127.
- **Follower Minimum Delta** - adjusts the follower minimum delta value in the MIDI Output section, from Off to 127.

- **Audio Level Channel 0** - shows the incoming audio level, channel 0.
- **Audio Level Channel 1** - shows the incoming audio level, channel 1.
- **Follower Level Channel 0** - shows the incoming audio level altered by the Follower section, channel 0.
- **Follower Level Channel 1** - shows the incoming audio level altered by the Follower section, channel 1.
- **Peak State Overall** - shows the Threshold led in the Peak Detector section (Off/On).
- **Peak Level Channel 0** - shows the incoming audio level altered by the Peak Detector section, channel 0.
- **Peak Level Channel 1** - shows the incoming audio level altered by the Peak Detector section, channel 1.
- **Noise Gate State** - shows the state of the noise gate (Closed/Open).

### 10.9.2 MIDI Channel Constrainer

The **MIDI Channel Constrainer** is the only plugin that isn't shaped rectangularly; this plugin provides a single MIDI input pin and a single MIDI output pin:



The MIDI Channel Constrainer allows you to set the MIDI input channel that is allowed through, remap the input channel, or block all MIDI traffic that goes through this plugin.

1. **Input channel is the same as the output channel** - this is the default setting for all MIDI Channel Constrainer plugins and this is how they look like when you insert them onto the design surface.
2. **Input channel is remapped** -  $X > Y$  (in this case  $1 > 16$ ) indicates that option (8) is activated and the input channel is remapped to a different output channel.

3. **OMNI input channel is remapped** - "|" indicates that OMNI input channel is selected; it must be remapped to the desired output channel.
4. **MIDI traffic is blocked** - selecting the option **(9)** in the plugin's window causes this plugin to change its visual representation.
5. **Elliptic shape** - when OSC and/or GScript are activated for this block and with a defined handle name, this plugin adjusts its shape to include that information.
6. **Input channel** - this plugin provides 16 input channels and a OMNI input. By default, the input channel is the same as the output channel. This channel is allowed through and all other channels are muted.
7. **Output channel** - ticking the option **(8)** allows you to remap the input to the specified output channel (1 - 16).
8. **Use a different output channel** - tick this check box if you want to remap the input to a different output channel.
9. **Block all MIDI traffic** - when selected, all incoming MIDI events to this plugin will be blocked.

#### MIDI Channel Constrainer parameters

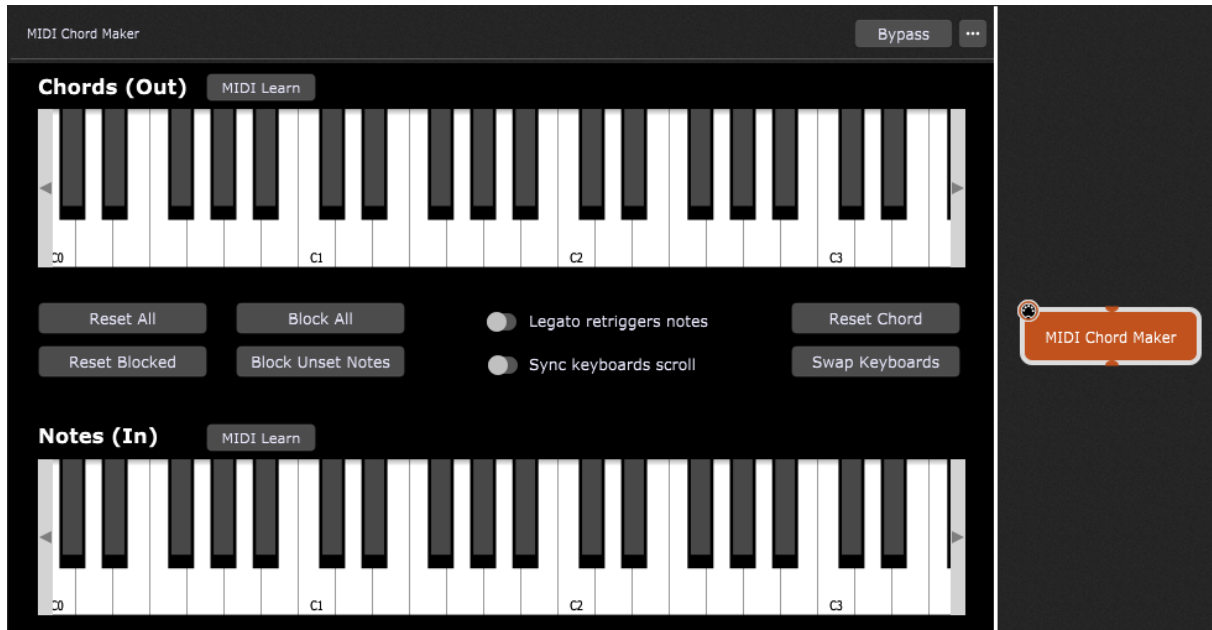
The following parameters are available for this plugin:

- **InputChannel** - allows you to change the input channel (1-16).
- **OutputChannel** - allows you to change the output channel (1-16).
- **BlockAll** - allows you to pass through or block all MIDI traffic.
- **RemapChannel** - toggles remapping the input channel.
- **OmniMode** - toggles the OMNI mode for the input channel.

Check out [this blog article](#) to learn more.

### 10.9.3 MIDI Chord Maker

The **MIDI Chord Maker** plugin allows you to map notes to chords and block unwanted notes. This plugin provides a single MIDI input pin and a single MIDI output pin:

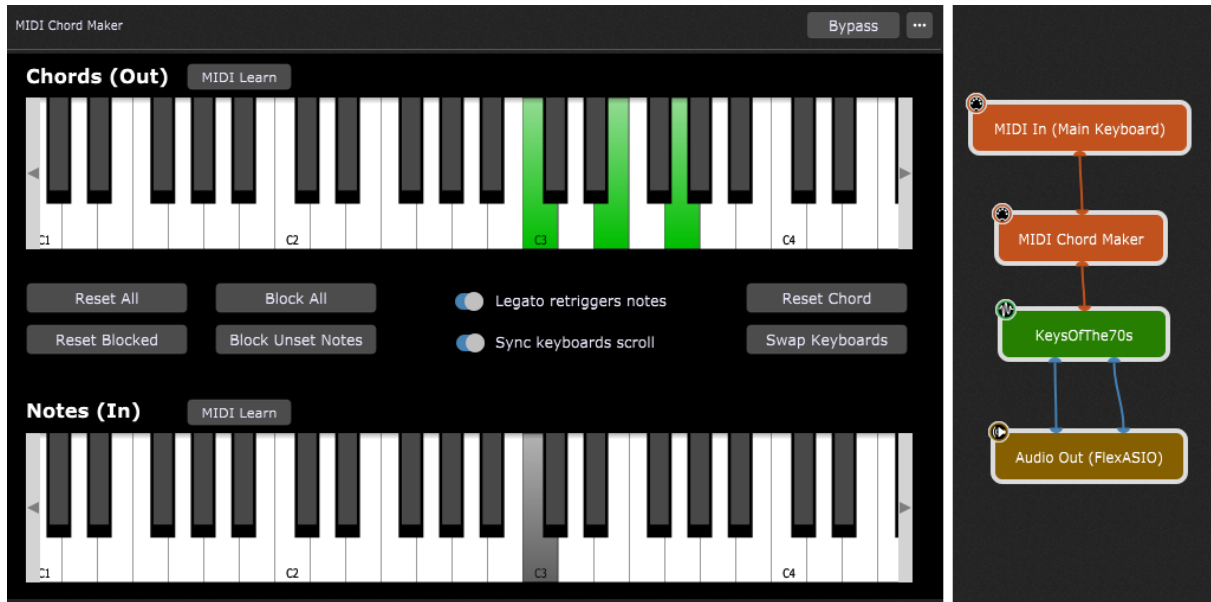


To map a note to a chord, follow these steps:

1. Click on the left or right arrows on the keyboard to scroll the keyboard view and locate the desired key. Click on the desired note in the Notes (In) section, for example, C3. The key will start flashing.

Alternatively, you can connect the MIDI output pin of your MIDI controller, connect it to the MIDI input pin of the MIDI Chord Maker, click on the MIDI Learn button in the Notes (In) section and then play a key. The played key will start flashing.

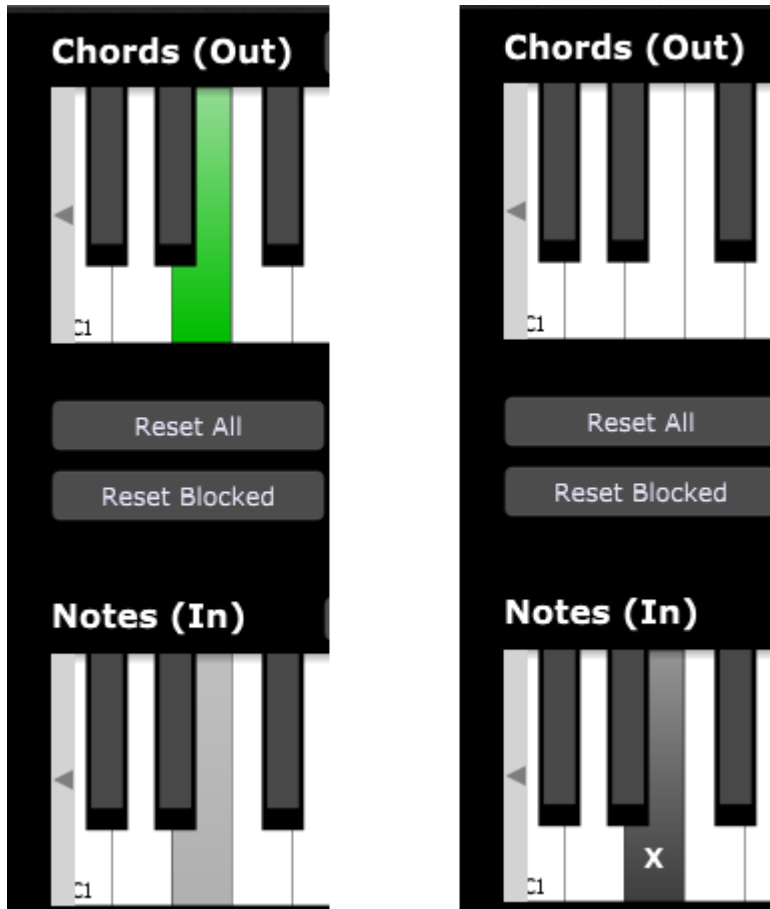
2. While the selected key is flashing in the Notes (In) section, click on the desired notes in the Chords (Out) section. Every clicked note in a chord will be colored green.



Needless to say, you can click on the MIDI Learn button in the Chords (Out) section and then play a chord on your MIDI controller. If the MIDI Chord Maker is connected to a synth plugin (such as KeysOfThe70s in the screenshot above), and that synth plugin is connected to the outputs of your audio interface, you can hear the played chord.

All notes in the Notes (In) section mapped to chords will be displayed in gray color.

To block a note, click on the desired note in the Notes (In) section and then click on the corresponding green key in the Chords (Out) section to deselect it.



The blocked key will be overlaid with the 'X' symbol.

Other elements in the MIDI Chord Maker plugin editor window are:

**Reset All** - clicking this button reverts every key to its initial (default) state. All note assignments, and all previously blocked notes are now passed through without change.

**Note:** If you accidentally click on this button, or make any unwanted change in this window, you can undo that change using **Ctrl+Z** (or **Cmd+Z** on a Mac).

**Reset Blocked** - clicking this button resets all blocked notes, so all blocked notes are passed through.

**Block All** - clicking this button blocks all notes.

**Block Unset Notes** - clicking this button blocks all notes except for the notes that have been mapped to chords.

**Legato retriggers notes** - turn off this toggle button if you want this behavior: if a note in the new chord is the same as the one in the previous chord, playing the new chord before you released the first one will not retrigger that note a second time (it will just stay on).

**Sync keyboards scroll** - turn off this toggle button if you want to scroll keyboards independently of each other.

**Reset Chord** - clicking this button resets the chord for the currently selected note (the note that is currently flashing).

**Swap Keyboards** - clicking this button swaps the position of the Notes (In) and the Chords (Out) keyboards.

#### **MIDI Chord Maker parameters**

Apart from the common parameters **Bypass Plugin** and **Open/Close Plugin Editor**, the MIDI Chord Maker plugin provides no other parameters.

#### **10.9.4 MIDI Filter**

The **MIDI Filter** plugin can be used either to block specific MIDI messages or to remap them to other messages. Some messages can only be blocked (MIDI note messages and Active Sense). This plugin provides a single MIDI input pin and a single MIDI output pin:





Double-clicking this plugin opens its plugin editor:

1. **Passthru MIDI message** - indicates that this MIDI message is passed through this plugin.
2. **Blocked MIDI message** - indicates that this MIDI message is blocked.
3. **Remapped MIDI message** - indicates that this MIDI message is remapped (converted into messages of another type).
4. **Contextual menu** - clicking on any line in the MIDI Filter plugin editor opens a contextual menu where you can make one of four choices: Passthru, Block, Map to... or Cancel. Hovering over the Map to... item expands the menu item, allowing you to select the desired MIDI message.
5. **Block All** - blocks all MIDI messages. If you enable this option, everything, including Notes, will be blocked. This can be very useful if you use multiple variations and want to allow MIDI messages to go to some plugins but not others, depending on which variation is selected. You should, however, exercise caution when doing this in real-time as you could get left with a hanging note if you block notes before the

Note Off message is sent out. If this occurs, don't panic - just click on the Panic icon in the main toolbar to turn all notes off.

6. **Allow All** - allows all MIDI messages. By default, all MIDI messages are allowed.

This plugin is typically inserted between a MIDI In plugin and a sound-generating plugin, where common usages include blocking sustain pedal messages, converting modulation wheel messages into volume messages, and mapping aftertouch messages to mod wheel messages.

### MIDI Filter parameters

The following parameters for the MIDI Filter plugin are available:

- **Pitchbend** - allows you to control Pitchbend messages (passthru, block or remap).
- **Aftertouch** - allows you to control Aftertouch messages (passthru, block or remap).
- **Polyphonic Key Aftertouch** - allows you to control Polyphonic Key Aftertouch messages (passthru, block or remap).
- **Program Change** - allows you to control Program Change messages (passthru, block or remap).
- **Notes** - allows you to control Notes messages (passthru or block only).
- **CC# (0-127)** - allows you to control Control Change messages (passthru, block or remap).
- **Note On** - allows you to control Note On messages (passthru or block only).
- **Note Off** - allows you to control Note Off messages (passthru or block only).
- **Active Sense** - allows you to control Active Sense messages (passthru or block only).

MIDI Filter plugin parameters can be widget-controlled. When using on-off widgets (i.e. switches or LED buttons), the selected parameter will be passthru when in the OFF position, and blocked when in the ON position.

The use of continuous widgets (knobs and sliders) allows you to scroll through all remapping options in the in-between settings (the minimum setting will pass through the selected message, while the maximum setting will block the message).

**Tip:** refer to [this chapter](#) to learn how you can remap MIDI messages without using the MIDI Filter plugin.

### 10.9.5 MIDI Monitor

This plugin provides a single MIDI input pin and a single MIDI output pin. It displays MIDI messages from the plugin block which is connected to it and it is intended primarily for diagnostic purposes:



Optionally, turn on the **SysEx indices** toggle button to display the position of each byte of a SysEx message (this feature is intended for advanced users as a helper when writing GPCScript functions that manipulate SysEx messages).

Clicking the **Copy** button copies all the contents of this window to the clipboard. To clear the window, click on the **Clear** button.

You can set the colors for different MIDI messages in the [Display Options](#) window.

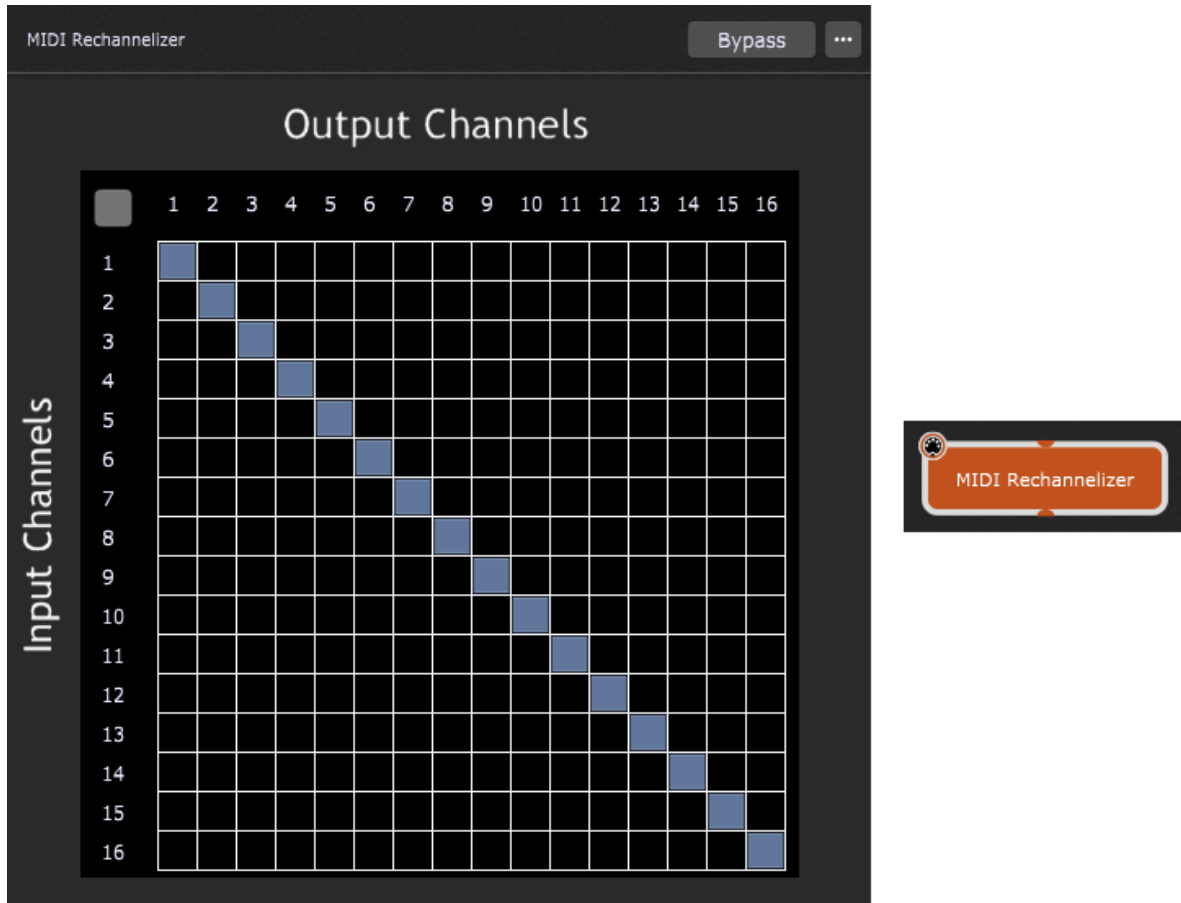
#### MIDI Monitor parameters

Apart from the common parameters **Bypass Plugin** and **Open/Close Plugin Editor**, the MIDI Monitor plugin provides no other parameters.

**Note:** If you want to monitor all incoming MIDI messages, see the [Global MIDI Monitor](#) chapter.

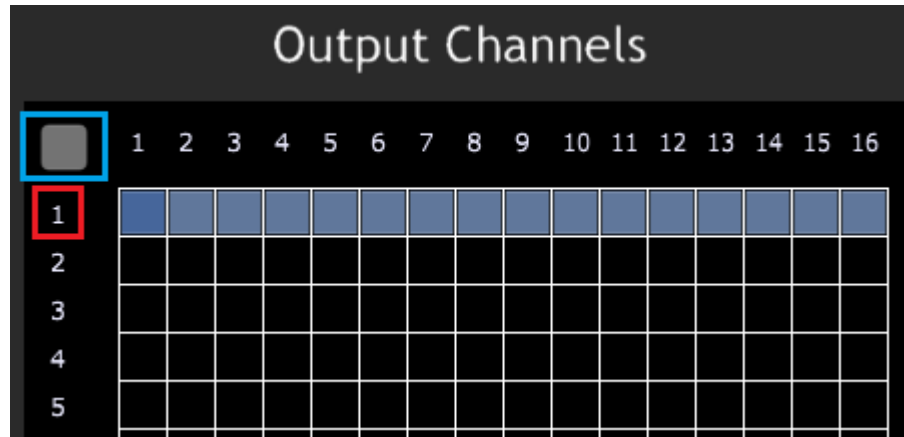
## 10.9.6 MIDI Rechannelizer

The **MIDI Rechannelizer** allows you to easily map MIDI input channels to MIDI output channels using a grid-based system. This plugin provides a single MIDI input pin and a single MIDI output pin:



All MIDI input channels are mapped to their corresponding MIDI output channels by default.

To add another MIDI mapping, simply click on the desired field in the grid. You can also click on the row or column number to toggle the whole row or column respectively (indicated with the red rectangle):



The **Reset to default** button (indicated with the blue rectangle) allows you to reset all the mappings to their default state (as shown in the first screenshot). Clicking this button while holding down the **Shift** key allows you to clear all the mappings and start with a clean slate.

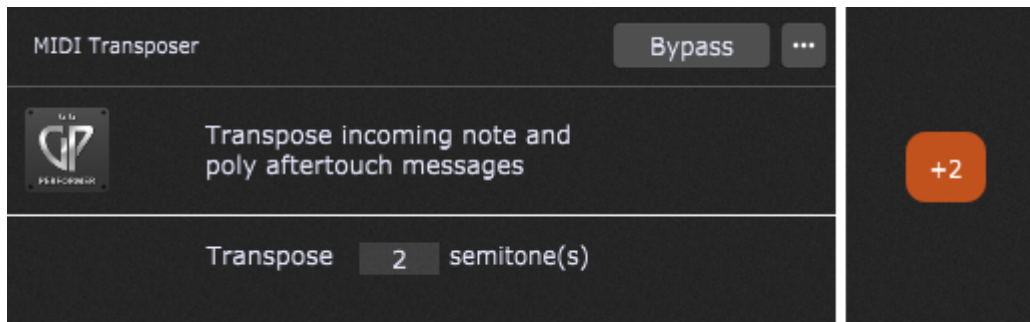
**Note:** If you accidentally make any unwanted changes in this window, you can undo those changes using **Ctrl+Z** (or **Cmd+Z** on a Mac).

#### **MIDI Rechannelizer parameters**

This plugin includes 256 parameters, which include mappings from **1-1** to **16-16**. They are Off/On type so assign these parameters to a button widget.

### **10.9.7 MIDI Transposer**

The **MIDI Transposer** is a simple plugin that allows you to transpose incoming note and poly aftertouch messages. This plugin provides a single MIDI input pin and a single MIDI output pin:



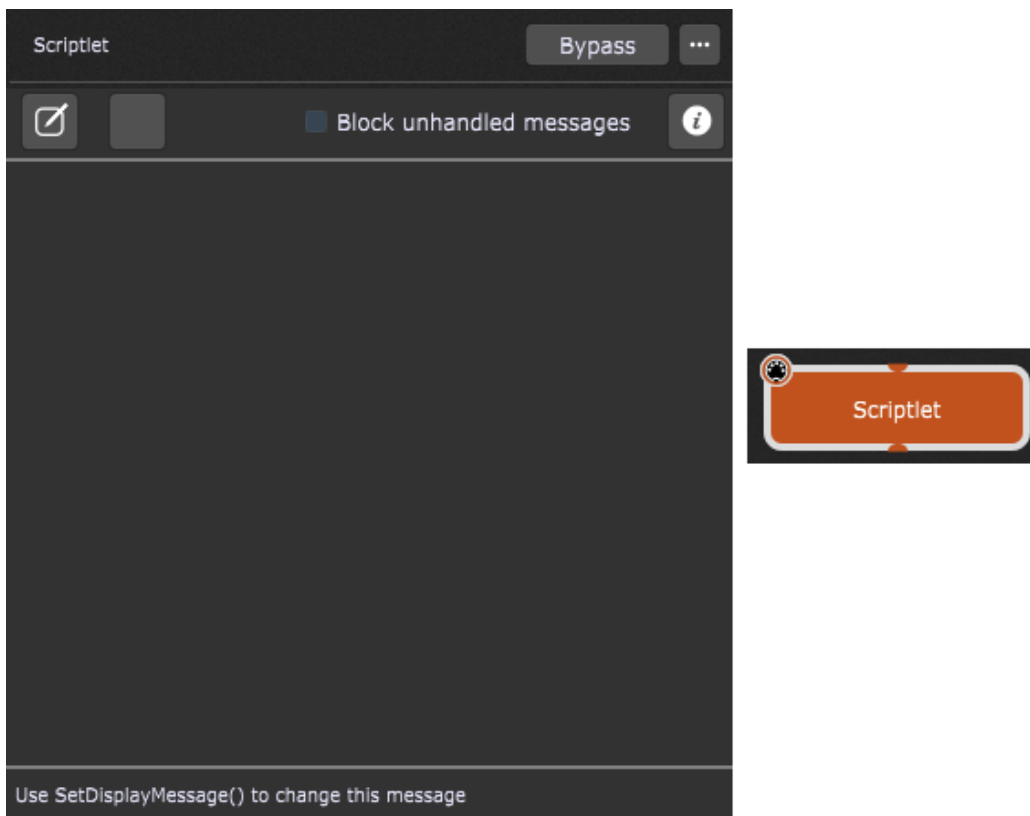
You can transpose from -127 to +127 semitones.

### MIDI Transposer parameters

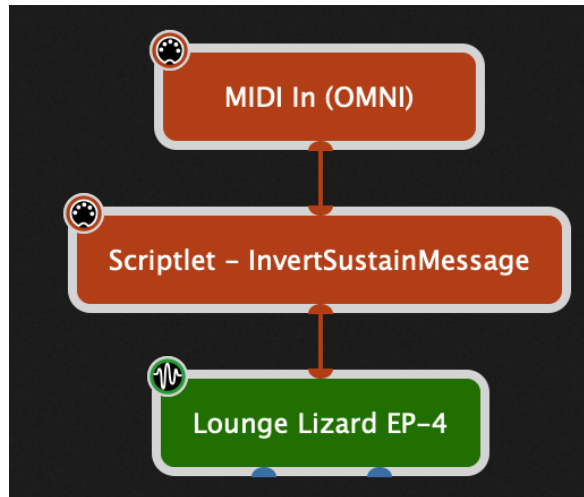
This plugin exposes one parameter **Transpose Amount**, and can go from -127 to +127 semitones.

### 10.9.8 Scriptlet

The **Scriptlet** plugin allows you to create your own MIDI processor plugins. This plugin provides a single MIDI input pin and a single MIDI output pin:

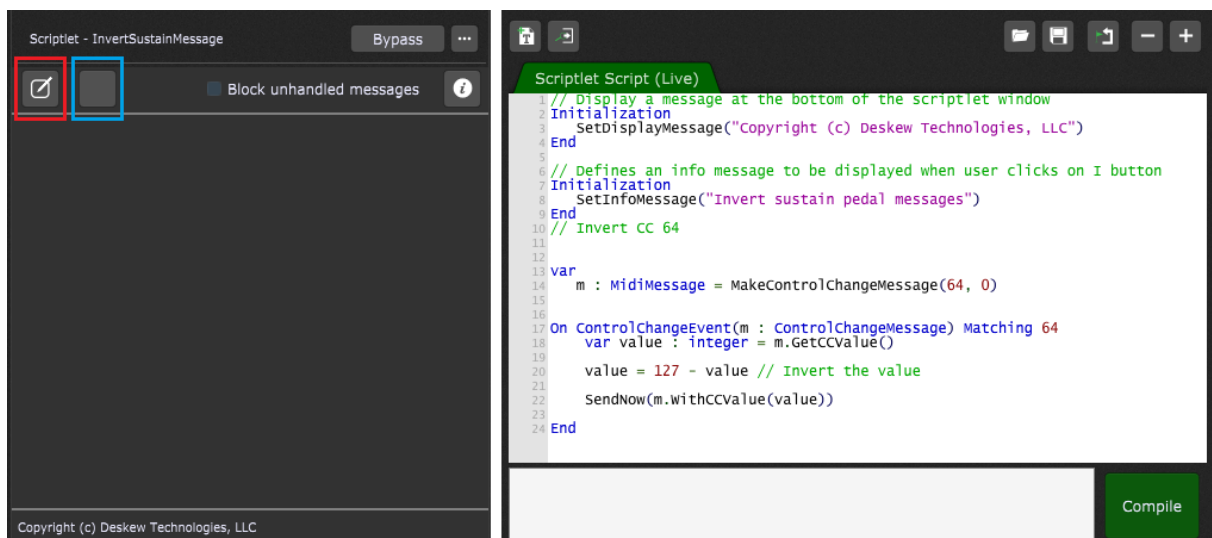


For example, if you have a sustain pedal that is producing CC messages the wrong way round (0 when pressed, 127 when released) then one of the solutions to fix this issue can be a simple scriptlet:



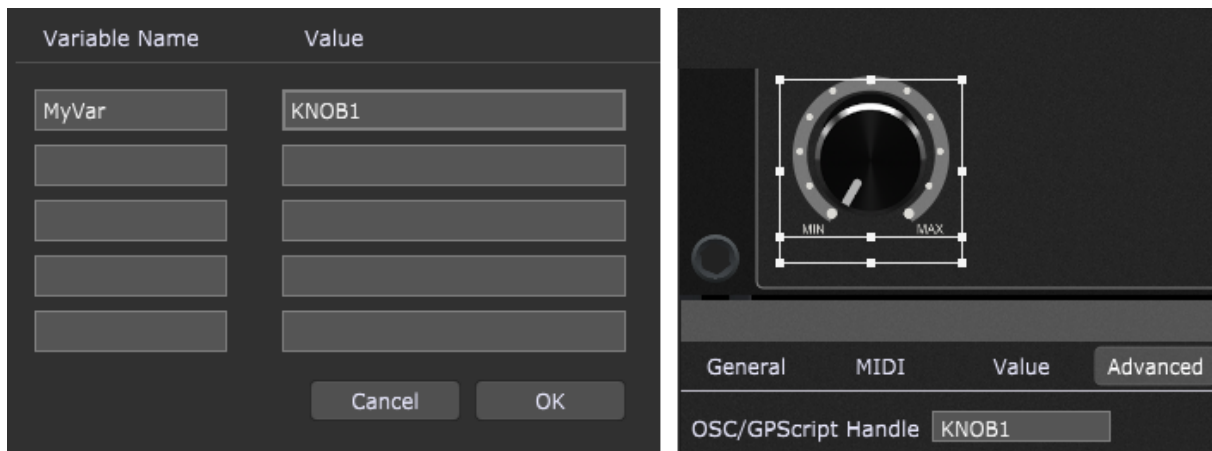
This scriptlet receives the MIDI In events from the MIDI In block, then captures CC64, inverts it, and lets everything else through.

If you want to see the underlying script behind this scriptlet, either click on the **Edit Script** button in the Scriptlet plugin editor (indicated with the red rectangle), or hold the **Shift** key while double-clicking the scriptlet block (which will open both the scriptlet's plugin editor and the Script Editor):



Using the Script Editor, scripts can be entered or modified; refer to the [GPScript](#) chapter to learn more.

The **Define User Variables** button (indicated with the blue rectangle) allows you to define up to 5 custom variables with their respective values.



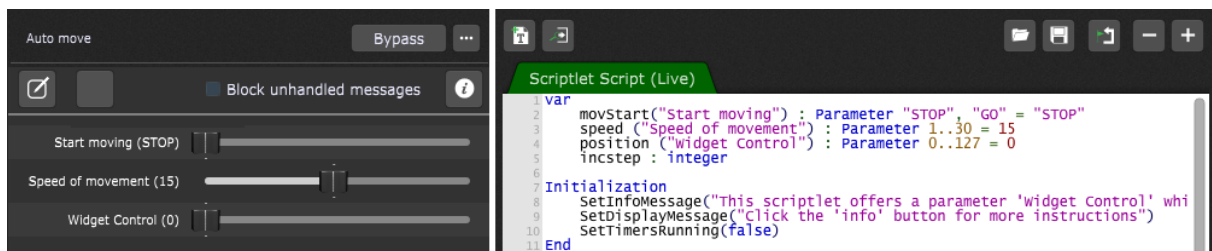
For example, type in the variable name **MyVar** and the value **KNOB1**. Afterwards, add a knob widget and in the *Advanced* tab type in **KNOB1**. Now you can use this example script to set widget values from this scriptlet:

```
var ExW : ExternalWidget
    W    : String

//Called when a NoteOn message is received
On NoteOnEvent(m : NoteMessage)
    W = GetUserVariable("MyVar")
    if BindWidget(ExW, W) then
        SetExternalWidgetValue(ExW, 1.0)
    end
End
```

You are also provided with the option *Block unhandled messages* that controls whether unhandled MIDI messages are passed through (the default) or blocked (when this option is checked).

Scriptlets can also have their parameters, which are then displayed in the scriptlet's plugin editor, like shown in the screenshot below:





Moreover, these parameters can be learned and controlled by widgets:



Refer to the [How to connect a widget to a plugin parameter](#) chapter to learn more.

Scriptlet parameters can also be captured in the [Probabilistic Sound Designer](#).

Click *Help* -> *Community* -> *Community Shared Gigs, Rackspace and More* to check out many Community generated content, including scriptlets.

## 10.10 System

This chapter presents the built-in plugins that belong to the System group: Auto Sampler Generator, Auto Sampler Recorder, Comment, GP Relay, and System Actions.

### 10.10.1 Auto Sampler Generator

The Auto Sampler plugins (generator and recorder) work together to automatically create a collection of samples from sounds you have created in Gig Performer using one or more synth and effect plugins connected together. The samples can then be loaded into

various samplers, giving you a lightweight solution to play back sounds that might otherwise be too CPU-intensive to use live. External synth sounds can also be captured.

The **Auto Sampler Generator** plugin produces a sequence of MIDI note messages, tailored according to the options you selected in this plugin block such as highest and lowest note, highest and lowest velocity and others that will be described in this chapter. The generated Note messages will then be processed by your plugins or a hardware synth and recorded with the [Auto Sampler Recorder](#) block. This plugin provides one MIDI output pin:



The Auto Sampler Generator block allows you to specify the following options:

**Highest and lowest note** - allows you to enter the desired highest and lowest note; type in numbers from **0** to **127** that correspond to notes from **C-2** to **G8** respectively. Depending on your use case, you may want to sample only a range of notes and not the complete set. Provided that you connected this block to a plugin and afterward the plugin to the Audio Out block, clicking on a note name will produce the corresponding sound.

**Important note:** In trial mode, you cannot change any of these values. You can continue to evaluate the usefulness of this plugin but you will not be able to create arbitrary sets of samples.

**Note interval** - you can select whether you want to sample every note (1), every second note (2) or every third note (3), and so on. Depending on the sound, you may not need to sample every single note; instead, let your sampler downsample single notes.

**Highest and lowest velocity** - allows you to select the desired values (from 1 to 127) for the highest and lowest velocities for each note.

**Velocity count** - some sounds sound quite different depending on the played velocity, so you can create separate samples for different velocity ranges (e.g. use one sample for velocities between 1 and 10, another sample for velocities between 11 and 20, etc.). The upside is that you get more accurate samples for different velocities. The downside is that you create multiple wave files (in fact, this is why some sample banks are so huge; they have each note sampled at almost every velocity value). Most modern samplers support multiple velocities for the same note.

**Note length** - choosing the right value (from 250 ms to 60000 ms) depends on the sound — if the sound doesn't change, then a short length may work if your sampler provides you with tools to loop the sample. If the sound changes, or you don't want to deal with loops, then you would want a larger note length. Note that longer note lengths also means longer files.

**Release tail** - determines how long the release portion of the sound should persist (from 20 ms to 20000 ms).

**Calculate using BPM** - turn on this toggle button if you want to calculate the required note length using BPM and beats.

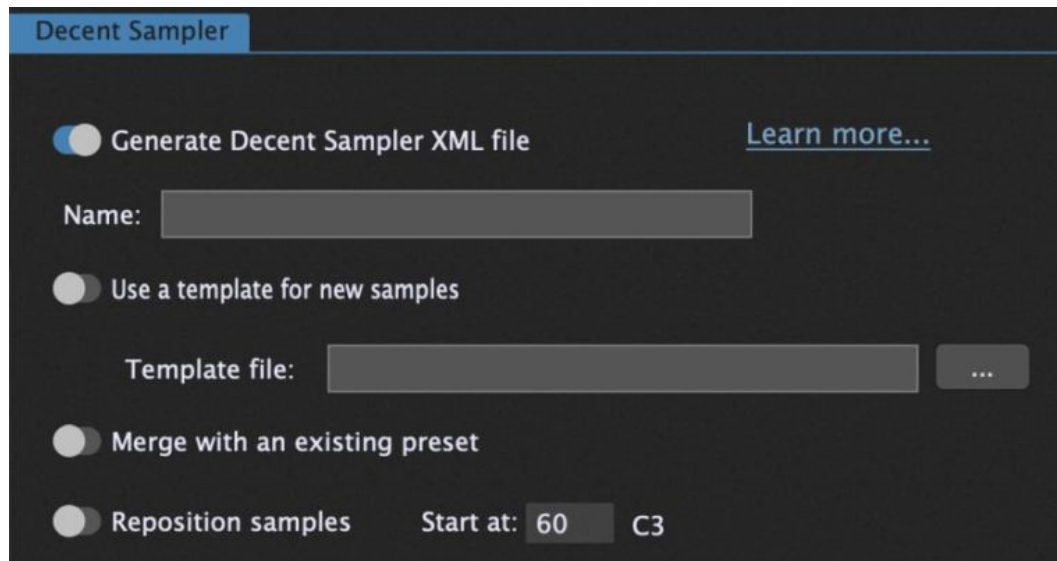
**BPM and beats** - if the *Calculate using BPM* toggle button is turned on, you can define BPM and number of beats to calculate the required note length automatically.

**Sample group** - allows you to define the group name for the sample collection. It is optional.

**Direction** - suppose you're only sampling every 5th note (Note Interval = 5). Now suppose you have the range: C C# D D# E. If you sample **Down**, the Auto Sampler will create the wave file for E and the sampler will down-sample that note to produce the other four notes below. If you sample **Up**, the Auto Sampler will create the wave file for C and the sampler will have to up-sample to create the notes at higher frequencies. That requires it to interpolate missing information whereas down-sampling lets you discard some information. Therefore, generally you will want to use the **Down** direction.

**Output folder** - click on the "..." button to select the desired output folder for your samples. If you want to open that folder, click the *Open Output Folder* button.

**Options** - clicking on this toggle button shows sampler-specific options:



**Generate Decent Sampler XML file** - turn on this toggle button to create a Decent Sampler preset (XML file) using the just generated samples. **Tip:** if you need an easy way to make a collection of sound effects easily accessible and triggerable from Gig Performer, read [this blog article](#) where we describe how to use the **Decent Sampler XML Generator**, a small free application we developed that will help you to use the free Decent Sampler plugin to accomplish this need.

**Name** - allows you to provide a name for the preset.

**Use a template for new samples** - if you have created a template XML file for Decent Sampler presets, the Auto Sampler copy that template and insert a reference to the newly created samples. It will be stored in the same location as the samples (check out the Output Folder above).

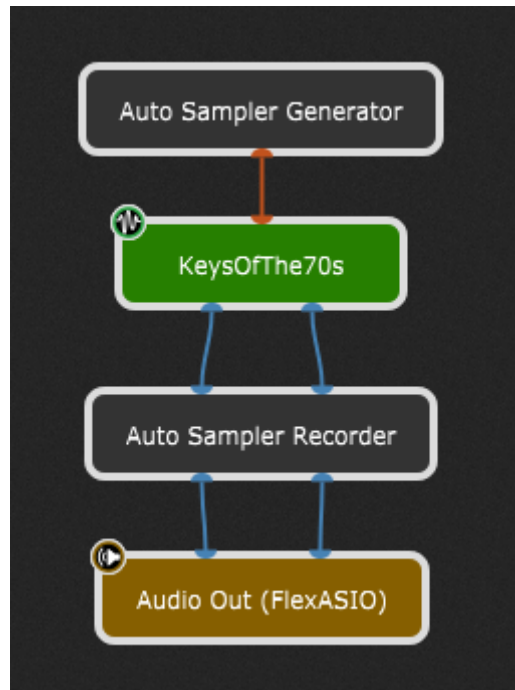
**Merge with an existing preset** - if you have already created a preset, you can have the Auto Sampler modify that preset to insert the sample group. In conjunction with "Reposition samples", this gives you the ability to create multiple groups of samples spread out to different locations on the keyboard.

**Reposition samples** - by default, Decent Sampler will place the samples using the note numbers from *Lowest Note* to *Highest note*. However, if you want those samples to be placed somewhere else, specify a new location for them.

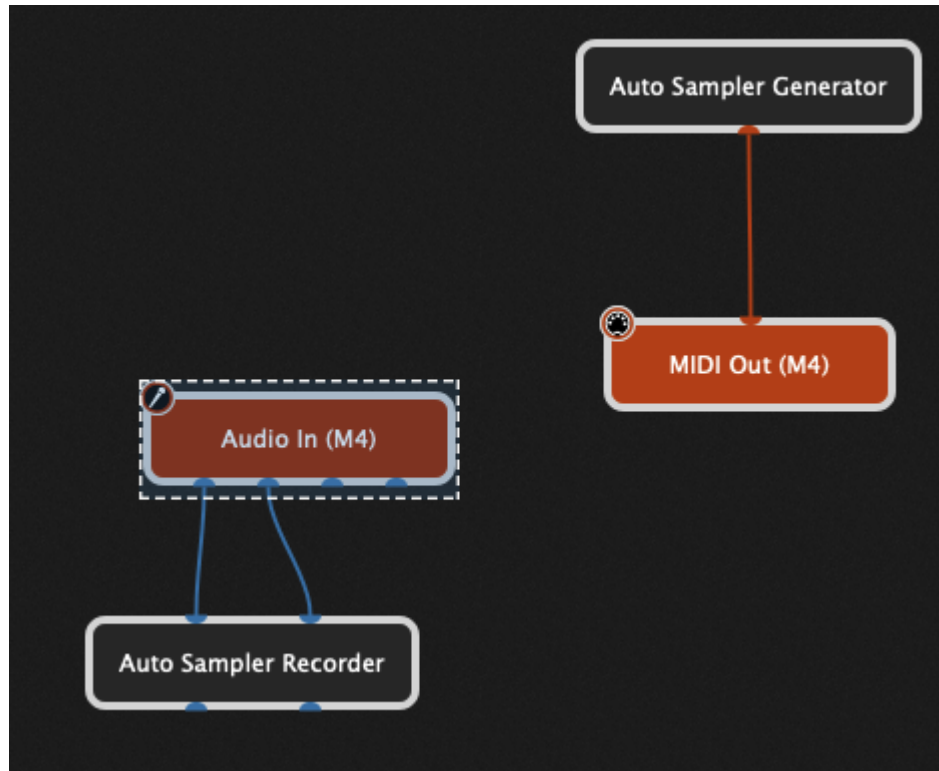
**Start at** - allows you to set the starting note.

Once you are finished with setting the desired options, click on the **Start** button to start the sampling process.

The screenshot below shows a simple setup that allows you to sample the sounds from the KeysOfThe70s plugin:



If you would like to sample an external synth, this is an example setup in the Wiring view:



First, connect audio outputs of the external synth to inputs of your audio interface and then connect the Audio In block with the [Auto Sampler Recorder](#). Afterward, connect the Auto Sampler Generator block with the corresponding MIDI Out block. Set the desired options in the Auto Sampler Generator and then click the Start button to start the sampling process.

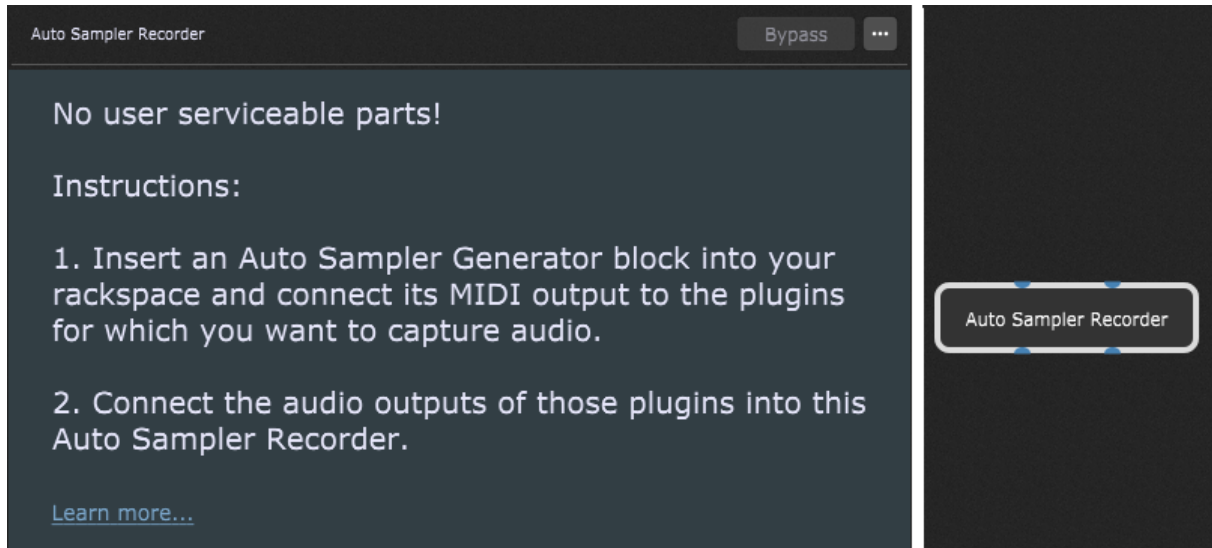
Check out [this YouTube video](#) to see the Auto Sampler in action.

### **Auto Sampler Generator parameters**

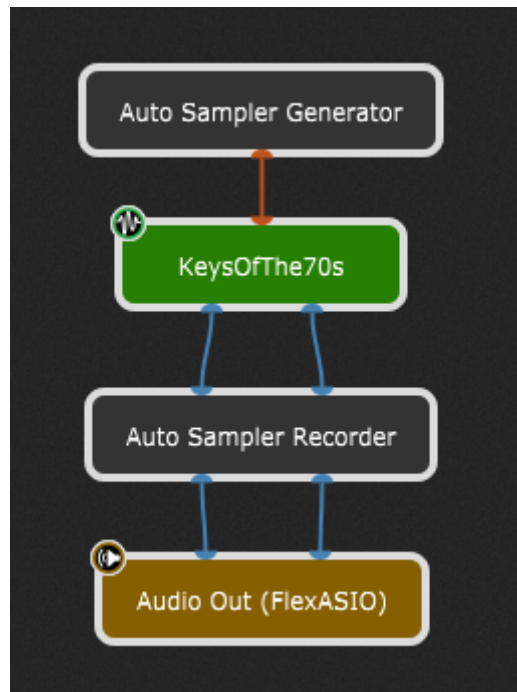
Apart from the common parameter **Open/Close Plugin Editor**, the Auto Sampler Generator plugin provides no other parameters.

### **10.10.2 Auto Sampler Recorder**

The **Auto Sampler Recorder** plugin allows you to record samples from your plugins or hardware synths. This plugin provides two audio input pins and two audio output pins:



This plugin's editor window doesn't have any controls that a user can control - it only provides guidelines on setting up and properly connecting plugins with Auto Sampler Generator and Auto Sampler Recorder. The screenshot below shows a simple setup that allows you to sample the sounds from the KeysOfThe70s plugin:



The auto-sampling process is started from the [Auto Sampler Generator](#) plugin.

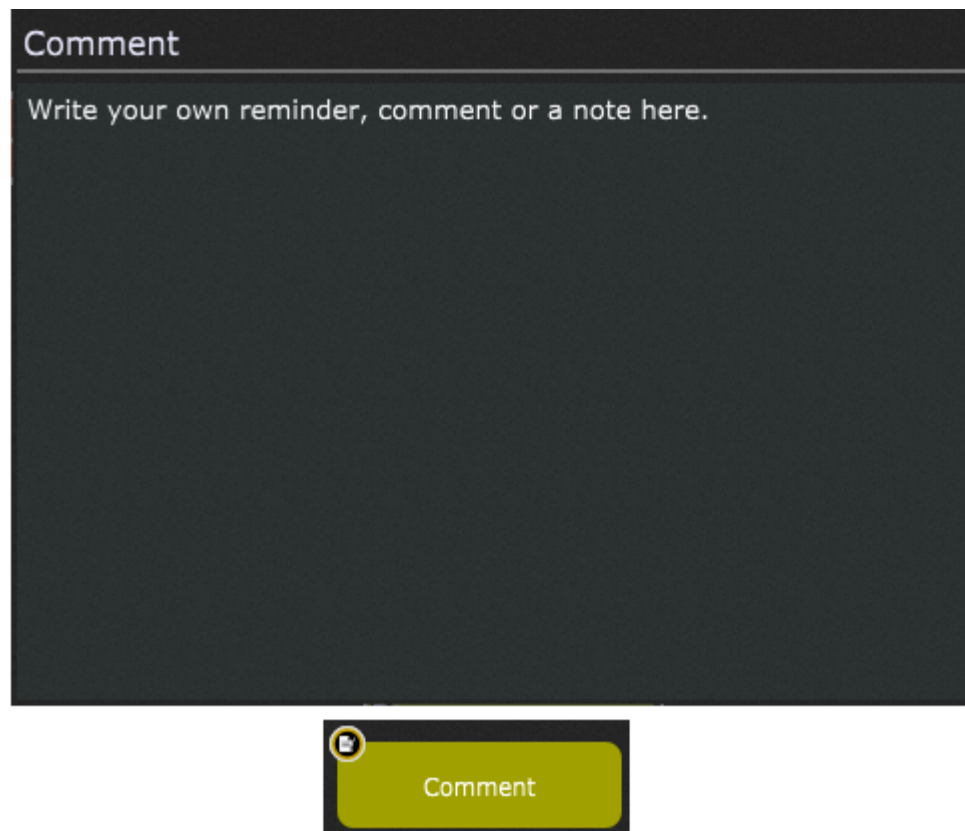
**Tip:** we also suggest that users place an Auto Sampler Generator and an Auto Sample Recorder in the Wiring view, select them both and save them as a [favorite](#). That's a convenient way to create both plugins when you need them again.

### **Auto Sampler Recorder parameters**

Apart from the common parameter **Open/Close Plugin Editor**, the Auto Sampler Recorder plugin provides no other parameters.

### **10.10.3 Comment**

The **Comment** plugin allows you to write your own reminder, comment or note; use it to keep notes about your rackspace design, for example. No input and output pins are available for this plugin:



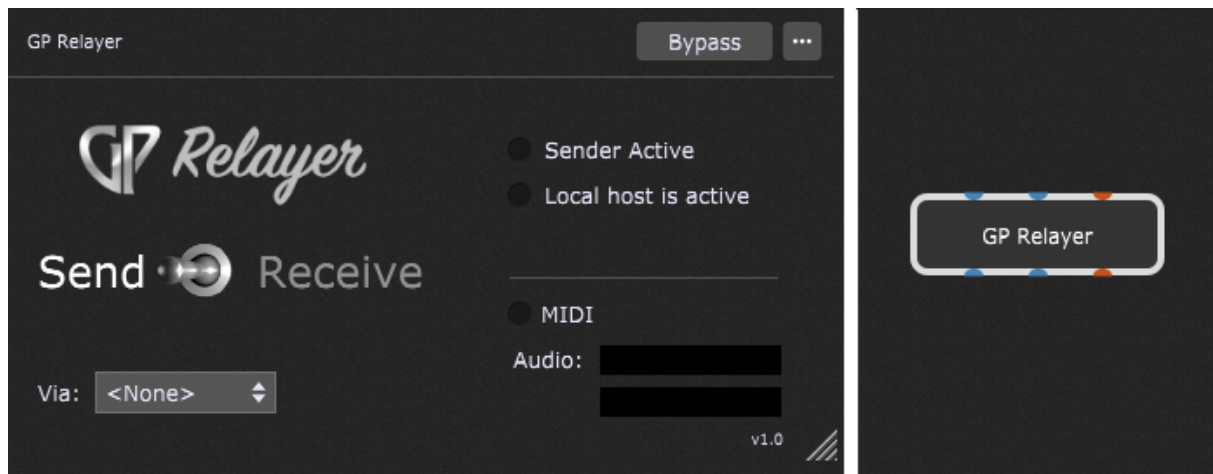
### **Comment parameters**



Apart from the common parameter **Open/Close Plugin Editor**, the Comment plugin provides no other parameters.

#### 10.10.4 GP Relayer

The **GP Relayer** plugin allows you to transport audio and MIDI between Gig Performer instances, to other DAWs for easy recording and from DAWs back to Gig Performer to utilize their proprietary sounds and effects. This plugin provides two audio input pins, a MIDI input pin, two audio output pins, and a MIDI output pin:



The elements in the GP Relayer plugin editor window are:

**Send / Receive** - 'Send' sets the GP Relayer to send audio and MIDI to another instance of GP Relayer set as receiver; 'Receive' sets the GP Relayer to receive audio and MIDI from another instance set as sender.

**Via** - select the desired relay (a transmission channel) from 1 to 10.

**Sender Active** - this indicator will light up if there is an active audio or MIDI transmission in the selected relay.

**Local host is active** - this indicator will light up if there is an active sender. When this indicator is not lit, the host (DAW, VST host or another Gig Performer instance) is not driving the GP Relayer plugin audio/MIDI. This state can happen with certain DAWs and plugin hosts (e.g. Logic and GarageBand) that have an optimization where they don't process plugins unless they are actively playing and recording. In this case, select play or record to receive audio from the GP Relayer plugin.

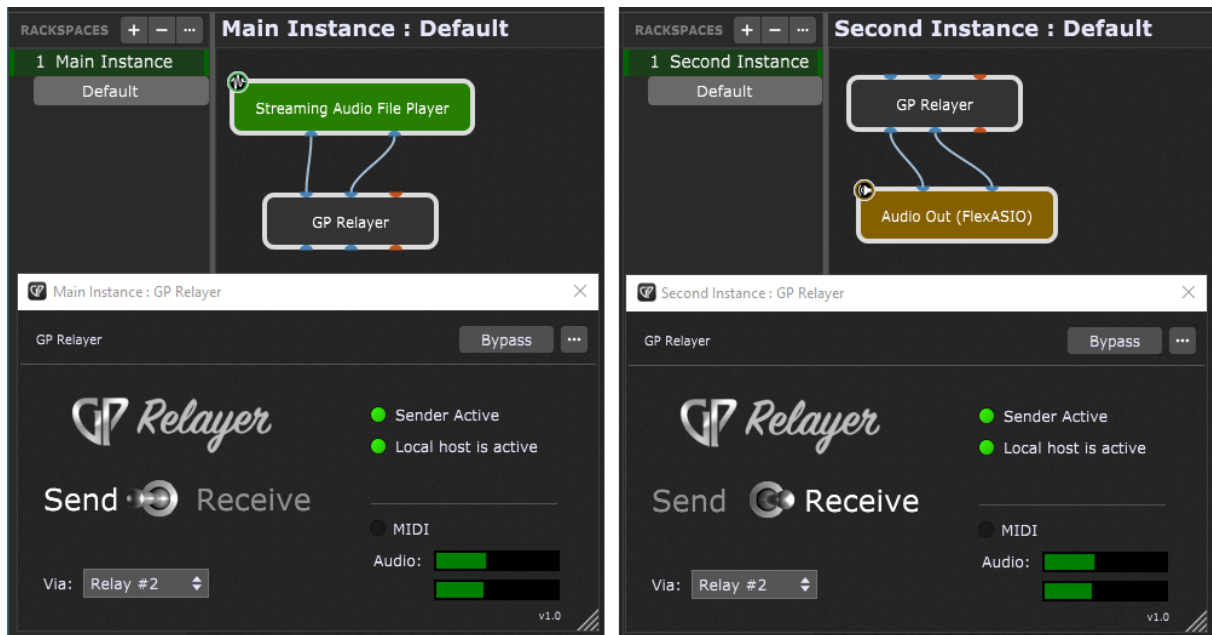
**GP Application** - this indicator appears when GP Relay is used inside a DAW or other VST host and indicates that Gig Performer is running. It doesn't appear in the Gig Performer internal version of GP Relay.

**MIDI** - this indicator shows MIDI activity.

**Audio** - shows the levels of audio that is being sent or received.

The example below shows how to send audio from the main [Gig Performer instance](#) to the second Gig Performer instance.

First, connect the audio source (in this case, a Streaming Audio File Player) to GP Relay in the main instance. Next, in the GP Relay plugin editor window, make sure that "Send" is selected and then choose the desired relay (in this example, Relay #2):



Switch to the second instance; in the GP Relay plugin editor window make sure that "Receive" is set, and then choose the relay number you set in the main instance (in this example Relay #2).

To hear the audio, connect the GP Relay outputs to your audio interface's outputs.

**Note:** multiple sender instances can simultaneously send audio/MIDI via the same relay, and multiple receiver instances will receive the mix of all senders for that relay.

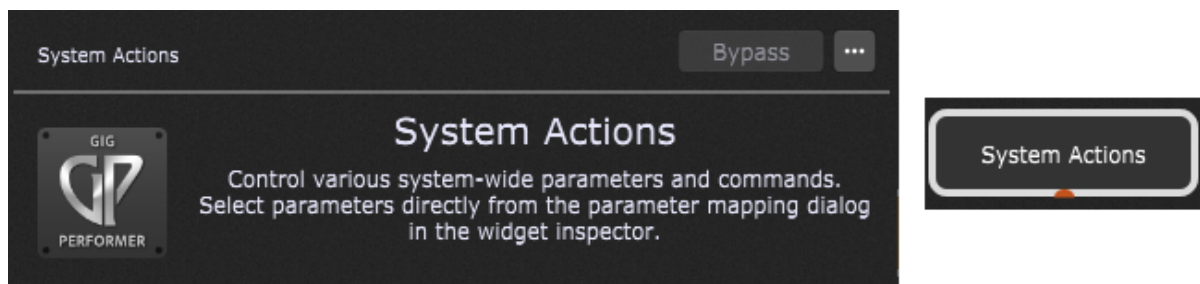
**Tip:** to see examples of how to configure GP Relay inside various DAWs, please visit [this community thread](#).

### GP Relayer parameters

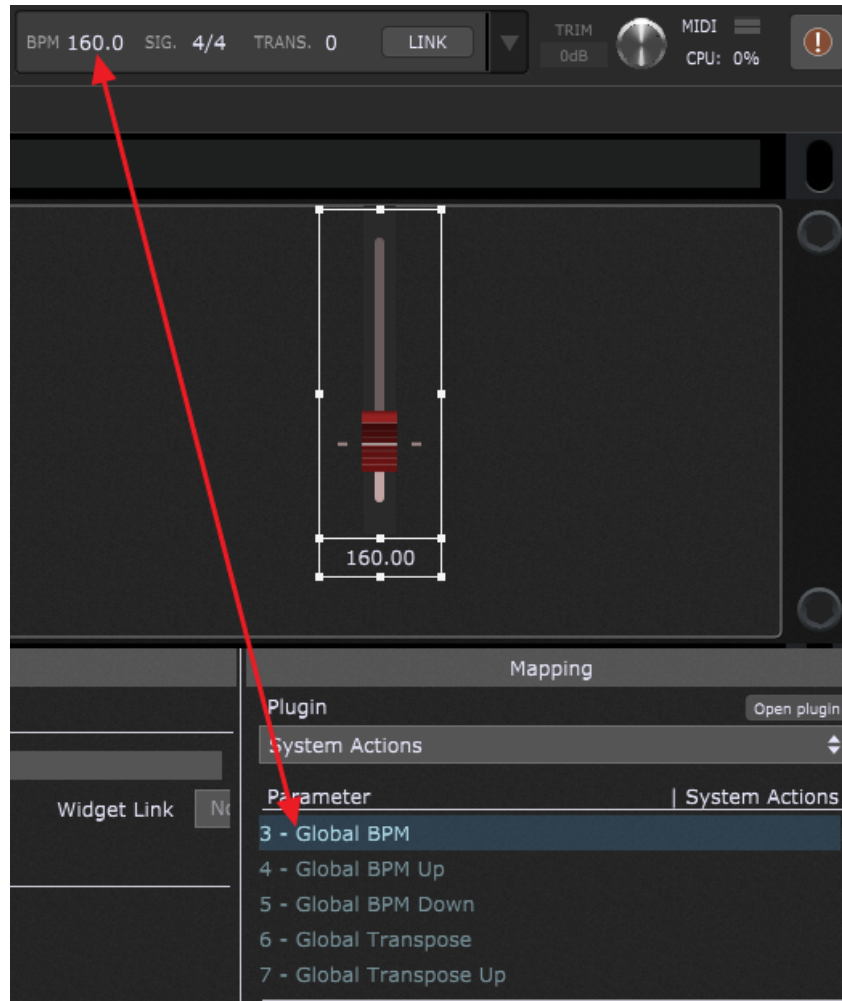
Apart from the common parameters **Bypass Plugin** and **Open/Close Plugin Editor**, the GP Relayer plugin provides no other parameters.

#### 10.10.5 System Actions

The **System Actions** plugin allows you to control various system-wide parameters and commands using widgets (and further through your MIDI controllers). This plugin provides one output pin:



The plugin editor for the Systems Actions does not have any configurable parameters. Select parameters directly from the parameter mapping dialog in the [Widget Properties Inspector](#):



You can, for example, set up the Red Slider widget to control your global tempo, as indicated in the screenshot above. Using a [scaling curve](#) with a small vertical range gives you the ability to easily push or pull the speed very slightly.

For more information, check out the blog article [The System Actions Plugin](#) and also this blog article [How to create a custom metronome with visual feedback](#) to see an example of using the System Actions block.

### System Actions parameters

The following parameters are available for the System Actions plugin:

- **Play** - turns on the global playhead (value greater than 0.5).
- **Stop** - turns off the global playhead (value greater than 0.5).

- **PlayStop** - value greater than 0.5 turns on the global playhead and value less than 0.5 turns off the global playhead.
- **GlobalBPM** - allows you to set or view the global BPM.
- **GlobalBPMUp** - a value greater than 0.5 increases the beat by 1 BPM.
- **GlobalBPMDown** - a value less than 0.5 decreases the beat by 1 BPM.
- **GlobalTranspose** - allows you to set or view the global transpose.
- **GlobalTransposeUp** - a value greater than 0.5 increases the transpose by 1 semitone.
- **GlobalTransposeDown** - a value greater than 0.5 decreases the transpose by 1 semitone.
- **MasterTrim** - allows you to set or view the master trim value.
- **Metronome** - turns the metronome on or off.
- **MetronomeVolume** - allows you to set or view the metronome value.
- **TapTempo** - a value greater than 0.5 generates a tap message.
- **AbletonLink** - turns AbletonLink on or off.
- **RackspaceName** - displays the name of the current rackspace (use in a label).
- **VariationName** - displays the name of the current variation (use in a label).
- **Rackspace-Variation-Name** - displays the name of the current rackspace and the current variation (use in a label).
- **SongName** - displays the name of the current song (use in a label).
- **SongpartName** - displays the name of the current song part (use in a label).
- **Song-Songpart-Name** - displays the name of the current song and the name of the current song part (use in a label).
- **ProgramChange** - if used in a label, it displays the last received PC value. If used in a knob (use in the Global rackspace), it will switch rackspaces.
- *Reserved1* - reserved for future use.
- *Reserved2* - reserved for future use.

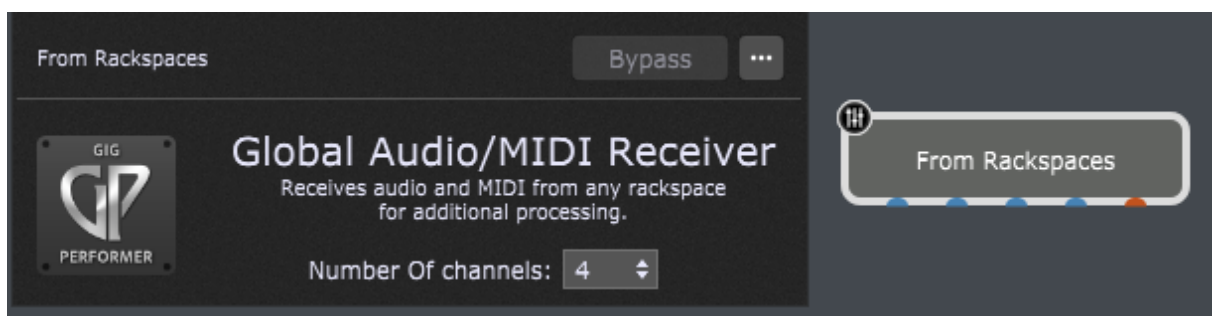
- **BeatMode** - Off; Countoff only (1 to 16 beats); On - Always.
- **Beat** - value will briefly go to 1 when a non-first beat occurs (attach to a pad widget to see a visual beat).
- **Accent** - value will briefly go to 1 when accent occurs, i.e. the first beat (attach to a pad widget to see a visual beat).
- **MidiBeat** - controls whether MIDI Note messages should be sent out on beats.
- **MidiBeatNoteNumber** - sets the note number for the regular beat.
- **MidiBeatNoteVelocity** - sets the velocity of the regular beat.
- **MidiBeatAccentNoteNumber** - sets the note number for the accent beat.
- **MidiBeatAccentNoteVelocity** - sets the note number for the accent beat.
- **Tuner** - a value greater than 0.5 toggles the tuner mode.
- **Panic** - a value greater than 0.5 triggers the panic message.
- **Next** - a value greater than 0.5 moves to the next variation or rackspace (depending on options).
- **Prev** - a value greater than 0.5 moves to the previous variation or rackspace (depending on options).
- **NextRackspace** - a value greater than 0.5 moves to the next rackspace.
- **PrevRackspace** - a value greater than 0.5 moves to the previous rackspace.
- **NextVariation** - a value greater than 0.5 moves to the next variation.
- **PrevVariation** - a value greater than 0.5 moves to the previous variation.
- **NextSong** - a value greater than 0.5 moves to the next song.
- **PrevSong** - a value greater than 0.5 moves to the previous song.
- **NextSongpart** - a value greater than 0.5 moves to the next song part.
- **PrevSongpart** - a value greater than 0.5 moves to the previous song part.
- **SelectPartOrVar** - allows you to select any variation or song part (up to 128 values) of the current rackspace or song, respectively.

- **SelectPartOrVar1-24** - allows you to directly select a song part or rackspace variation (1 to 24) of the current rackspace or song, respectively.
- **Songkey** - displays the current song key.
- **ChordProPageOffset** - controls the position and allows scrolling in the Song Lyrics/Chords Editor window (when the View tab is active). **Note:** this feature works as a proportion of the overall length of your ChordPro file, and it does not jump to song parts (it simply allows scrolling the lyrics file from the Global rackspace).
- **ScrollFrontPanel** - allows you to scroll the front panel. **Note:** attach to a global knob widget or learn a MIDI control.
- **PrevSongName** - displays the name of the previous song.
- **NextSongName** - displays the name of the next song.
- **RecordAudio** - allows you to easily start or stop recording. Attach it to (say) the *LED Button Red* in the Global rackspace and optionally learn a MIDI control on your MIDI device or surface.
- **TogglePanelSetlistView** - allows you to toggle between Setlist view and Panels view (note: don't use it in a local rackspace - use it from the Global rackspace).
- **TakeSnapshot** - when in the Setlist view, it allows you to take a snapshot, i.e. capture a variation into a song part.
- **CancelSnapshot** - when in the Setlist view, it allows you to delete the previously created snapshot.
- **PushSnapshot** - when in the Setlist view, it allows you to push the captured snapshot to the underlying variation.
- **RecompileAllScripts** - allows you to recompile all scripts (experimental).
- **DisplayBarBeat** - this is the control parameter for the CurrentBarAndBeat parameter; assign it to a button and turn it on so that you allow the current bar and beat to be updated on every beat.
- **CurrentBarAndBeat** - displays the current bar and beat; assign it to a label and make sure that the DisplayBarBeat parameter is on.
- **FullScreen** - maximizes the Gig Performer's window; assign it to a button to toggle the display.

- **KioskMode** - allows the Gig Performer's window to enter the full-screen mode; assign it to a button to toggle the display.
- **SetGlobalTranspose** - allows you to set the Global transpose value.
- **GetGlobalTranspose** - allows you to get the Global transpose value.
- **SetGlobalBPM** - allows you to set the Global BPM value.
- **GetGlobalBPM** - allows you to get the Global BPM value.
- **Scroll Rackspace** - allows you to scroll the rackspace list.
- **Scroll Setlist** - allows you to scroll songs in a setlist.
- **Save Gig File** - allows you to save the current gig file.
- **Next Setlist** - moves to the next setlist.
- **Prev Setlist** - moves to the previous setlist.
- **Setlist Name** - displays the name of the current setlist.

## 10.11 From Rackspace

The **From Rackspace** block is used to receive audio and MIDI from any rackspace and send them to the Global rackspace for additional processing. By default, this plugin provides four audio output pins and one MIDI output pin:



Double-clicking this plugin opens its editor window, where the number of channels can be selected. You can select up to 64 channels for this plugin.

**Important note:** changing the number of channels will update all the global plugin blocks in all rackspace. Affected plugins are:



- To Global Rackspace
- From Global Rackspace
- From Rackspace
- To Rackspace

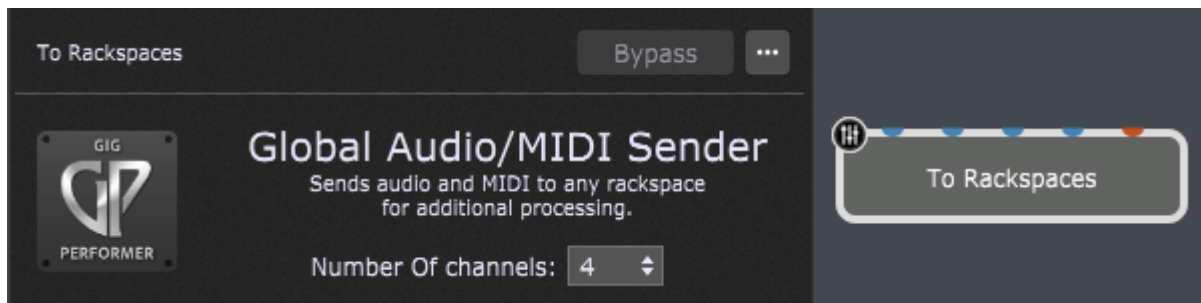
See more information on how to connect the From Rackspace plugin in the [Global Rackspace](#) chapter.

#### **From Rackspace parameters**

Apart from the common parameter **Open/Close Plugin Editor**, the From Rackspace plugin provides no other parameters.

## 10.12 To Rackspace

The **To Rackspace** block is used to send audio and MIDI from the Global rackspace to any rackspace for additional processing. By default, this plugin provides four audio input pins and one MIDI input pin:



Double-clicking this plugin opens its editor window where the number of channels can be selected. You can select up to 64 channels for this plugin.

**Important note:** changing the number of channels will update all the global plugin blocks in all rackspace. Affected plugins are:

- To Global Rackspace
- From Global Rackspace
- From Rackspace
- To Rackspace

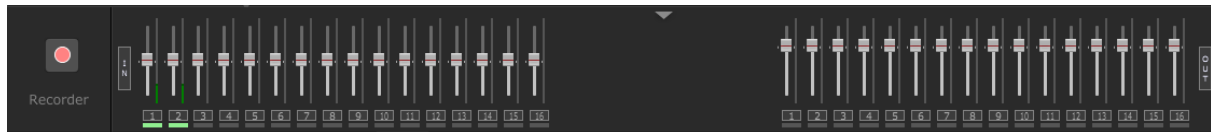
#### **To Rackspace parameters**

Apart from the common parameter **Open/Close Plugin Editor**, the To Rackspaces plugin provides no other parameters.

## 11 Recording your performance

Gig Performer includes a built-in recording feature that allows you to capture not only audio arriving at your audio interface's inputs, before it has passed through any of the plugins, but also outputs and all incoming MIDI messages.

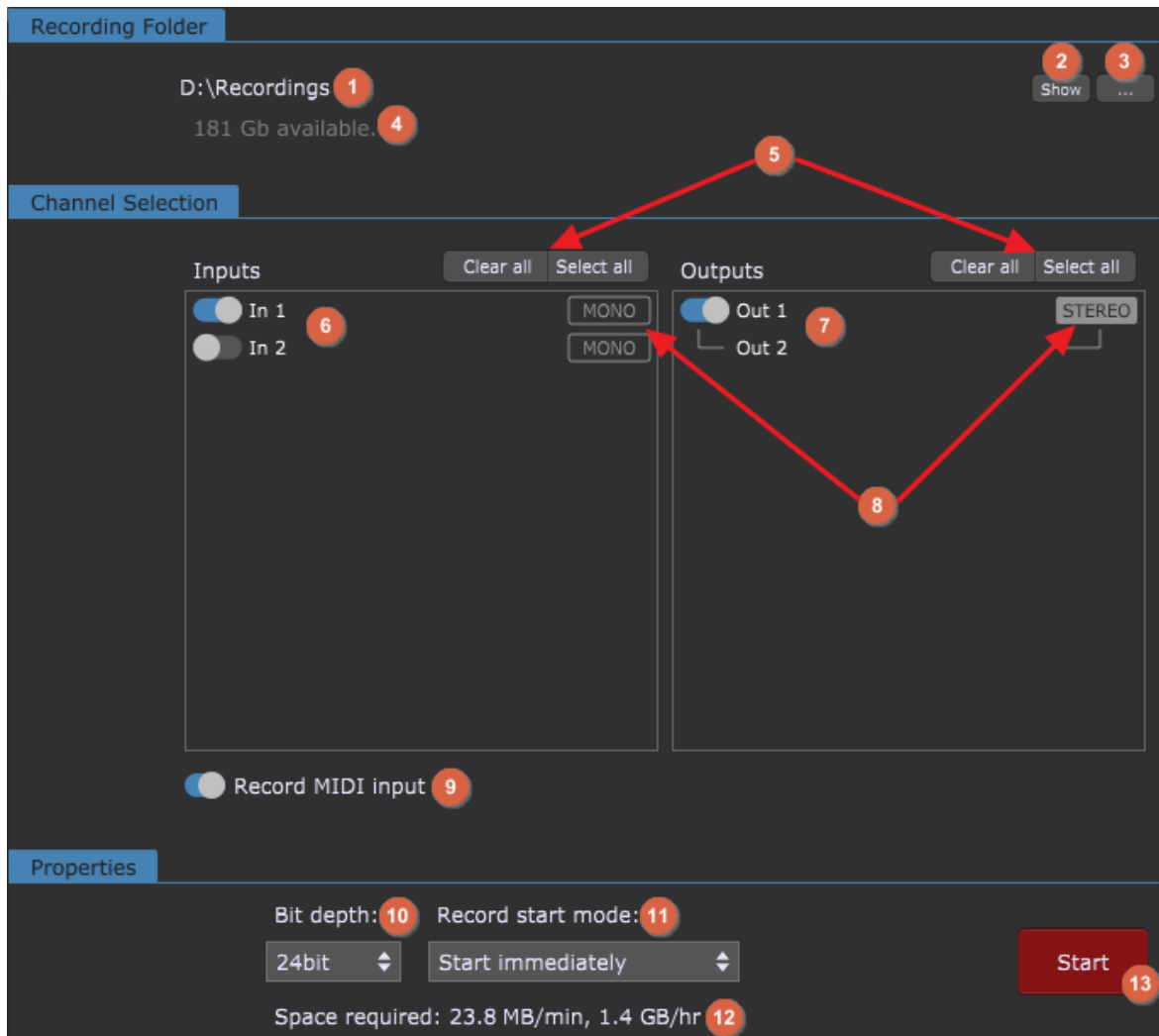
By capturing the audio and saving your gig, you can replicate the performance and modify it in post-production. To access the Recorder, just click the **Recorder** button in the bottom toolbar:



Files are recorded in WAV format. Each time you start the recorder, a subfolder is created within the (master) recording folder and named with the date of the recording (see the [File locations in Gig Performer](#) chapter for more information about default locations in Gig Performer). All files from your recording are placed within that subfolder. This makes it very easy to organize recordings made at different performances during a tour.

### 11.1 Recorder settings

When you click on the [Record button](#), the following dialog opens:



1. **Recording folder** - all files are recorded into subfolders of this recording folder. Subfolders are named with the date of the recording.
2. **Show button** - clicking this button opens the recording folder.
3. **Set master folder button** - clicking this button brings up a file browser that enables you to choose the (master) folder for your recordings.
4. **Available space display** - shows the amount of recording space available on the drive where the master folder is located.
5. **Clear All and Select all buttons** - clicking the *Clear All* button deselects all inputs (or outputs) so that no channels will be recorded. When you only need to record a few channels, it will be quicker to Clear all and then select only the ones you need. Clicking the *Select All* button enables all input or output channels for recording.

When you want to record most or all channels, it will be quicker to select all channels and then deselect the ones you do not need.

6. **Input channels** - click a toggle button to select an input channel for recording. On the screenshot above, one channel is selected for recording ("In 1").
7. **Output channels** - click a toggle button to select an output channel for recording. In the screenshot above, both channels are selected for stereo recording.
8. **Mono/Stereo buttons** - you can record either in mono or stereo. To create a stereo pair, click on the first of two consecutive mono channels.
9. **Record MIDI Input** - when checked, MIDI input from all physical MIDI devices is recorded at the same time as audio. A separate track is created for each device. The resulting MIDI file contains all the MIDI events separated into tracks by device name. These files are easily imported into any DAW with MIDI capabilities or other MIDI sequencing software.
10. **Bit depth field** - click this to select 8, 16, 24, or 32-bit (floating point) depth for recording. Note that choosing a higher bit depth results in larger files.
11. **Record Start mode** - see the [next section](#) for more information about this option.
12. **Space required** - shows the amount of disk space required to record the selected channels per minute or hour. This display adjusts as you select or deselect channels.
13. **Start button** - clicking this button immediately starts recording all selected input and/or output channels.

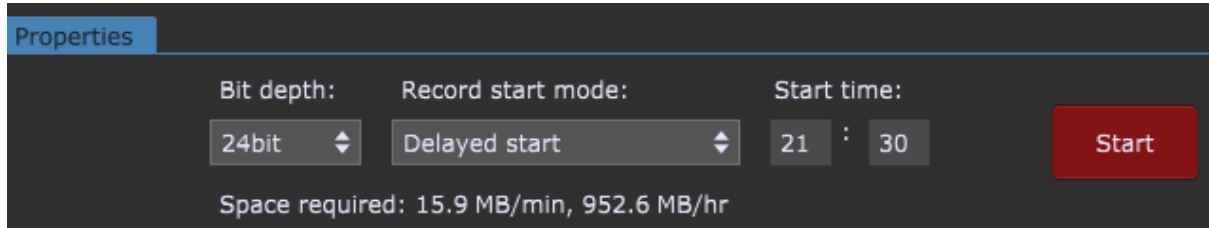
This dialog is similar to the [Recording Options](#) window, where recording settings can be configured.

## 11.2 How to use the Recorder?

To record your performance:

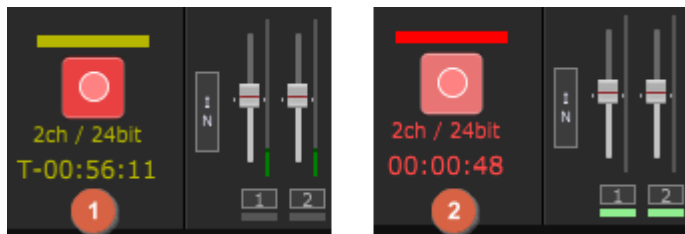
- First, make sure that there is enough free space on the drive where the (master) recording folder is located,
- Select the audio input and/or output channels that you want to record; if you also want to record MIDI input, turn on that option in [Recorder settings](#),

- Select the desired bit depth and check again if the space required for an hour (or a couple of hours) of your gig can fit onto your drive,
- Choose the record start mode: **Start immediately** or **Delayed start**; the **Start immediately** mode allows you to begin recording once the red **Start** button is clicked. The **Delayed start** mode activates some extra options and allows you to enter a start time in the minutes and seconds fields, as seen below:



This feature is useful if you want to configure and test the Recorder at soundcheck and then set it to start recording just before the performance begins.

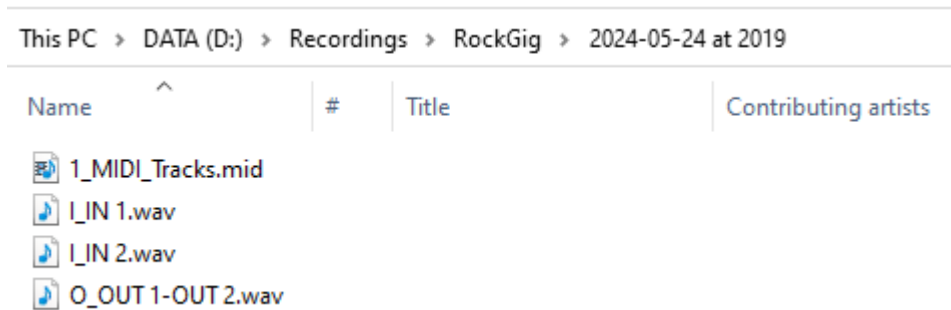
- Click on the **Start** button; note how the record button changes depending on the **Record start mode**:







If Record start mode is set to **Delayed start**, it will blink yellow and show you how much time remains until the recording process starts **(1)**. If the **Start immediately** mode is selected, it will blink red and show you elapsed time **(2)**.

- Clicking on the Recorder button will either stop the recording process or cancel the scheduled recording countdown.

- To access your recording, navigate to your master folder (see default locations on Windows and Mac [here](#)) and then enter the subfolder with the same name as your gig file (or Untitled, if you didn't save your gig); note that another subfolder (or subfolders if you made multiple recordings) is automatically named with date and time, i.e. "2024-05-24 at 2019":



The screenshot shows a Windows File Explorer window with the following path: This PC > DATA (D:) > Recordings > RockGig > 2024-05-24 at 2019. The window displays a table of files with columns for Name, #, Title, and Contributing artists. The files listed are:

Name	#	Title	Contributing artists
 1_MIDI_Tracks.mid			
 I_IN 1.wav			
 I_IN 2.wav			
 O_OUT 1-OUT 2.wav			

The name of each WAV file matches the name of each selected audio input and/or output. Note: recorded inputs are prefixed with "I\_" and recorded outputs are prefixed with "O\_". The name "O\_Out 1-Out 2.wav" implies that outputs are recorded in stereo.

**Note:** incoming MIDI events from all MIDI devices known to Gig Performer are recorded, each device to its own track in a Type 1 Standard MIDI File. Generally, each MIDI device corresponds to a physical MIDI port. Incoming messages are recorded using a resolution of 480 ppq (ticks per quarter note).

All real-time Gig Performer tempo changes are captured as Meta Tempo events in the MIDI file.

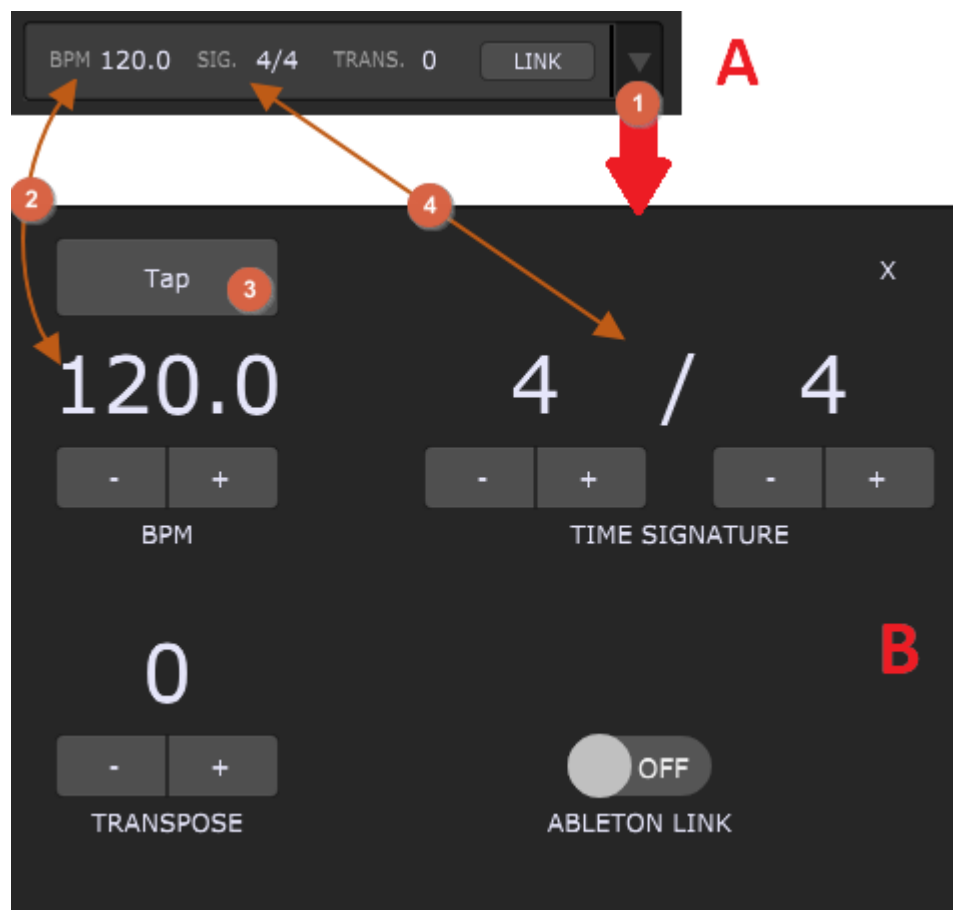
**Note:** the [System Actions](#) block has a parameter called *RecordAudio* and it allows you to easily start or stop recording. Attach it to (say) a *LED Button* in the Global rackspace and optionally learn a MIDI control on your MIDI device or surface.

## 12 Time controls

In this chapter, you will learn more about time controls in Gig Performer, how to set tempo and time signature, use the metronome, and activate Ableton Link.

### 12.1 Tempo and Time signature

Tempo and time signature settings are always visible on the upper toolbar (A) and can be adjusted using the drop-down window (B):



1. **Dropdown menu button** - click on this button or anywhere other than the actual BPM or Time Signature edit fields to open the drop-down menu window (B).
2. **Global tempo** - this is the tempo of Gig Performer's playhead. It can be edited directly in the upper toolbar (A) by typing in the desired value, or in the drop-down window (B), where you can either type in the desired value or use -/+ buttons to decrease/increase the tempo value. The maximum tempo value is 640 BPM. If the



metronome is enabled, its speed and accents will be defined by the global tempo and time signature settings.

3. **Tap tempo** - click or tap (if you are using a touch screen device) on the Tap button at least twice to set a corresponding tempo. Keep clicking/tapping to see how the tempo changes, based on your clicking/tapping speed. **Note:** once you start tapping, the BPM display will display the detected current tempo. You can choose between the several tap tempo algorithms; visit the [Tap Tempo](#) options to learn more. Alternatively, you can add a button (e.g. a pad button) to a panel in Edit mode of the Panels view and the [System Actions](#) plugin in the Wiring view; afterward, map the button to the *TapTempo* parameter of the System Actions plugin so you can tap tempo by clicking on that button (optionally, MIDI learn that widget if you want to tap tempo from your MIDI controller).
4. **Global time signature** - this is the time signature of Gig Performer's audio metronome. It cannot be edited directly in the upper toolbar (A), but only in the dropdown menu window (B). Click on the numerator and type in the number of beats per measure, then click on the denominator and type in the number of beats per measure. Alternatively, use -/+ buttons to decrease/increase their values. **Note:** you can choose any time signature, up to 32/64.

All plugins that respond to global tempo can be synchronized to the current tempo. For this to work, you will generally need to enable "sync" to external tempo on individual plugins. Please check your plugin documentation for more information.

**Note:** there are tempo settings in the [Song Properties](#) and [Song Part Properties](#) dialogs, besides the global tempo, and these settings are accompanied by the Override toggle button that makes that tempo active. Selecting a song that has a specified tempo and turning the the Override toggle button on sets the global tempo to that song tempo. The global tempo will revert to its default when you switch to a song part that does not override the global tempo. A song part with a specified tempo and the Override toggle button turned on is the highest priority, and it overrides both global and song tempos when the song part is selected.

## 12.2 Metronome

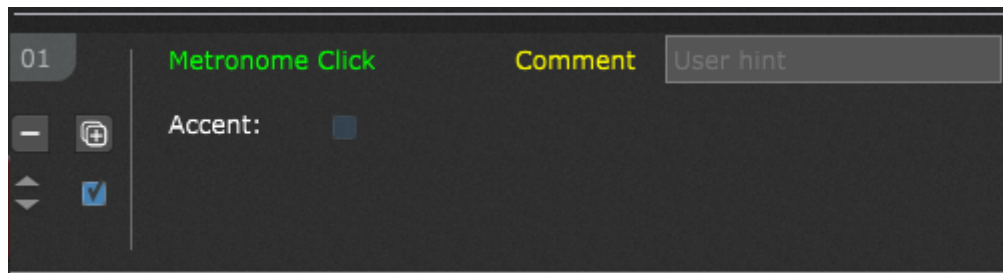
Gig Performer offers two separate metronomes: Audio Metronome and MIDI Metronome.

**Note:** the MIDI metronome is a built-in Audio File Player plugin feature. Refer to the [MIDI metronome](#) chapter for more information about the MIDI metronome.

Check out [this blog article](#) to learn how to create a custom metronome with visual feedback.

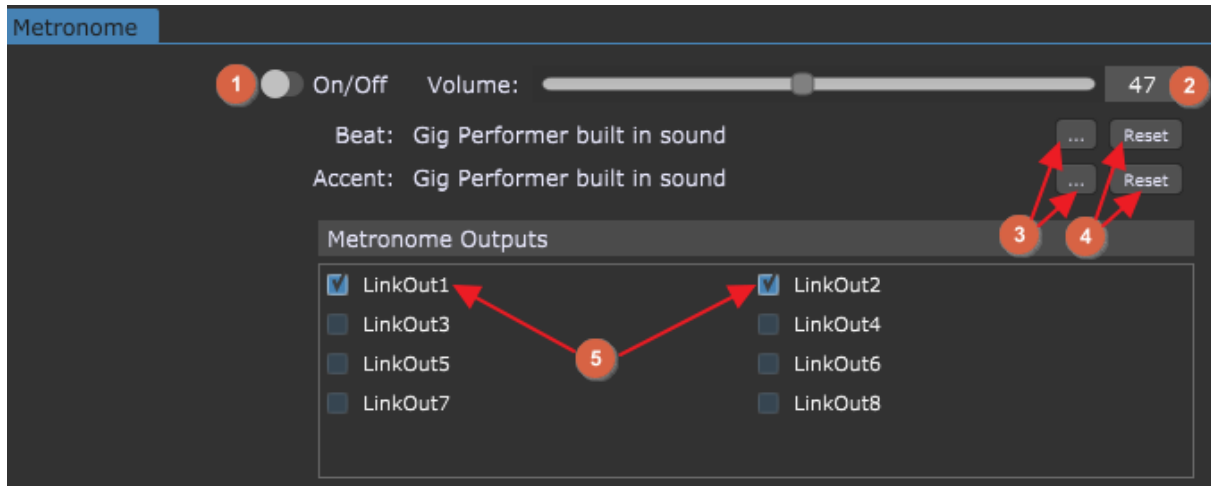


The [Action Management](#) window, accessible through the [Streaming Audio File Player](#) and the [Song Part Properties](#) window, allows you to send a single "Metronome Click" action.



### 12.2.1 Audio metronome

Setup for the audio metronome is done through the [General Options](#) window. Choose *Options* -> *General* or, press **Ctrl + ,** (or **Cmd + ,** on Mac) to open General options, and then note the Metronome section:



1. **On/Off button** - click this toggle button to activate/deactivate the metronome.
2. **Metronome playback volume** - to set the volume of the metronome, drag the slider or click in the field and enter the value directly.
3. **Metronome sound select area** - click the "..." buttons to select a custom sound file (aif, wav, mp3, flac, and ogg files are supported) for Beat and Accents sections respectively.
4. **Reset Metronome sound to default** - clicking on the Reset buttons resets the Beat and Accent sounds to Gig Performer's default sounds.
5. **Metronome outputs** - click on the check boxes for each output to which you want the metronome routed.

The audio metronome is simple to use: set the outputs to which you want the metronome routed in the General tab (refer to the screenshot above, indicated with (5)). Then set the desired global tempo and time signature in the top toolbar, and activate the metronome.

**Note:** the metronome on/off buttons in the top toolbar and on the General tab have identical purposes; changing one changes the other as well.

The first beat of each bar is accented.

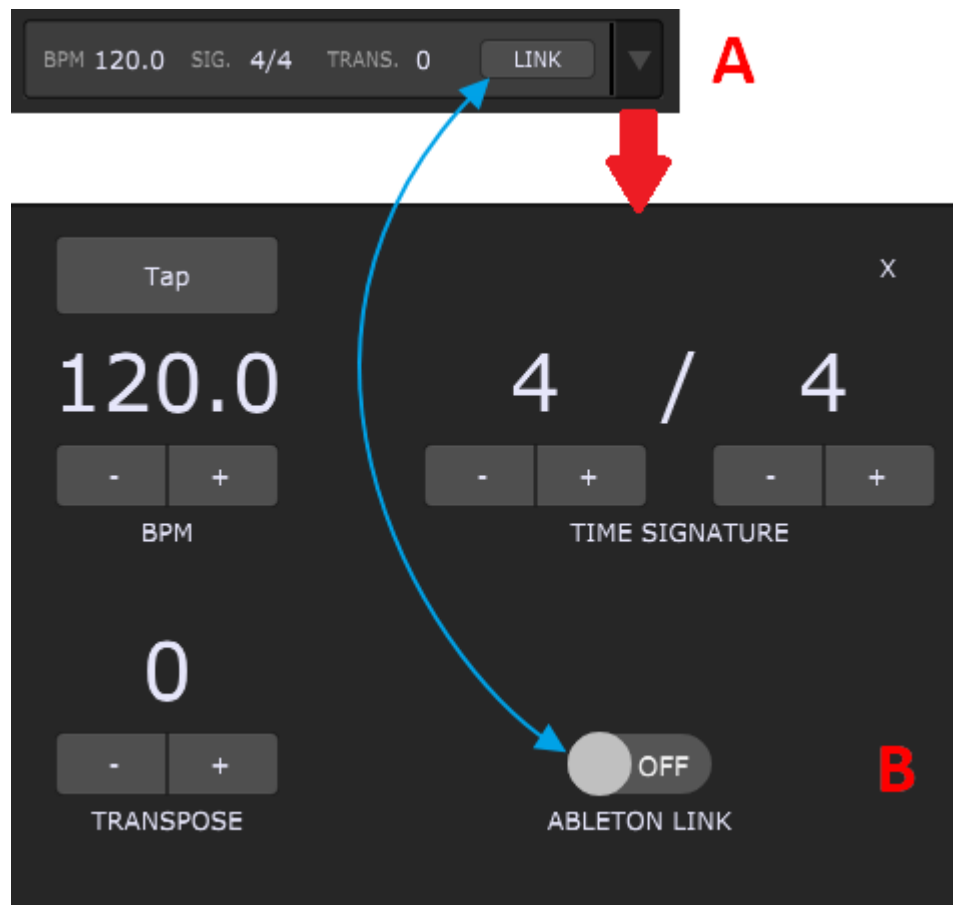
To learn more about setting the tempo and time signature, refer to the [Tempo and Time signature](#) chapter.

To see more about the buttons in the top toolbar, refer to [The main Gig Performer Window](#) chapter.

### 12.3 Ableton Link support

Ableton Link is a system for synchronizing multiple music applications, which may be running on different devices, by providing unified, real-time tempo control. Link-enabled programs find each other automatically when they are on the same network. Any "player" in the network can change the tempo and the others follow.

To enable Gig Performer to sync to Ableton Link, click on the Link button in the upper toolbar (A) or enable the toggle button in the drop-down menu window (B):



No configuration of any kind is necessary, just turn it on.

**Note:** Optionally, you can synchronize Start/Stop with Ableton Link. Refer to the *Sync Start/Stop with Ableton Link* option in the [General Options](#).

Visit [this blog article](#) to learn how to synchronize Gig Performer's tempo to an external MIDI clock or Link-enabled products.

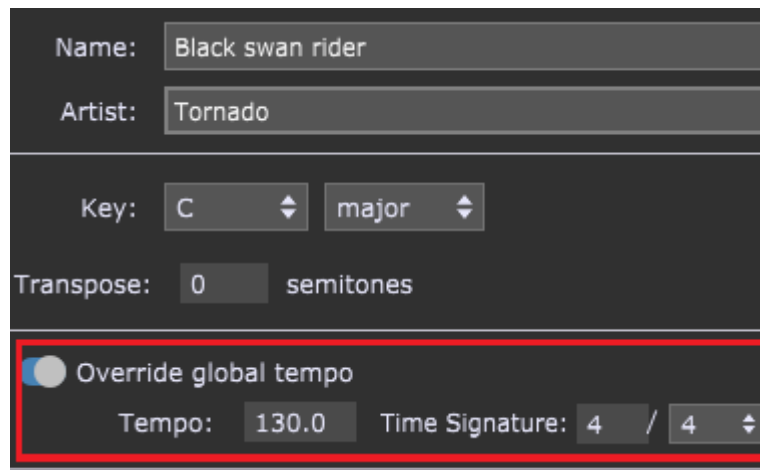
## 12.4 Rackspace or Song tempo overrides

The [global tempo](#) and time signature can be overridden by custom tempo and time signature that can be set for each rackspace:



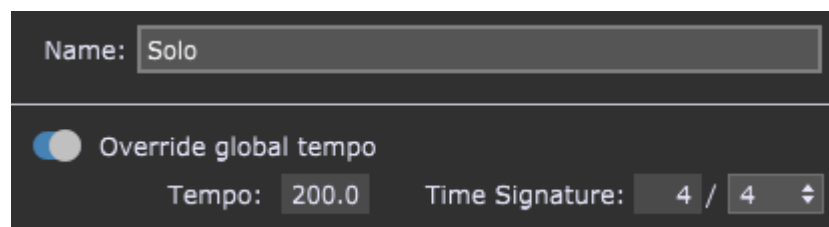
Check the [Rackspace Properties](#) chapter for more information.

When you enter the Setlists view, the global tempo and time signature can be overridden by a custom tempo and time signature that can be set for each song:



Refer to the [Song Properties](#) chapter to learn more.

Song parts can also have their overrides and they represent the highest priority in the override order:



This option is configured in the [Song Part Properties](#) dialog.

**Note:** if you add the **Set Global BPM** action in the Song Part properties dialog, it will override the **Override global tempo** setting described above.

## 13 OSC and Gig Performer

Gig Performer provides extensive support for OSC ([Open Sound Control](#)) - a protocol for communication among various multimedia devices and applications, with a particular focus on applications used by artists and performers. In Gig Performer you can associate OSC names with plugins and widgets and then send messages to them from any remote device, such as Android phone or tablet, iPhone, iPad or other laptop to control their operation. Learn more about OSC support in Gig Performer [here](#).

Every parameter available in your VST, VST3 and AU plugin (if you're on macOS) can be controlled by widgets in Gig Performer, and every widget can be controlled just with a tap of or sliding your fingers on your favorite mobile phone via OSC protocol. Other OSC capable devices, like various digital mixers for example, can be controlled directly from Gig Performer using widgets and OSC. Gig Performer's proprietary programming language ([GPScript](#)) also supports the OSC protocol.

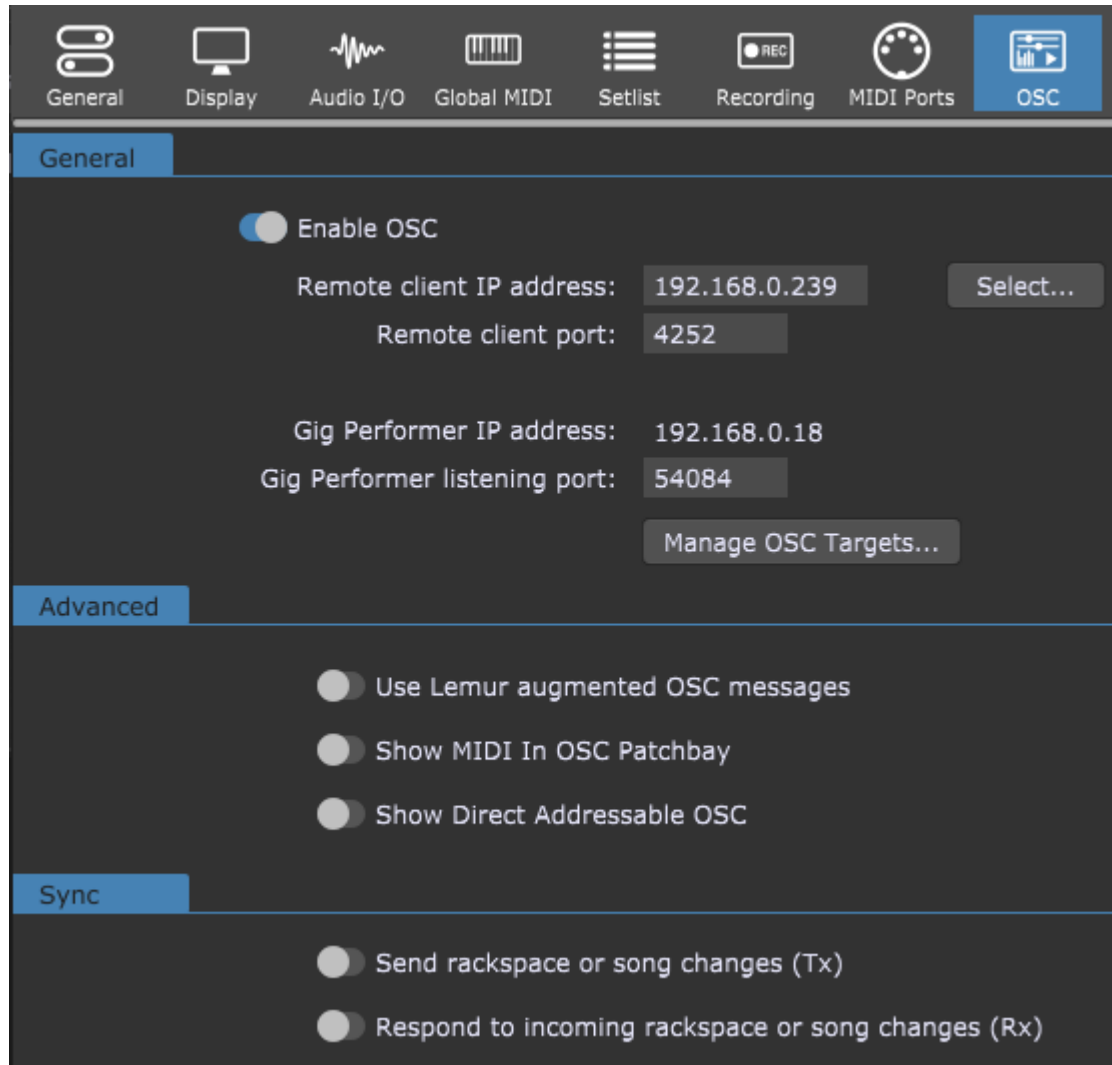
For example, you can use OSC to change the output levels of each plugin independently or adjust the cutoff frequency of a synth plugin. It can also be used to set Tap Tempo, turn Tuner View on and off, and select the desired rackspace and variation. In addition to well-known OSC products like Lemur and TouchOSC, it is also possible to use programming tools like Max/MSP, Ableton Live, and Plogue Bidule (to name just a few) to create very sophisticated applications that can also work interactively with Gig Performer.

For the most current version of Gig Performer's OSC implementation, see Help -> Documentation -> OSC Messages.

### 13.1 How to set up OSC

For Gig Performer and an OSC application to work with one another, ensure that both your computer and the device running the OSC application are connected to the same local network.

Launch Gig Performer and then click Options -> [OSC Setup](#):



Enable OSC by clicking on the **Enable OSC** button. Type in the IP address and the port of the device running the OSC application in the **Remote client IP address** and **Remote client port** boxes respectively. Alternatively, click on the **Select...** button to easily select an OSC target from the list that must be previously set by clicking on the **Manage OSC Targets...** button. Check the [OSC Options](#) to learn more.

Note that the **Gig Performer IP address** value cannot be changed, as this is defined by your computer's network configuration. The address is displayed to make it easier for you to configure your remote device to connect to Gig Performer. The **Gig Performer listening port** can be changed to a custom value. **Note:** make sure that Gig Performer is not blocked by any kind of software firewall.

When you perform live, you might want your FoH engineer to have access to your controls, too. In cases when OSC applications are running on multiple devices (i.e. a



tablet, a smartphone or another computer) and you want to use multiple devices onstage, there is a simple way to accomplish that: if you simply set three last digits of the OSC applications IP address to 255 (i.e. 192.168.0.255) then **Broadcast Mode** is activated. Broadcast Mode enables Gig Performer to send OSC messages and communicate simultaneously with all listening clients on the specified port (i.e. 4252) on the same local network. [Click here](#) to access our Knowledge Base article with more information.

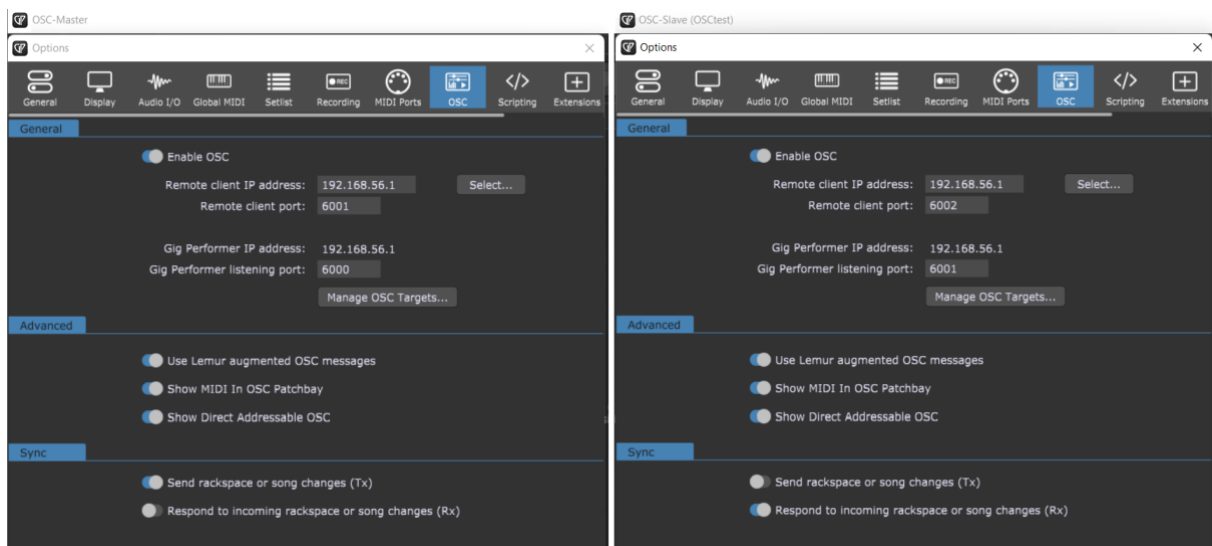
If you use an OSC application such as Lemur or another OSC application that allows detailed control, turn on the **Use Lemur augmented OSC messages** toggle button.

Other advanced options include **MIDI In OSC Patchbay** and **Direct Addressable OSC**. Check the [OSC Options](#) chapter for more information.

The **Sync** section is dedicated to synchronize rackspace or song changes between [multiple instances](#) of Gig Performer. The requirement is that rackspaces and songs have the same names across instances. Under this section there are two toggle buttons:

- **Send rackspace or song changes (Tx)** - allows you to send messages to other instances of Gig Performer to switch rackspaces or songs.
- **Respond to incoming rackspace or song changes (Rx)** - allows you to set this instance of Gig Performer to respond to incoming requests to change rackspaces or songs from another instance of Gig Performer.

The example screenshot below shows the OSC options of the main Gig Performer instance and the "OSCTest" instance:



In this example, these two instances are on the same computer. Provided that rackspace names are the same (variation names don't need to be the same, as their index number

is used for the sync), clicking on the desired rackspace in the main instance will automatically change to the corresponding rackspace in the "OSCtest" instance.

Gig Performer's supplied [MIDI In \(OSC\)](#) plugin is designed to receive MIDI messages from remote OSC sources. One of the requirements of OSC is that each object in both the host computer and remote device must have matching unique case-sensitive names comprised of characters only (no spaces). These names must be unique within a single Gig Performer rackspace, although they can be reused in different rackspaces. To see an example of how to use MIDI OSC blocks, refer to [this blog article](#).

## 13.2 OSC Resources

The OSC-related topics are extensively covered in the Gig Performer blog, and we will list useful blog articles in this chapter:

- [Working with TouchOSC and Gig Performer](#)
- [How to control your guitar or keyboard effects from a Web browser](#)
- [How to use your mobile phone to control your guitar or keyboard effects](#)
- [Using Lemur with Gig Performer](#)
- [Using TouchOSC app as a remote display and a touch control surface with Gig Performer](#)
- [The Lemur and TouchOSC templates](#)
- [Using MIDI OSC blocks to send MIDI to the Global rackspace](#)
- [Gig Performer as an OSC controller](#)

Check out our [blog](#) for more tutorials, and also visit our [Community](#) for more tips on using OSC.

If you want to check out our current application-specific tutorials, please visit [this community thread](#).

## 14 Multi-instance support

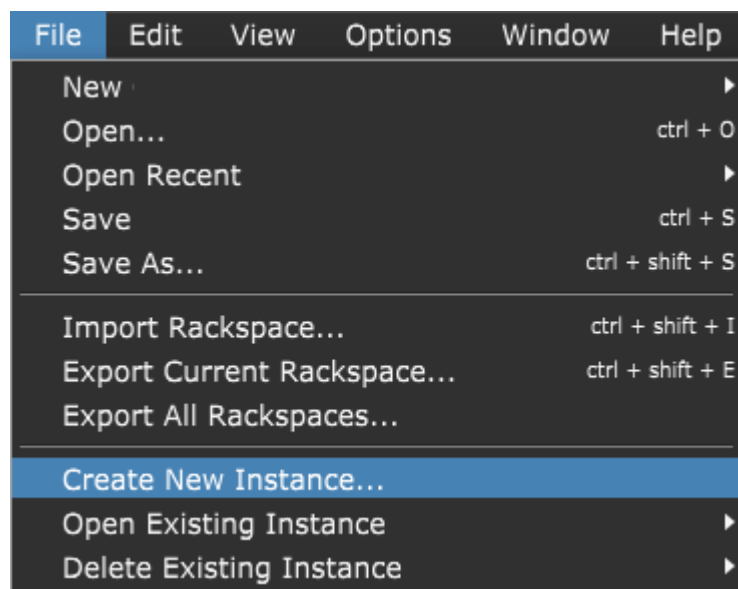
Multi-instance support means that you can run Gig Performer multiple times on the same computer, with each such "instance" operating independently.

This has many different practical applications. For example, each band member can have the ability to change the rackspace affecting their particular instrument without changing the rackspaces affecting different instruments; they can even use separate MIDI controllers or pedal-boards to do so.

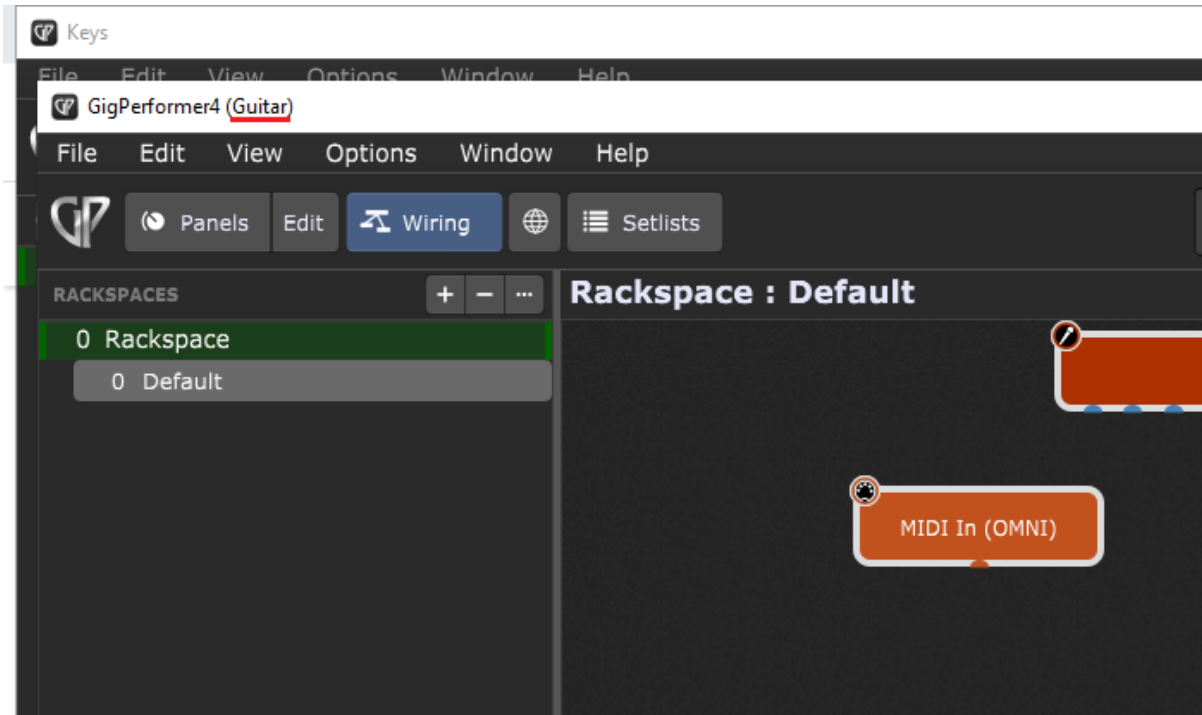
There are also many advanced usages: for example, you can use a MIDI In plugin to send program change messages from one instance to another so that one instance can be used to change the rackspace in another instance.

The number of instances you can successfully open at any one time depends on your system's RAM and CPU resources. Each instance can access entirely different plugins and MIDI In blocks. Separate instances can also access different audio interfaces.

Instances are managed through the File menu, where you can create new instances, and open or delete existing instances:



Creating another instance of Gig Performer is easy. Just select **File** -> **Create New Instance...** type in a name, e.g. Guitar, and you are all set:



**Note:** do not open the same instance twice as both opened instances will respond to the same MIDI and OSC commands and changing any settings in one of the instances will change it for both, but only one will get the change immediately.

If you want to use two or more instances with the same audio interface, make sure to set the same sample rate in [Audio Options](#); otherwise, one of the instances will have a sound that's slightly out of tune and will sound incorrect.

To learn more about Gig Performer instances, check out this in-depth blog article: [How to use multiple instances in Gig Performer](#).

Gig Performer allows you to synchronize rackspace or song changes between multiple instances. The requirement is that rackspaces and songs have the same names across instances. Check out the [How to set up OSC](#) chapter for an example of the sync setup.

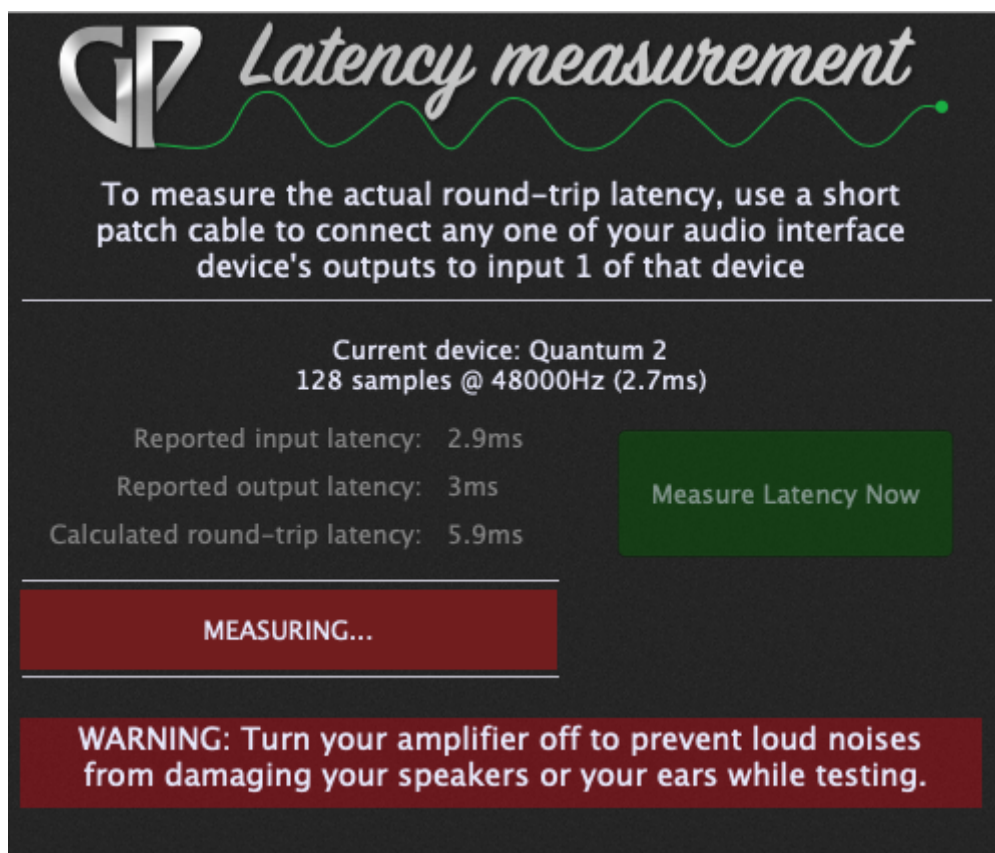
**Note:** some Windows ASIO drivers still allow only one application to use the audio interface at a time; these are known as single-client drivers. Multi-client audio drivers started coming out years ago, but, there are still many single-client drivers in place. Two instances of Gig Performer count as two applications, which is a problem when running on a single-client driver. Read [this blog article](#) to find out how to run multiple instances of Gig Performer with a single-client ASIO driver on Windows.

You can easily send audio and MIDI between Gig Performer instances using [GP Relayer](#).

## 15 Latency measurement tool

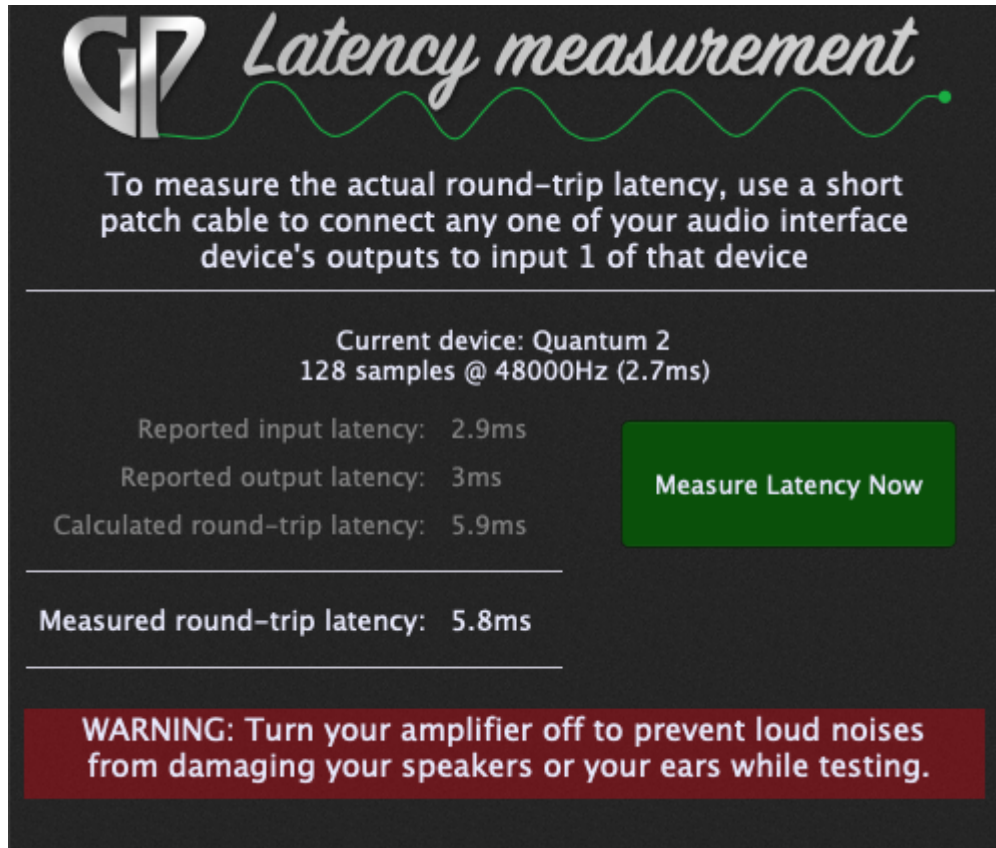
One of the factors that impacts your overall latency is the additional latency introduced by your audio interface. Not all audio interfaces are created equal. Some have low internal latency for both A/D and D/A conversions while others add more latency than perhaps you might like, so you have to lower your buffer size to get the overall desired latency.

You can use the built-in Latency Measurement Tool to find out how much extra latency your interface introduces. Click **Window** -> **Measure Device Latency** to measure actual round-trip latency:



To measure, follow these steps:

1. Mute your speakers or turn them all the way down.
2. Connect a physical patch cable from any output from your audio interface to input 1 of the interface.
3. Click the **Measure Latency Now** button.
4. The result will appear in the Measured round-trip latency display.



**GP** *Latency measurement*

To measure the actual round-trip latency, use a short patch cable to connect any one of your audio interface device's outputs to input 1 of that device

---

Current device: Quantum 2  
128 samples @ 48000Hz (2.7ms)

Reported input latency: 2.9ms  
Reported output latency: 3ms  
Calculated round-trip latency: 5.9ms

**Measure Latency Now**

---

Measured round-trip latency: 5.8ms

---

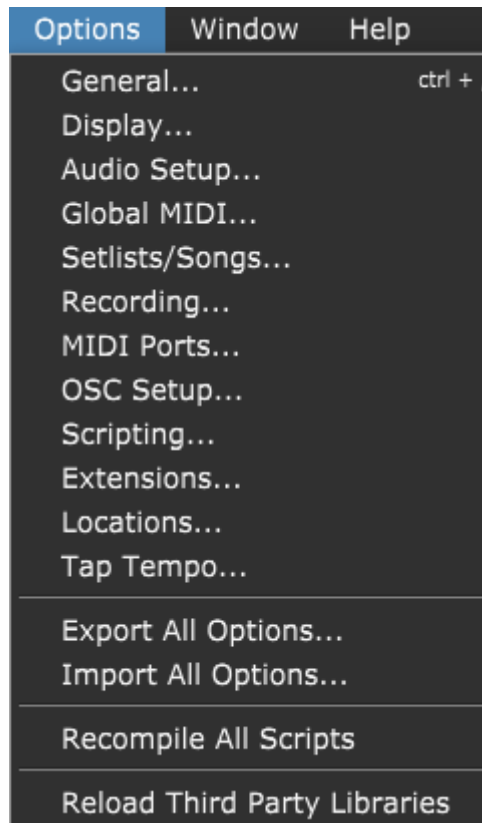
**WARNING: Turn your amplifier off to prevent loud noises from damaging your speakers or your ears while testing.**

**Note:** make sure you do not use any plugins or connections while measuring (or simply create a new empty gig before using this tool), and that your input signal is not too low.

Find out more about latency, sample rate and buffer size in our free e-book [The Ultimate Guide to Optimize your Windows PC for the Stage](#).

## 16 Options and preferences

In this chapter you will learn about all the options and preferences that you can adjust in Gig Performer. To open Options, simply click on the Options menu and select the desired menu item:



Selecting any menu item will open the Options window where you can also navigate through tabs to configure different options. Refer to the following subchapters for more information.

After you configure Gig Performer options and preferences to your taste, you can export them to a **gpsettings** file. Simply click on the **Export All Options...** menu item and save your gpsettings file; you can later import that file to any Gig Performer installation by clicking on the **Import All Options..** menu item and selecting the previously saved gpsettings file.

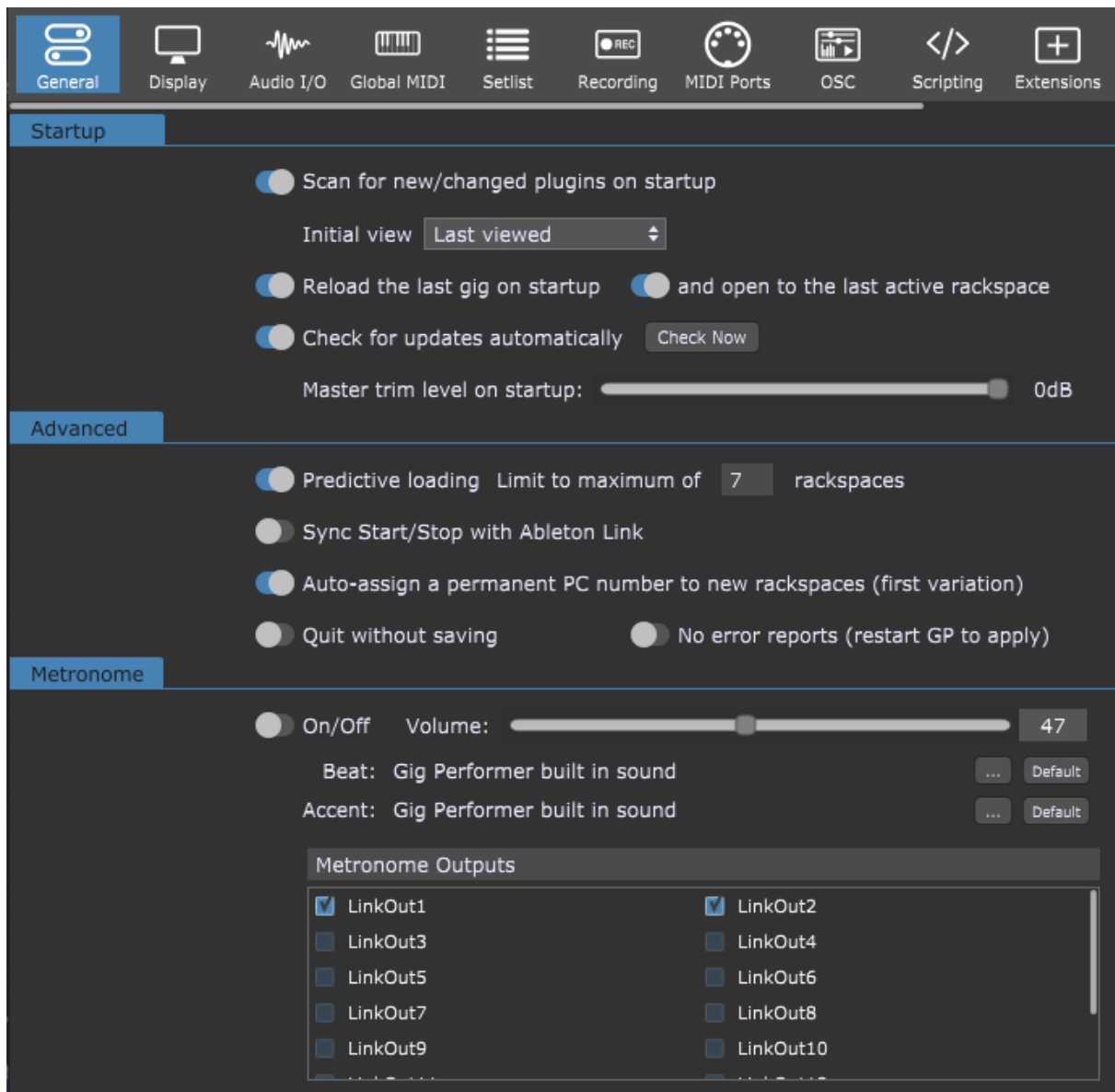
**Note:** a gpsettings file stores not only all settings in the Options screens, but also all global settings including global MIDI assignments, and the entire Rig Manager setup.

Visit [this chapter](#) to learn more about various Gig Performer file extensions and [this chapter](#) to learn more about the default file locations in Gig Performer.

**Tip:** Read the blog article [Gig Performer Maintenance: Save, export, and backup](#) to learn how to save and export various elements in Gig Performer, load and import them, and create a backup of your important files and the whole computer.

## 16.1 General

The **General** tab appears when the *General...* menu item is chosen in the Options menu, or when its keyboard shortcut is used (**Ctrl + ,** on Windows or **Cmd + ,** on Mac):





**Scan for new/changed plugins on startup** - when enabled, Gig Performer will check for new or updated plugins every time it starts. The best practice is to leave this enabled unless you're certain that your plugins have not been updated since you last used Gig Performer. If you're wondering why your latest plugin or update has not shown up in the plugin list, not scanning for plugins is the most probable cause (see the [Troubleshooting](#) chapter for more information).

**Initial view** - clicking on this drop-down list enables you to select the default view that shows up when you start Gig Performer. The **Last Viewed** menu item is the default (the other possibilities are the Panels view, the Wiring view, or the Setlists view).

**Reload the last gig on startup** - when enabled, Gig Performer will automatically load the last gig file that was used; you can also choose whether you want to open to the last active rackspace (default) or - if disabled - to load the first rackspace or open the template dialog on startup. **Note:** you can temporarily prevent Gig Performer from loading the last gig file by holding down the **Shift** key while starting Gig Performer and turning off the *Load last gig file* toggle button ([click here](#) for more information).

**Check for updates automatically** - unless you're actually on tour, it's recommended that you keep this option enabled so that you are informed of new updates. **Note:** updating any critical software while on tour or otherwise engaged in a production project is strongly discouraged until you have time to thoroughly test the update to ensure that it is functioning correctly. That applies not only to Gig Performer, but to any plugin and even other applications or operating system updates. We recommend that you do not update anything on your computer within 2 weeks of a show so that you have time to address any issues without stress.

**Check Now** - click on this button to manually check for updates. **Note:** this requires an Internet connection.

**Master trim level on startup** - the master trim is normally reset to 0db when you start Gig Performer. You can change that initial startup value by adjusting this slider.

**Predictive loading** - turn on this toggle button to activate the Predictive loading feature. Check the [Predictive loading](#) chapter to learn more.

**Limit to maximum of** - this field appears when the Predictive loading feature is enabled. Check the [Predictive loading](#) chapter to learn more.

**Sync Start/Stop with Ableton Link** - synchronizes start/stop with Ableton Link.

**Auto-assign a permanent PC number to new rackspaces (first variation)** - enable this option to auto-assign a new PC number to the first variation when creating a new

rackspace. **Note:** if explicit PC numbers are not used in variations the last selected variation will be called when using the PC for that rackspace (as when you click the rackspace with your mouse).

**Quit without saving** - enable this if you want to quit Gig Performer without prompting you to save any changes. This can be useful if you're running Gig Performer on a headless computer and just need it to shut down automatically at the end of a show.

**No error reports** - enable this if you don't want to send error reports to Gig Performer development team (not recommended). If you enable this toggle button, restart Gig Performer for changes to take effect.

**Metronome** - this complete section is covered in more detail in the [Audio metronome](#) chapter.

## 16.2 Display

The **Display** tab appears when the *Display...* menu item is chosen in the Options menu:



**Show plugin editor when inserted** - when checked, the plugin editor window will open automatically whenever a plugin is inserted. Disable this feature if you prefer to insert and route your plugin blocks first and edit them all later.

**Keep plugin windows on top** - when enabled, open plugin window editors always appear on top of other windows.

**Show information on plugin hover** - when enabled, moving the mouse over blocks in the Wiring view will cause a pop-up tooltip to appear with the information about the plugin (plugin's name, type, format and latency).

**Auto open/close plugin editors** - enabling this toggle button opens/closes plugin editors automatically as you switch from one rackspace to another. Note that this option doesn't open/close editor windows of all plugins in a rackspace, but only the currently opened ones.

**Touch friendly number editing** - if you run Gig Performer on a device such as Microsoft Surface, you may want to activate this option as it shows a large key pad for some dialogs and allows you to enter the desired values easier with your finger.

**Always start in full screen mode** - enable this option if you want to always start Gig Performer in full-screen mode. Press the **Esc** key or use the key combination **Ctrl+Alt+F** (or **Cmd+Option+F** on Mac) to exit the full-screen mode.

**Scroll bar width** - sets the scroll bar width of the Rackspace list, Songs list, Widget Properties area, the Plugin Manager, and the Wiring view. (from 6 to 50)

**Hide all tooltips** - when you move your mouse over some item (a button, a slider, or an option, for example), a little tooltip window will pop up to explain the purpose of that item. Once you become well acquainted with Gig Performer or if you find pop-up tooltips annoying, you can choose to prevent them from being displayed by checking this box.

**Note:** all tooltips are hidden except the one that shows up when you hover over the plugin block in the Wiring view and which can be disabled separately (see *Show information on plugin hover*).

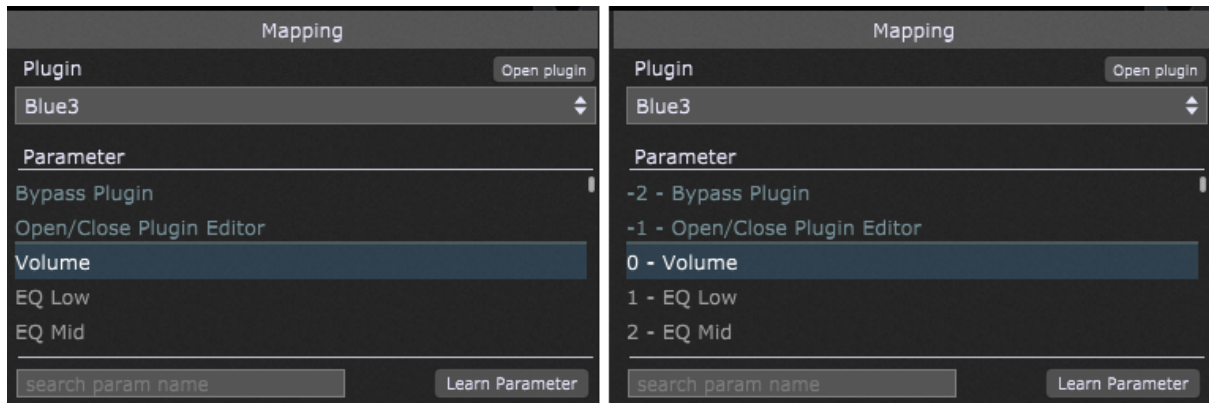
**Show after X milliseconds** - allows you to set the delay from 400 to 4000 milliseconds before a tooltip appears.

**Maximum number of audio channels to display in blocks** - this option prevents plugins from showing more than a specified number of audio channels. Some plugins have many audio channels, most of which may never be used, making their blocks unwieldy to view and causing them to take up a lot of horizontal space in the Wiring view. Note that the value entered here does not apply to the audio interface Audio In and Audio Out blocks. You can also configure the maximum number of audio channels available on individually inserted plugins (check [this chapter](#) to learn more).

**Name sizes** - in this section you can configure the name size of rackspaces, variations, songs, and song parts (from 12 to 48). Click the **Reset to default** button to reset all fields to 24.

**Show parameter numbers in widget parameter selection list** - when this toggle button is turned on, each parameter name is displayed prefixed with its unique parameter

number. This can be useful if you need to control parameters from [GPScript](#) or [OSC](#). In both of these cases, parameter numbers rather than names are used:



**Always show Widget Link and Radio Group labels** - when you gang multiple widgets together in a Widget Link group (lettered A through Z) or a Radio Group (numbered from 1 to 99), the corresponding character (a letter or a number) is displayed on the top left of the widget while in Edit mode. If you want the widget groups labels to stay always (i.e. even if you are not in Edit mode), turn on this toggle button.

**Always indicate locked widgets** - if this toggle button is turned on, the widget lock indicator will always be displayed (i.e. even if you are not in Edit mode).

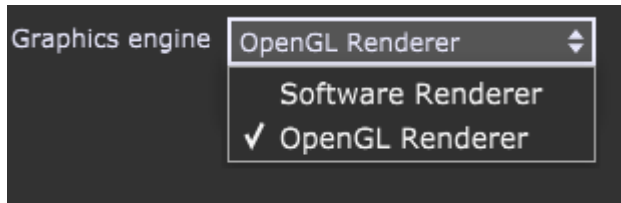
**Allow use of local fonts** - clicking this toggle button allows you to use other third-party fonts installed on your computer.

**Reset widget fonts** - click this button to reset all widget labels to the Gig Performer's default system font.

**Reset all warnings** - clicking this button will reset all "Don't show again" warnings in various dialogs.

**Show hints at startup** - by default, every time you create a new gig, a hint dialog pops up. You can browse through many useful hints and tips in that dialog. If you disable showing hints at startup, click this button to re-enable showing hints at startup.

**Graphics engine** - this section is available on Windows (macOS users always use the CoreGraphics renderer). Clicking on this dropdown menu allows you to select the graphics library that Gig Performer uses for its displays:



- **Software Renderer** - this choice means that Gig Performer uses your computer's CPU to render the display without the help of any graphics library optimizations. Windows users without high-performance graphics in their machines will likely get the best results with this setting.
- **OpenGL Renderer** - on systems with high-performance graphics capabilities, OpenGL may provide better visual quality, as such cards are often optimized to run OpenGL.

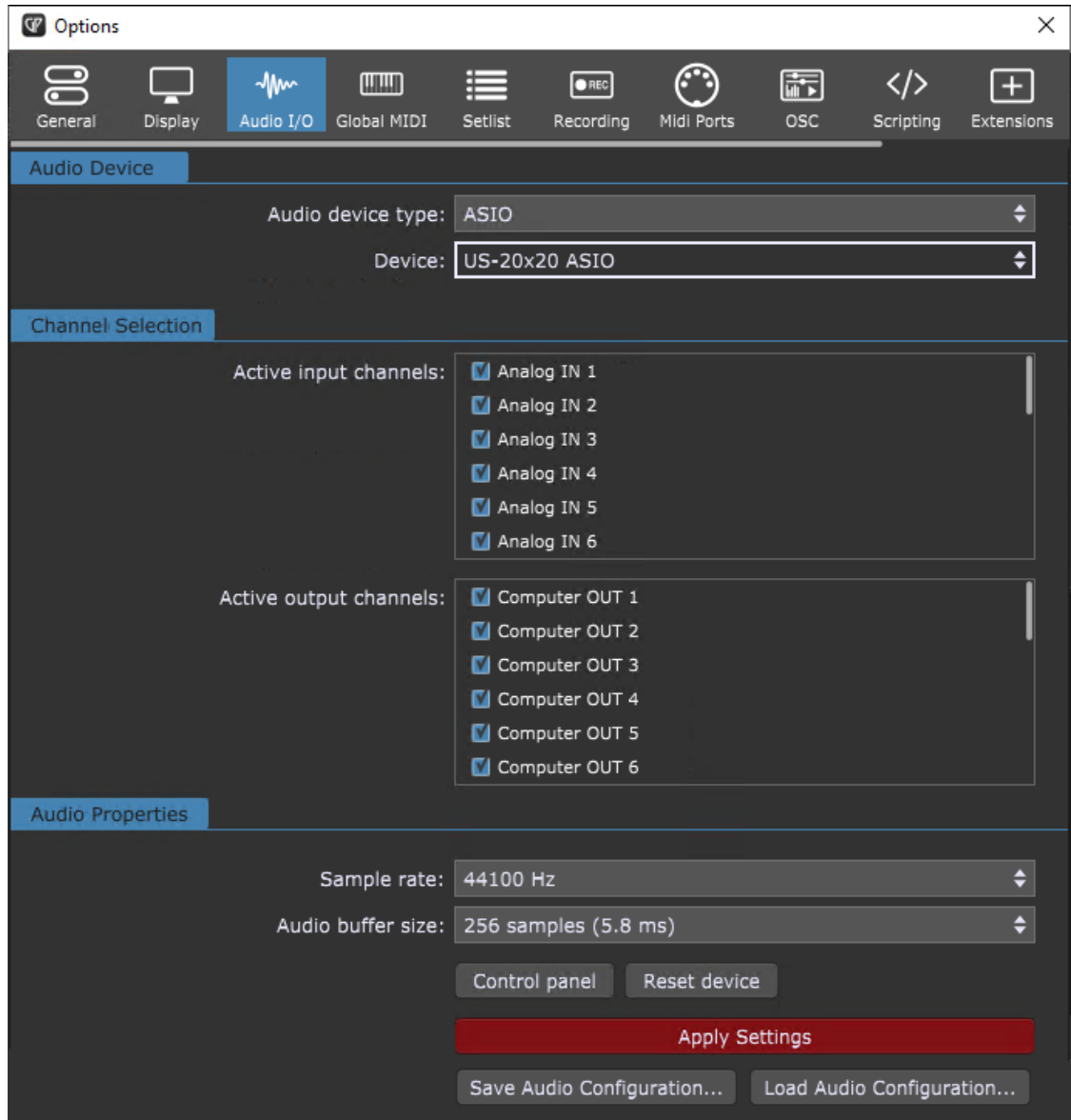
Depending on the graphics card in their machine, Windows users may find that OpenGL Renderer delivers better quality than Software Renderer.

**MIDI Monitor Colors...** - clicking this button allows you to set the colors for different MIDI messages.

### 16.3 Audio I/O

The **Audio I/O** tab appears when the *Audio Setup...* menu item is chosen in the Options menu; this window is slightly different on Windows and macOS:

**Windows:**



**MacOS:**



On macOS, you can separately select the input and output device and there is no device type selector as there is always just one type (Core Audio); on Windows, the audio device type selector is included, and one device must be used for both input and output.

**Audio device type** - for best results on Windows, make sure that you always select ASIO driver.

**Device** - this drop-down menu allows you to select one of the connected audio interfaces or an aggregate device (consisting of multiple interfaces) such as ASIO4ALL or ASIO Link Pro (if installed). On Mac this section is split into Input Device and Output Device (any aggregate devices created in the Audio MIDI Setup utility will appear here as options as well).



**Active input channels/Active output channels** - determines how many input and output channels are available to Gig Performer. If you use an audio interface with many channels, you may not want to access them all. For example, you may have an interface with 8 analog output channels and another 16 Lightpipe (ADAT optical) channels you're not using. So instead of displaying audio IN/OUT blocks with 24 or more channels, you can disable the ones you don't need by unchecking them in the list.

**Tip:** holding the **Shift** key down while clicking on an input or output checkbox will set all of them to the inverse of the one you selected.

**Sample rate** - the sample rate determines the quality of the audio signal. Typical values in the live world are 44.1kHz and 48kHz. Note that there is a tradeoff between higher sample rates, latency and CPU utilization. To view our website article about audio latency, buffer size, and sample rate, [click here](#).

**Audio buffer size** - the buffer size determines how often samples are actually processed. The smaller the buffer size (or the higher the sample rate), the lower (better) the latency, but at the cost of higher CPU utilization. Depending on your plugins and your computer, you may need to find a compromise that gives you the best tradeoff between CPU utilization and latency. If you start using too many CPU cycles, you will begin to hear clicks and other noise in the output. To view our website article about audio latency, buffer size, and sample rate, [click here](#).

**Apply settings** - this button will appear if you make any changes to the parameters in this dialog. Click on it to accept the changes, or simply close the Options dialog to maintain the current settings.

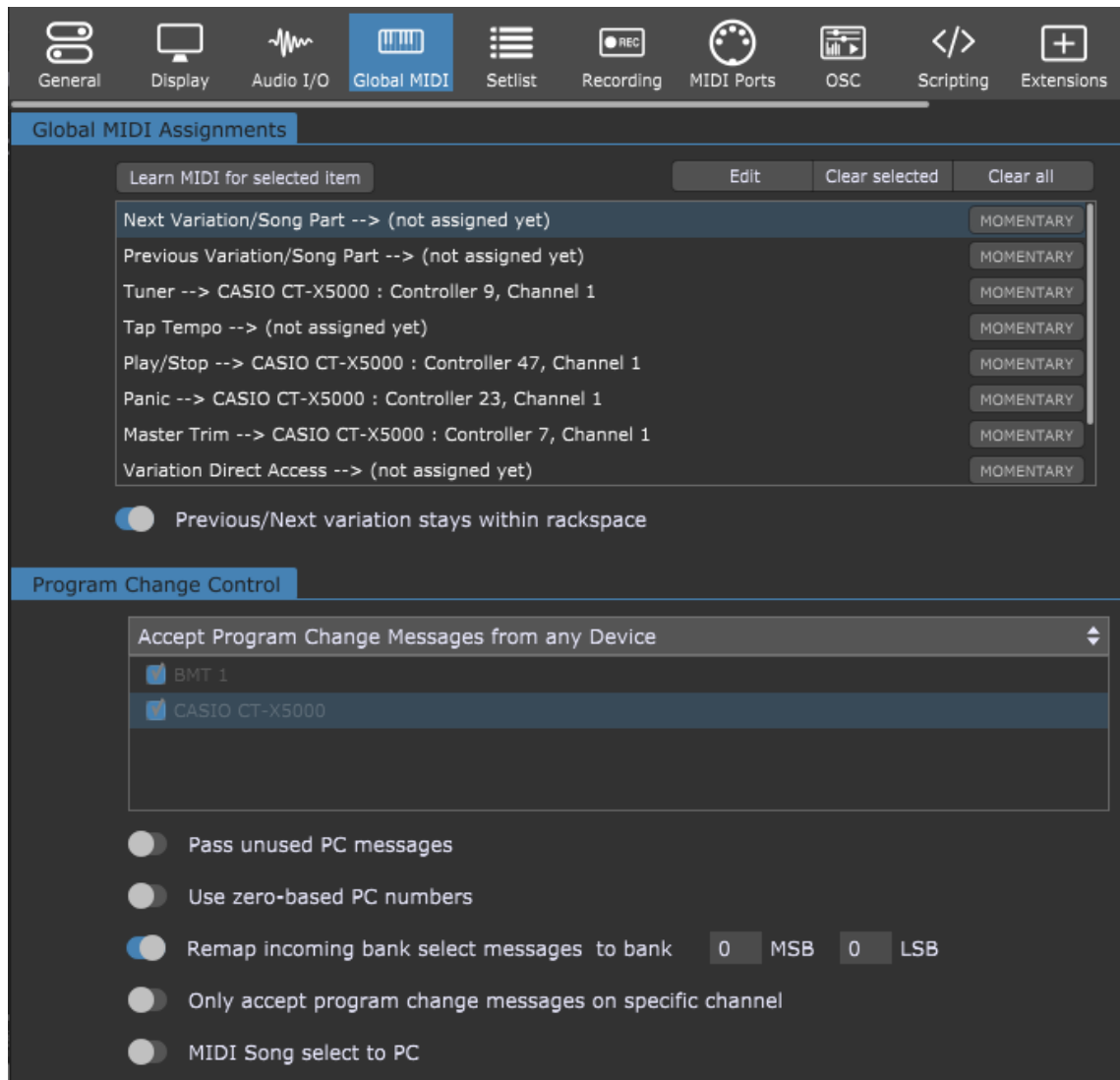
**Save Audio Configuration** and **Load Audio Configuration** buttons - Gig Performer allows you to save your audio configurations and load them back again. This feature is useful for supporting multiple audio configurations. Visit [this chapter](#) to learn more about various Gig Performer file extensions and [this chapter](#) to learn more about the default file locations in Gig Performer.

**Note:** changing the audio input or output device requires all plugins to be reset and so will cause your entire gig file to be reloaded immediately. Further, some plugins don't properly handle having their sample rate or buffer size changed after they have been loaded, and they may crash. Therefore, we recommend that you make such changes using an empty gig file. If you are changing your audio configuration (device, sample rate, buffer size, etc.), it can be useful to start Gig Performer with the **Shift** key held down, which opens the [Temporary Startup Options](#) dialog. Turning off the *Load last gig file* toggle button temporarily prevents Gig Performer from loading your previous gig file (this

applies if the *Reload last gig on startup* option is enabled in the [General Options](#)).

## 16.4 Global MIDI

The **Global MIDI** tab appears when the *Global MIDI...* menu item is chosen in the Options menu:



The Global MIDI Assignments section of this dialog is where Gig Performer can learn (or you can manually create/edit) MIDI messages, either from a continuous controller or a momentary switch (in which case you'll normally need to click on the accompanying "Momentary" button) to accomplish the following actions:

**Next Variation/Song Part** - moves to the next variation/song part in the current rackspace (or to the first variation of the next rackspace if you are at the last variation of the

current rackspace and the "Previous/Next variation stays within rackspace" toggle button is disabled). **Note:** if your variations or song parts are "skipping" (i.e. advance two steps), make sure to click the Momentary button (check out the tip number 5 from [this blog article](#)).

**Previous Variation/Song Part** - moves to the previous variation/song part in the current rackspace (or to the last variation of the previous rackspace if you are at the first variation of the current rackspace and the "Previous/Next variation stays within rackspace" toggle button is disabled).

**Tuner** - toggles the Tuner view.

**Tap Tempo** - Gig Performer can receive taps from a pedal or button on a connected MIDI controller and use that information to set the tempo of delay, echo, or looping plugins so that they can be synced to the tempo in which you are playing. To enter Tap Tempo, click on the **Learn MIDI for selected item** button and tap on the button or pedal you want to use to set the tempo (you will most likely need to enable the Momentary option as well).

Once you start tapping, the BPM display at the main toolbar will display the detected current tempo. All plugins that respond to the global tempo will now synchronize to this tempo (enabling "sync" to external tempo on individual plugins may be necessary. Please check your plugin documentation for more information). **Note:** the tempo used by connected plugins is always the global tempo unless it has been [overridden by individual rackspaces or songs](#).

**Play/Stop** - starts or stops the global playhead. This will typically start or stop the playback of audio files from the supplied [Audio File Player](#) plugin or from a third-party media player plugin.

**Panic** - generates "All Notes Off" messages on all MIDI channels, along with a Sustain Off message and a Center PitchBend message, thus stopping stuck notes in their tracks. This performs the same action as clicking on the MIDI panic button in the upper right-hand corner of the screen.

**Master Trim** - Gig Performer's master trim can be remotely controlled by a MIDI controller by clicking the **Learn MIDI for selected item** button and moving the physical control. Note: when controlling the Master Trim remotely, you cannot raise the value higher than 0dB.

**Variation Direct Access** - Direct access allows you to use a slider or knob to switch between variations in a rackspace. If you had a pedalboard with five MIDI switches, each switch would be set to the same channel and controller number, but assigned a different

control value. This value would select a variation in a rackspace, so if the five pedals were assigned values of 0, 1, 2, 3, and 4, they would recall the first five variations.

For example, if you assigned CC 42 to the "Variation Direct Access" item, then you could switch to 8th variation of the current rackspace by sending the CC number 42 and a value 7 (values go from 0 to 127). NB: this feature also works for song parts, i.e. if you're in the Setlist view, then the same message, CC 42 7, would switch to the 8th song part of the currently selected song.

**Previous Rackspace/Song** - moves to the previous rackspace/song. This function is useful when you wish only to move between rackspaces, not between variations in a rackspace.

**Next Rackspace/Song** - moves to the next rackspace/song. This function is useful when you wish only to move between rackspaces, not between variations in a rackspace.

**Note:** double-clicking a MIDI assignment puts that assignment in the Learn mode (you'll notice that the **Learn MIDI for selected item button** turns red). Double-clicking a MIDI assignment while holding the **Shift** key (or clicking the Edit button) opens the **Edit MIDI message** dialog where you can manually select the desired device and the desired MIDI message.

The only CC messages that Gig Performer cannot learn are CC 0 and CC 32 as these two messages are reserved for bank selection.

Other options:

**Previous/Next variation stays within rackspace** - if this option is enabled, previous/next messages are restricted to the current rackspace.

**Clear selected** - clicking on this button clears the selected MIDI assignment. **Note:** Holding the **Shift** key down while clicking on the **Learn MIDI for selected item button** also clears the selected MIDI assignment.

**Clear all** - clicking on this button clears all learned MIDI assignments.

The Program Change Control section of this dialog provides for the following options:

**Accept Program Change Messages from any Device** - determines whether Gig Performer should respond to MIDI program change messages from all external devices, or only from specific external devices.

**Pass unused PC messages** - when enabled, unused MIDI program change messages will be passed onto the active rackspace. By *unused* we mean that a particular program change message has not been associated with any rackspace or song part in Gig Performer. An unused PC message passed on to the rackspace is passed on to the MIDI In blocks as any other message. So any OMNI MIDI In blocks or MIDI In blocks for the device that generated the PC message will all receive the PC message and can pass it on to any plugins connected to them. If you are having trouble getting a particular program change message to arrive at a plugin, make sure that message is not already associated.

**Use zero-based PC numbers** - causes Gig Performer to display program change message numbers in the range 0-127 instead of 1-128.

**Remap incoming bank select messages** - enabling this toggle button causes Gig Performer to ignore incoming MIDI program bank select messages when selecting variations/song parts; note that also "to banks # MSB # LSB" shows up, enabling you to manually select the specified bank numbers that should always be used.

Note: When this toggle button is turned on, Gig Performer is not actually converting/remapping MIDI events (you can check this with MIDI monitors). This feature applies only in selecting variations/song parts and how Gig Performer interprets incoming bank select messages to select variations/song parts. Therefore, if you pass PC/bank messages to plugins these messages will not be modified in any way.

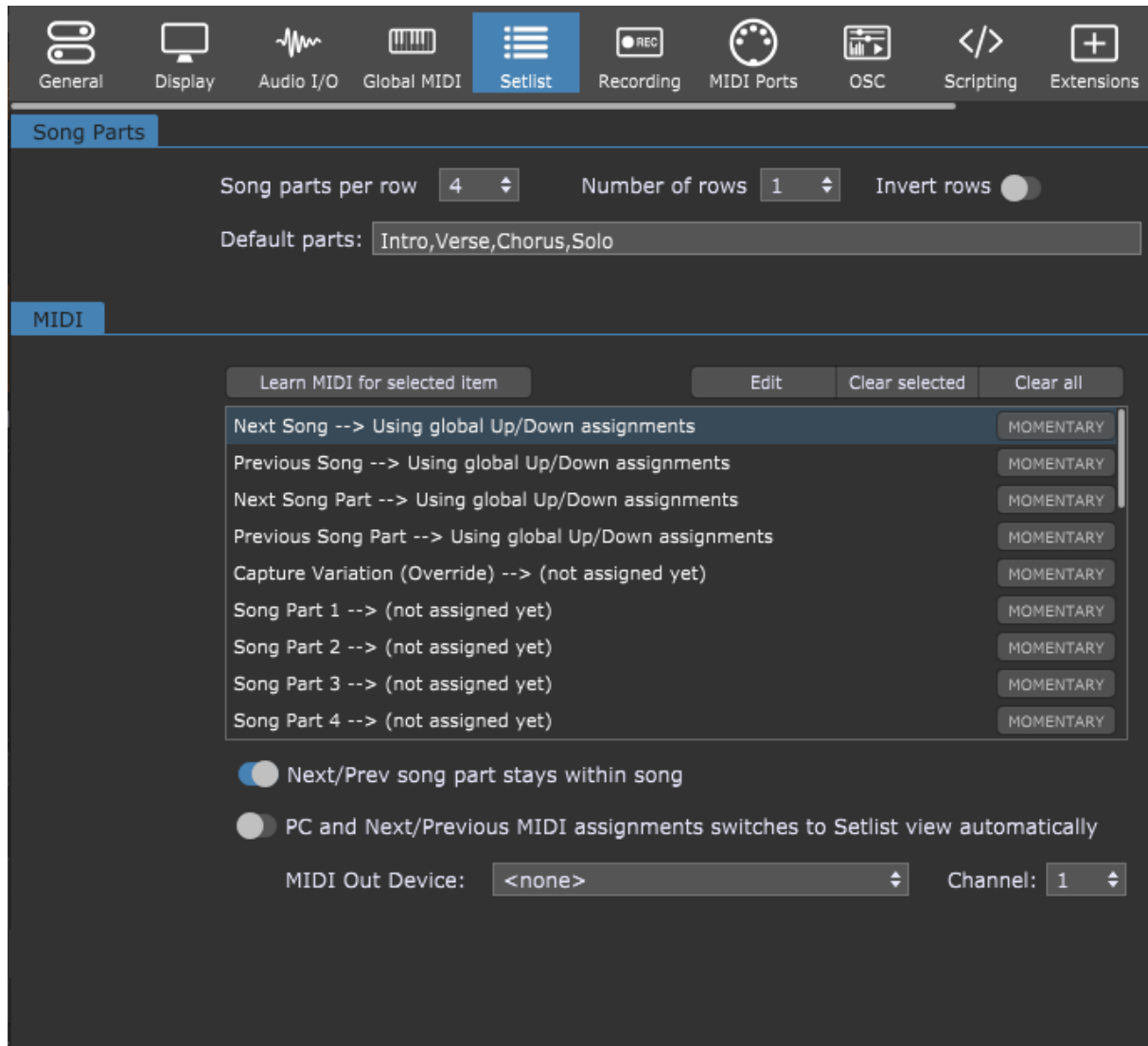
**Only accept program change messages on specific channel** - Gig Performer can change variations, rackspaces, and even instances in response to incoming MIDI program change messages. Enabling this toggle button limits Gig Performer to responding only to program change messages using the channel specified in the box.

**MIDI Song select to PC** - enable this toggle button if you want to interpret incoming MIDI song select messages as program change messages; note that also "and use MIDI channel #" shows up, enabling you to manually select the MIDI channel that you want to use when converting song select messages. This is because MIDI *Song Select* messages do not have channel numbers. This conversion is done at a very low level for efficiency purposes and so be aware that when this option is enabled, the [Global MIDI Monitor](#) will see incoming *Song Select* messages as program changes and display them as such.

**Note:** mappings to physical controllers made in the [Widget Properties inspector](#) override Global MIDI Assignments.

## 16.5 Setlists

The **Setlist** tab appears when the *Setlists/Songs...* menu item is chosen in the Options menu:



**Song parts per row** - choose a value (between 3 and 8) from the drop-down menu to set the number of tiles per row in the song part bar (see the [Setlist view](#) chapter for more information).

**Number of rows** - choose a value (between 0 and 3) from the drop-down menu to set the number of rows of tiles in the song part bar. Selecting 0 removes the song part bar.

**Invert rows** - flips the vertical order of the rows of tiles in the song part bar; instead of 1-2-3, when this toggle button is enabled the bar shows the rows 3-2-1.

**Default parts** - a semicolon or comma-separated list of song part names that will be added automatically when a new song is created. If no names are defined here, a single part called "Part 1" will appear instead.

**Learn MIDI for selected item button** - setlist operations (i.e. Next Song or Next Song Part) in the MIDI list can be mapped to MIDI controllers. Select the item in the list, click on this button, and use the MIDI controller you want mapped to it. Clicking on this button while holding the **Shift** key clears the selected assignment.

**Momentary button** - if the physical control you are mapping is a momentary switch (rather than a latching/toggle switch), click on this button.

**Setlist operation --> MIDI controls list** - this list shows setlist operations that can be externally controlled and the MIDI events they are mapped to. To map a control, select it in the list and click the **Learn MIDI for selected item** button. Double-clicking a setlist operation activates the Learn mode; Double-clicking a setlist operation while holding the **Shift** key (or clicking on the Edit button) opens the **Edit MIDI message** dialog where you can manually select the desired device and the desired MIDI message.

**Clear Selected button** - clicking this button removes the MIDI mapping from a selected setlist operation.

**Clear All button** - clicking this button removes MIDI mappings from all setlist operations.

**Next/Prev song part stays within song** - with this toggle button is enabled, the MIDI controller assigned to Next and Previous operations will never move to a different song. When the first song part in a song is reached, the Previous operation has no further effect; when the last song part is reached, the Next operation has no further effect. **Note:** this checkbox has no effect when using arrow keys on the computer keyboard to move between song parts; they will still move to the next or previous song.

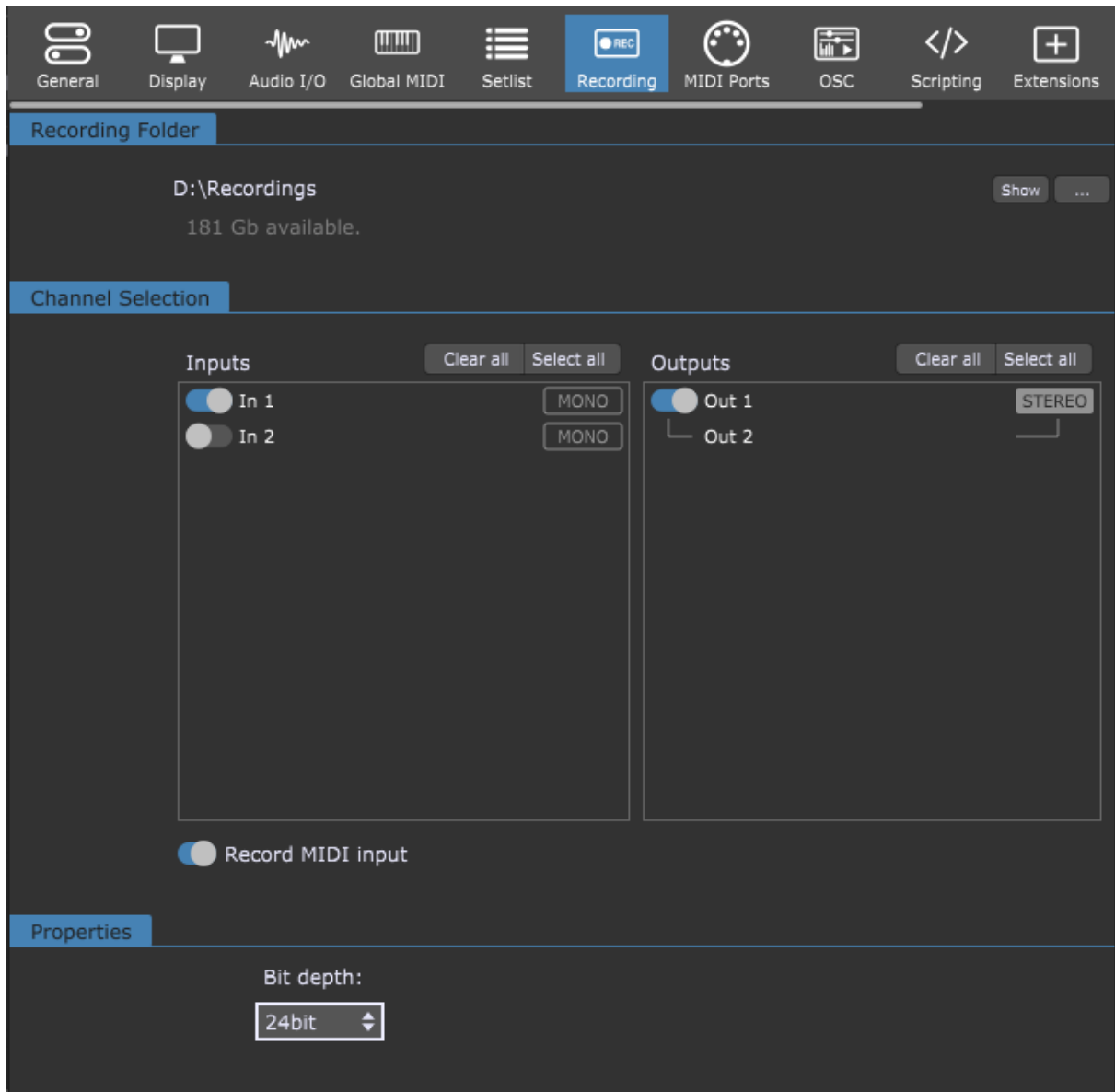
**PC and Next/Previous MIDI assignments switches to setlist view automatically** - when this toggle button is enabled, any incoming MIDI program change (PC) messages or Next/Previous Song Part or Song commands cause Gig Performer to switch to Setlist view immediately. **Note:** enabling this option will prevent you from switching your rackspace directly using MIDI controllers; if such a MIDI message arrives - Gig Performer will switch to the Setlist view automatically.

**MIDI Out Device and Channel** - these settings define a single MIDI Output device to where program change messages are sent on song changes. Choose a MIDI device from the drop-down menu and a corresponding Channel from its drop-down menu. If you need to send multiple MIDI messages to different physical devices or plugins via any MIDI In block when

you select a song part, you can do this from the [Song Part Properties](#) dialog.

## 16.6 Recording

The **Recording** tab appears when the *Recording...* menu item is chosen in the Options menu:

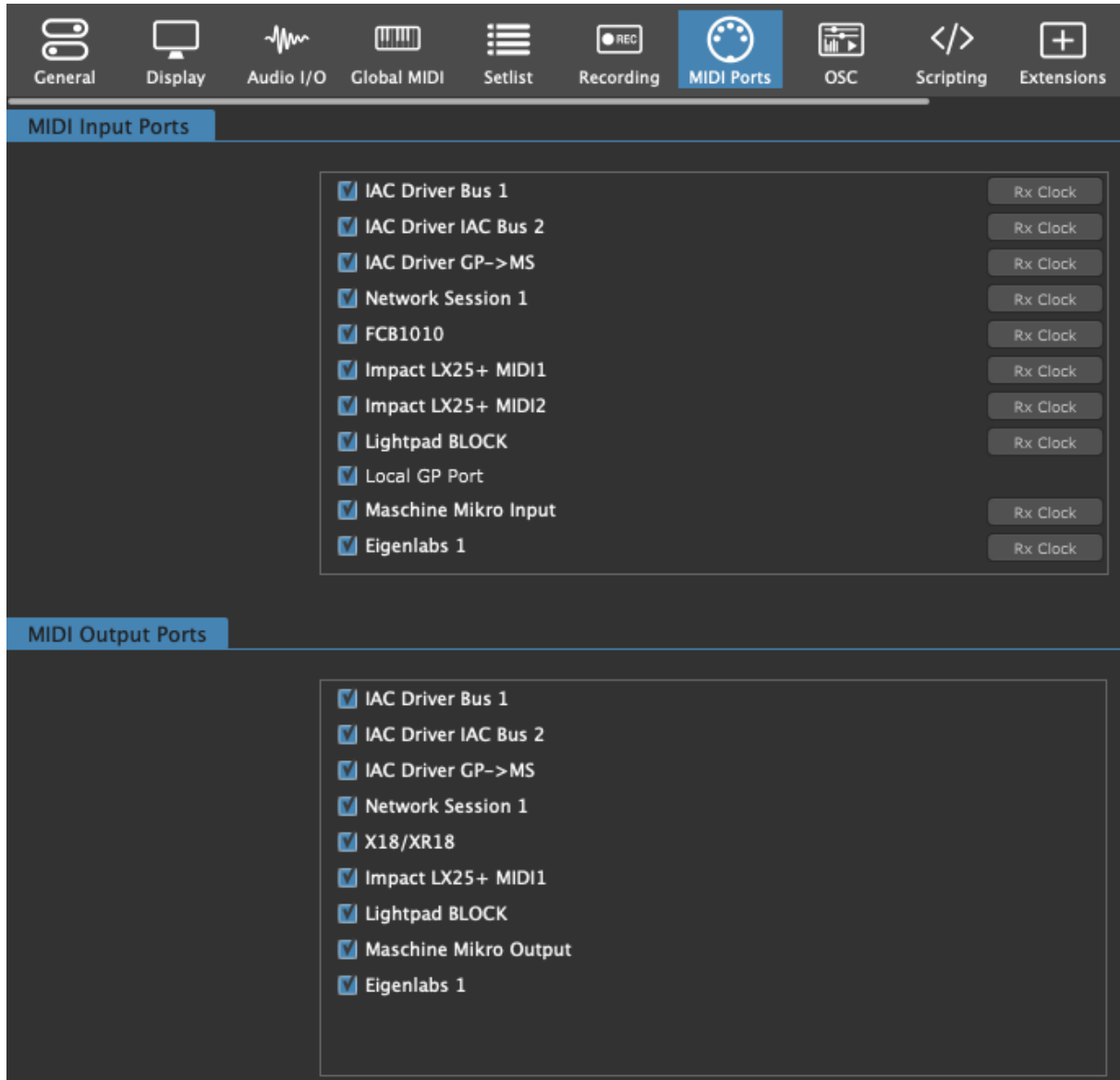


Details on this dialog are in the [Recorder](#) chapter.



## 16.7 MIDI Ports

The **MIDI Ports** tab appears when the *MIDI Ports...* menu item is chosen in the Options menu:



Normally, Gig Performer will open all MIDI input and output ports automatically. On macOS, this is never an issue. However, many Windows MIDI drivers do not allow multiple applications (and that includes multiple instances of Gig Performer if enabled) to access the same MIDI port. If you need to run some other application that needs access to a MIDI port, you will need to disable that port in Gig Performer so that it can be opened by the other application.

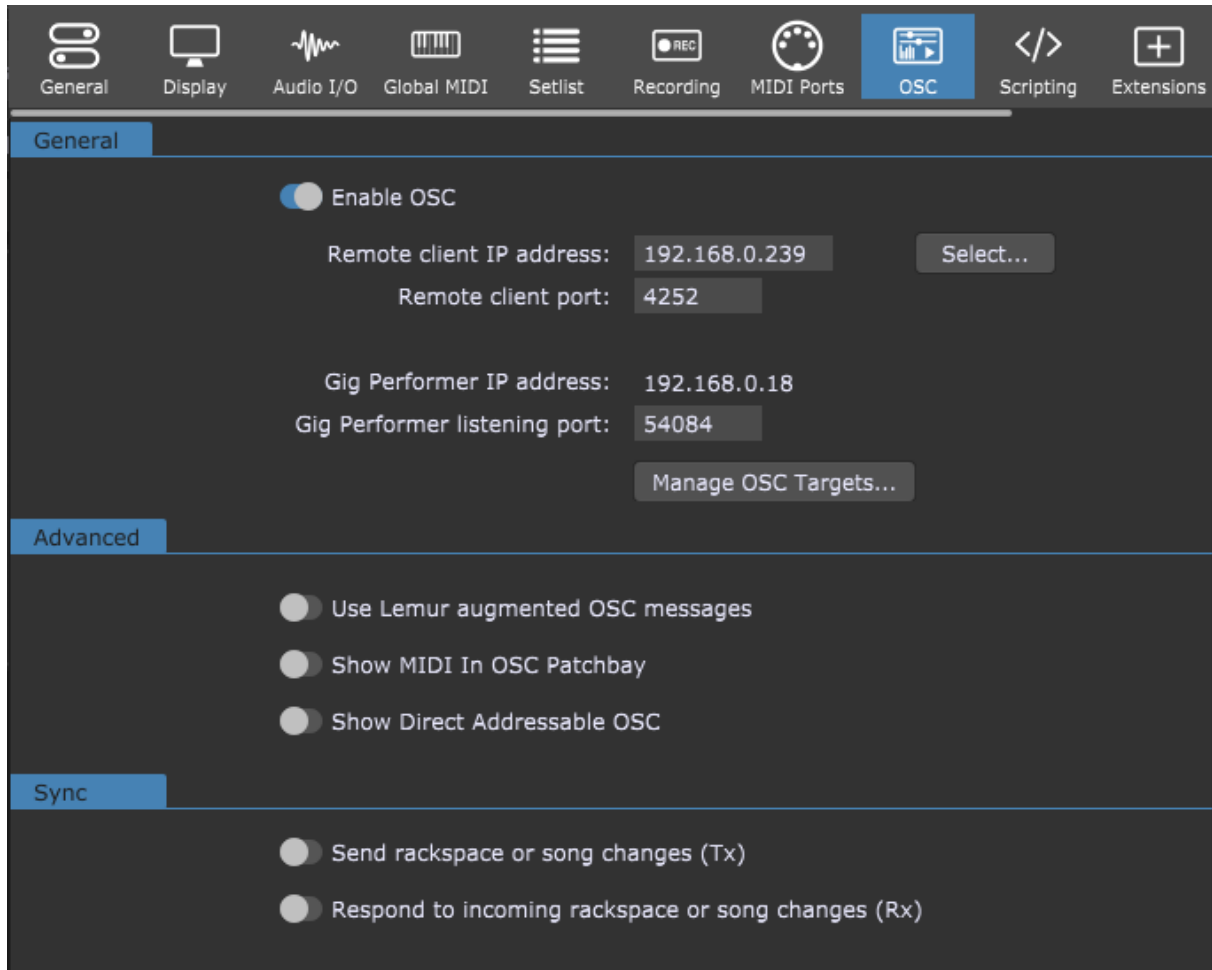
This also applies to special plugins inserted into Gig Performer, which themselves need to open a MIDI port to communicate with some specialized hardware.

In the MIDI Input Ports and MIDI Output Ports sections, simply uncheck the port or ports that you do not want Gig Performer to use. Normally, you would uncheck both the Input and Output ports but there may be special cases where you want to allow communication in one direction.

Note that every MIDI input port (except the special [Local GP Port](#)) has the **Rx Clock** button whose only purpose is to indicate whether the corresponding MIDI input port responds to incoming MIDI Clock messages; when the Rx Clock button is lit, that MIDI port responds to incoming MIDI Clock messages. The actual configuration is done through the [Rig Manager](#).

## 16.8 OSC

The **OSC** tab appears when the *OSC Setup...* menu item is chosen in the Options menu:

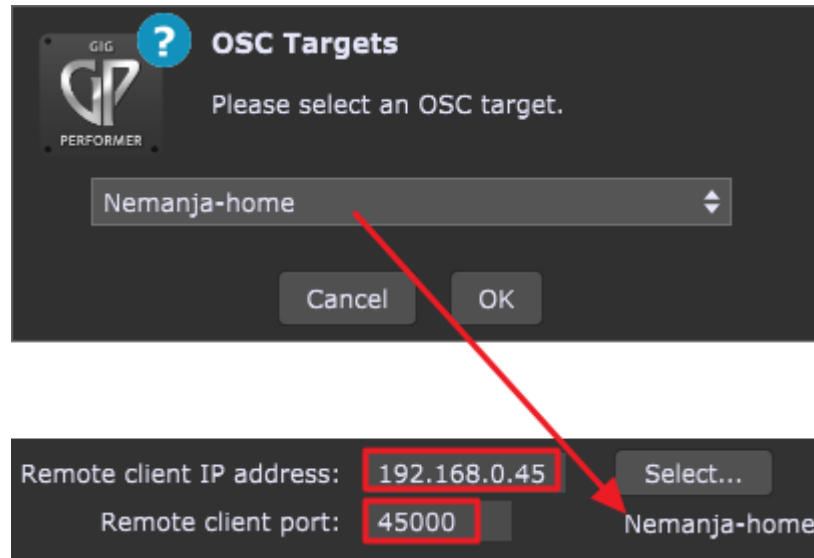


**Enable OSC** - click on this toggle button to turn on the OSC functionality.

**Remote client IP address** - the address of the remote device that is running an OSC application.

**Remote client port** - the port on the remote device that is listening for OSC messages over UDP (User Datagram Protocol).

**Select...** - instead of typing the Remote client IP address and port, you can click on this button to select one of the previously defined OSC targets:



Simply select the desired OSC target from the list and click on OK. Note that Remote client IP address (as indicated in the red rectangles) and port changed to match the selected target. Additionally, the name of the selected target will appear under the Select... button (as indicated with the red arrow).

**Gig Performer IP address** - the address of the computer on which is Gig Performer running. Configure your remote device to use this IP address to connect to Gig Performer.

**Gig Performer listening port** - the port on which Gig Performer is listening for incoming OSC messages over UDP (User Datagram Protocol).

**Manage OSC Targets...** - click on this button to create new OSC targets:

OSC Target Name	IP Address	Port
Nemanja-home	192.168.0.45	45000
		0
		0
		0
		0

Cancel OK

Type in the name of a target and then its IP address and port. Click on OK to save the changes. Afterward, the defined OSC targets will be available when you click on the

**Select...** button (not only in this window, but also in other locations that provide a selection of an OSC target).

**Use Lemur augmented OSC messages** - if you are using Lemur or another OSC application that supports scripting, enable this option to cause Gig Performer to transmit parameterized OSC messages. Disable this option if you are using TouchOSC or other applications that are not programmable.

**Show MIDI In OSC Patchbay** - this concept leverages applications like MaxMSP to modify incoming MIDI events. So, if the patchbay is enabled, an incoming MIDI message will not be sent out. Instead, it will be wrapped as an OSC message and sent to some remote listener (like MaxMSP, Ableton, or another OSC-supporting application). A remote listener application could then examine the incoming message, modify it and/or create new MIDI messages, which would then be wrapped up in OSC and sent back to the MIDI In Block patchbay in Gig Performer, which would then extract the MIDI from the OSC wrapper and send it on. Unless you have experience using and configuring OSC in other applications, you should not enable this feature as misconfiguration will cause incoming MIDI events to get lost.

**Show Direct Addressable OSC** - widget direct OSC enables support for the individual OSC devices which are bound to the specific IP address/port that can be set up for each widget. It is not bidirectional at this point. Refer to the [Widget Properties inspector](#) chapter for more details.

The **Sync** section is dedicated to synchronizing rackspace or song changes between [multiple instances](#) of Gig Performer. The requirements is that rackspaces and songs have the same names across instances.

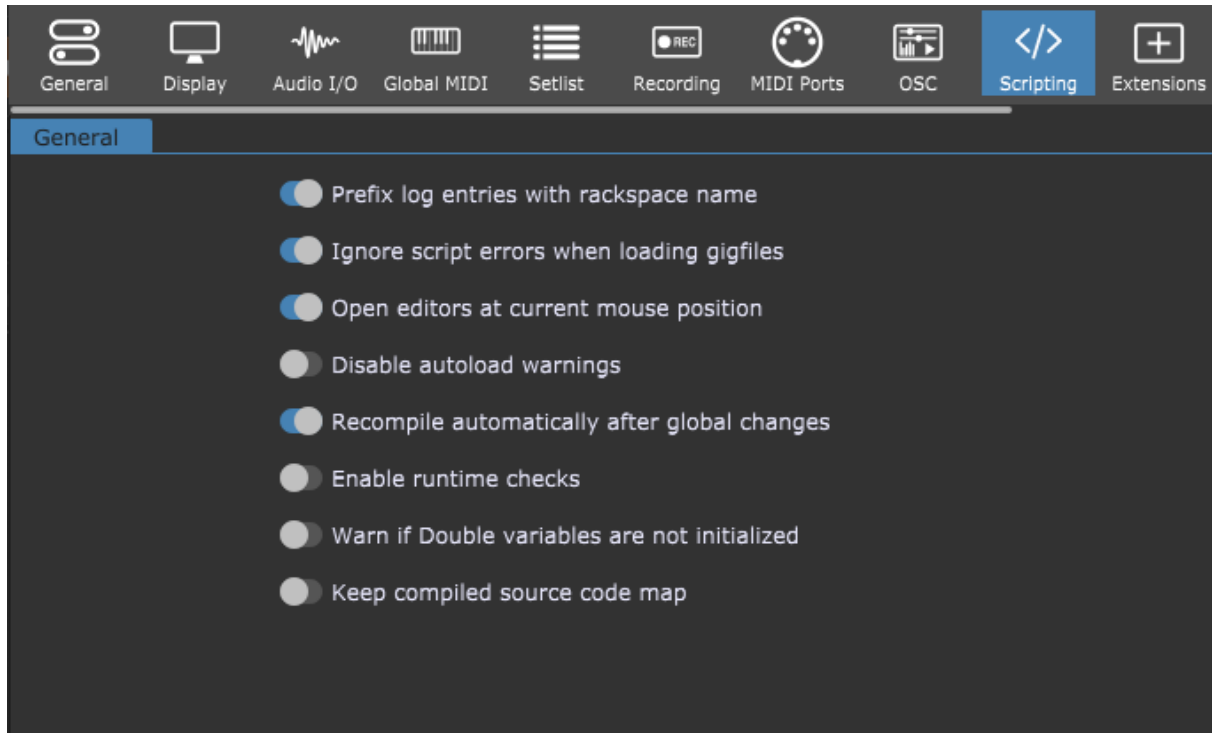
**Send rackspace or song changes (Tx)** - allows you to send messages to other instances of Gig Performer to switch rackspaces or songs.

**Respond to incoming rackspace or song changes (Rx)** - allows you to set this instance of Gig Performer to respond to incoming requests to change rackspaces or songs from another instance of Gig Performer.

Check out the [How to set up OSC](#) chapter to see an example of the sync setup.

## 16.9 Scripting

The **Scripting** tab appears when the *Scripting...* menu item is chosen in the Options menu:



**Prefix log entries with rackspace name** - enabling this toggle button causes messages in the Script Log window to be identified by the rackspace that generated them. Since there is only one log window for all rackspace scripts, troubleshooting is greatly aided when the log is formatted this way.

**Ignore script errors when loading gig files** - normally, attempting to load a rackspace with an invalid script triggers a warning dialog. Checking this option removes this warning. **CAUTION:** this is an expert function. Enabling this toggle button is not recommended unless you know exactly what you're doing, as a rackspace with a broken script may behave unexpectedly or fail to work at all.

**Open editors at current mouse position** - clicking on this toggle button centers the window of a script editor or the log window over the current mouse position when you open them (either by clicking the corresponding menu items or using their key combinations).

**Disable autoload warnings** - turning on this toggle button disables the warning when system scripts are not found.

**Recompile automatically after global changes** - turn on this toggle button to automatically recompile all scripts after changes are made in the [Rig Manager](#). **Note:** to manually recompile all scripts, click on the **Options** menu and then click the **Recompile All Scripts** menu item.

**Enable runtime checks** - adds additional safety by checking values. **Note:** this feature impacts performance and shouldn't be turned on unless needed; use it only while troubleshooting and then turn it off.

**Warn if Double variables are not initialized** - displays a warning if variables of type Double have not been initialized.

**Keep compiled source code map** - allows for easier errors troubleshooting by including a reference to an approximate line number of an error in the source code. Source code maps are stored in RAM, so turn this feature on only while debugging.

These menu items are related to script editing:

- File - Open Script From File (available when a Script Editor is opened)

- File - Open Recent Script File

- File - Save Script to File

- Edit - Script Helper

- Options - Recompile All Scripts

- Window - Current Rackspace Script Editor

- Window - Global Rackspace Script Editor

- Window - Song Script Editor

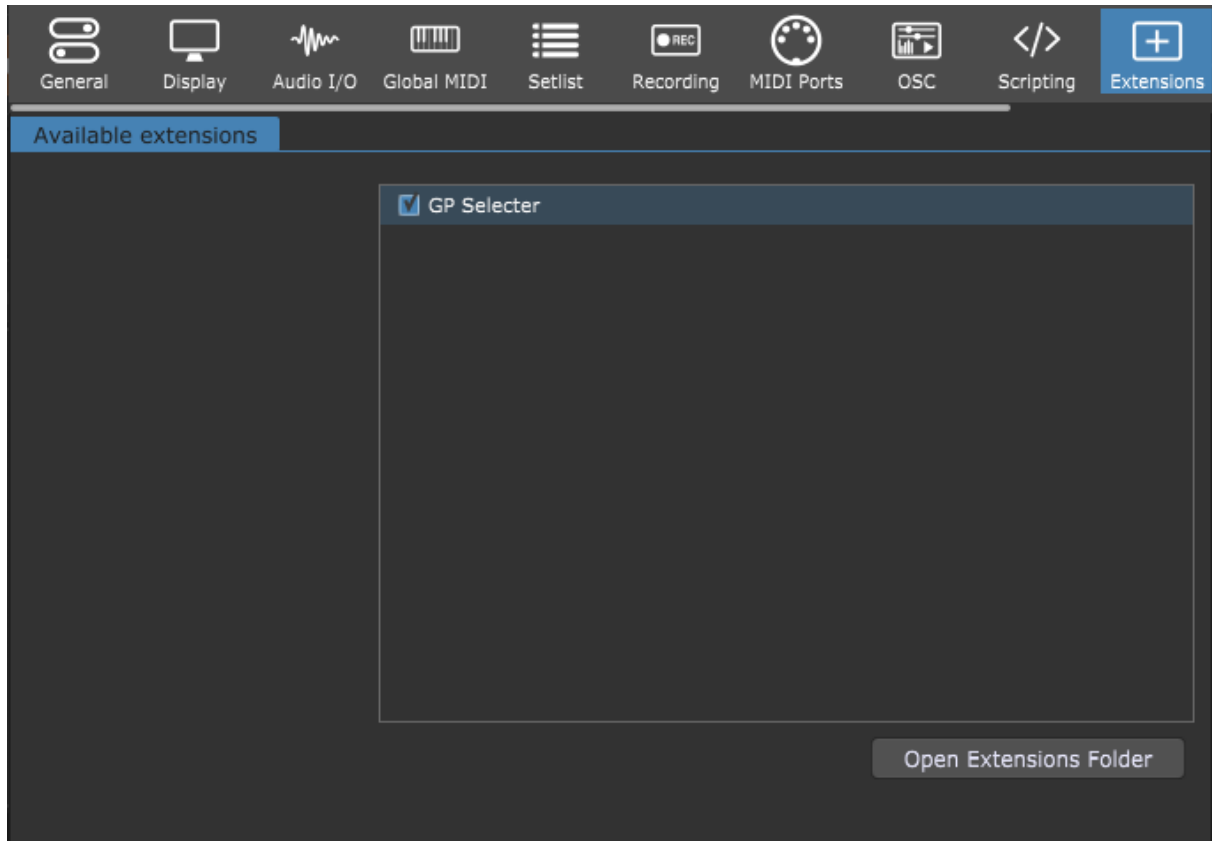
- Window - Gig Script Editor

- Window - Log Window

Gig Performer uses a language called GPScript, which was explicitly designed for Gig Performer. Refer to the [GPScript](#) chapter for more information.

## 16.10 Extensions

The **Extensions** tab appears when the *Extensions...* menu item is chosen in the Options menu:



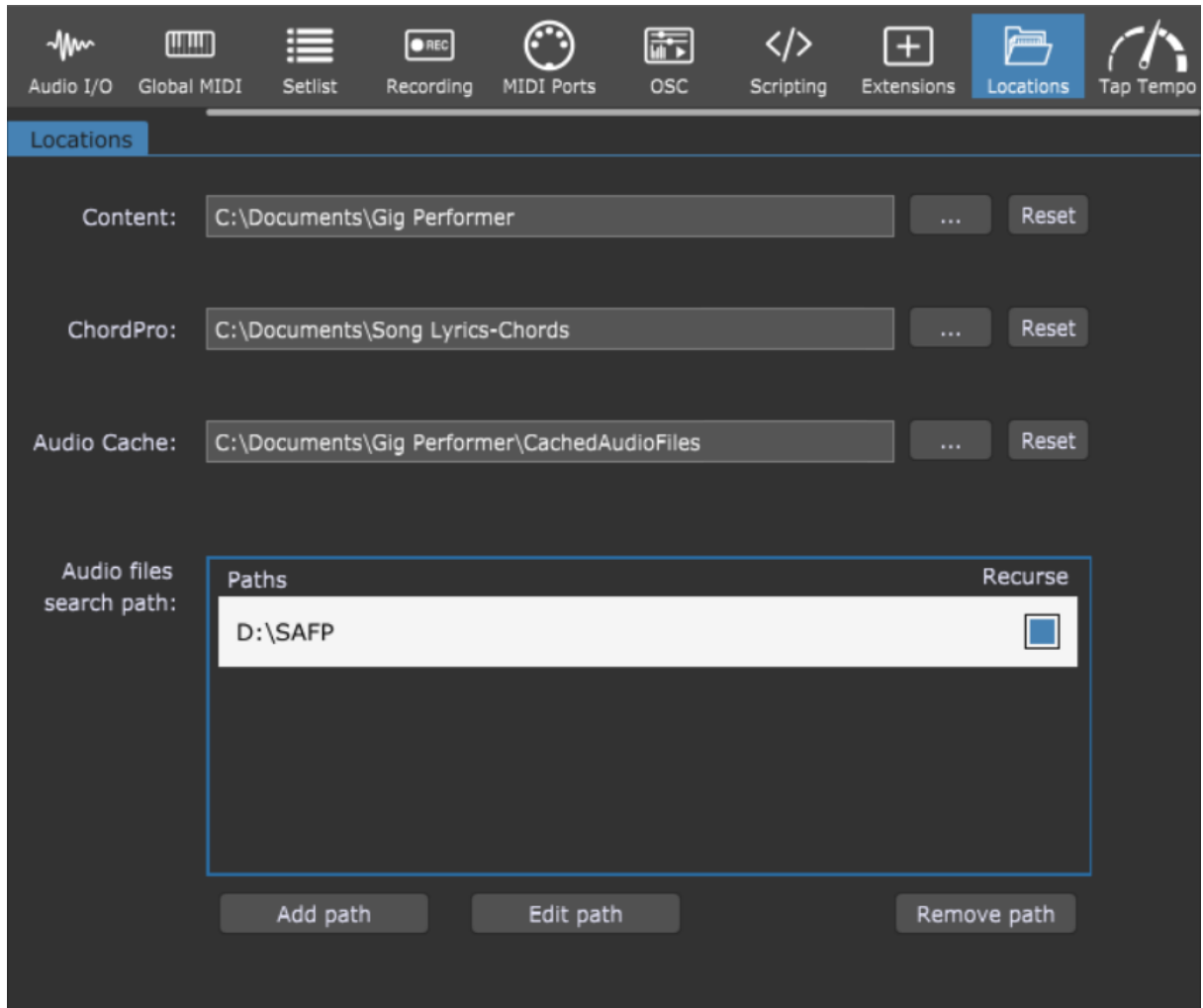
This window allows you to enable or disable Gig Performer extensions easily.

Refer to the [Extensions](#) chapter for more information.

## 16.11 Locations

The **Locations** tab appears when the *Locations...* menu item is chosen in the Options menu:





This screen allows you to change three different locations.

**Content** - this is the main (root) content folder of Gig Performer folder structure. In [this chapter](#) you can see the default file locations in Gig Performer, and this main folder is easily accessible through **Window** -> **Open Content Folders**.

**ChordPro** - changing the location of the root folder that contains ChordPro files is also supported. Note that *The Chords/Lyrics Editor* can only open ChordPro files in this folder. Refer to the [ChordPro](#) chapter for more information.

**Audio Cache** - this is the folder where your Streaming Audio File Player cached files are stored. If you change this location, you can delete the old location to save disk space.

**"..." button** - opens a dialog for locating the folder you want to designate as the root folder.

**Reset button** - reverts any user-defined folder path to the Gig Performer's default.

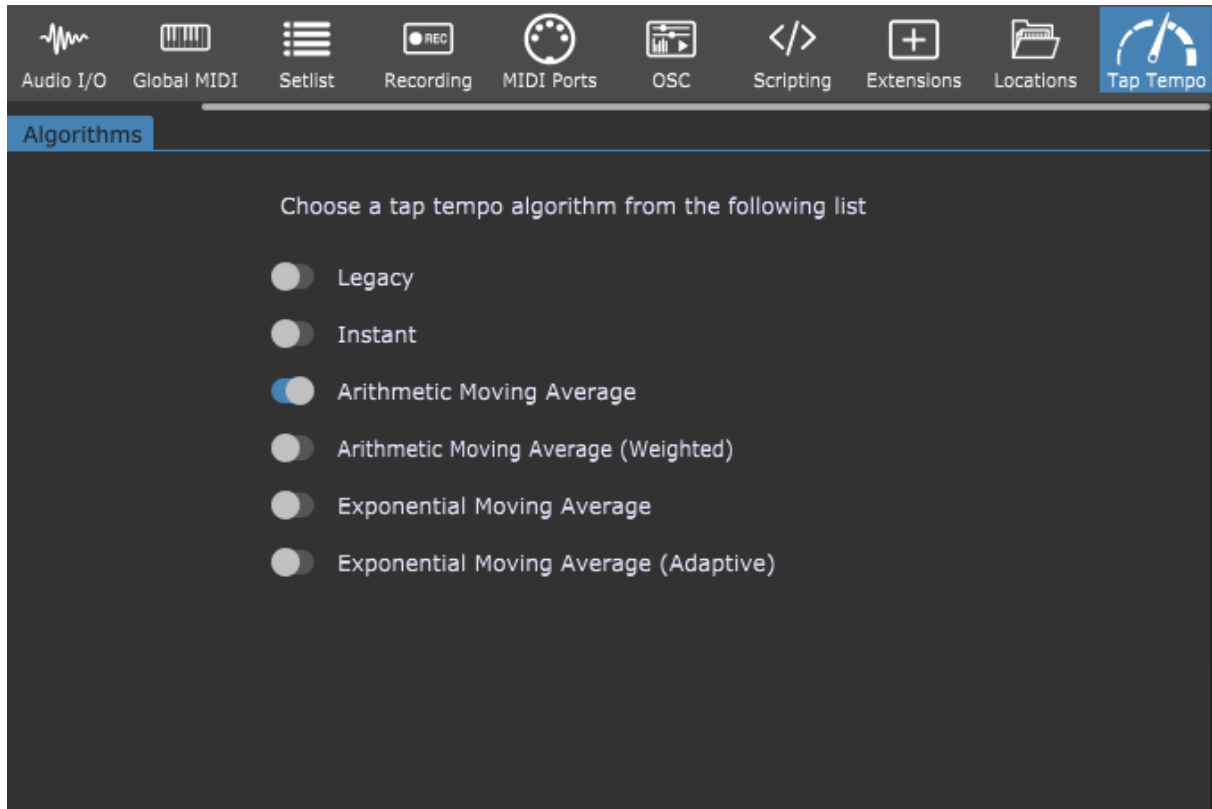
**Note:** Changing the default location does not move files from the corresponding sub-folders to the new ones. Certain features such as favorites, presets, and saved widget curves, will be unavailable until you relocate their files to the new location.

**Audio files search path** - in this list, you can specify search paths for your audio files for the [Streaming Audio File Player](#) plugin. If the Streaming Audio File Player can't load your audio files, it will search for these audio files in the specified folder. Check the **Recurse** check box if you want to also search all sub-folders under the specified folder.

NB: if the audio file is not found in the original location, Gig Performer will look in those locations for the audio file. If the audio file is found in the new location, the new file name (i.e. complete path) will replace what was previously stored in the Streaming Audio File Player configuration. Therefore, next time you load the gig file, it will first load audio files with that new path rather than searching again.

## 16.12 Tap Tempo

The **Tap Tempo** tab appears when the *Tap Tempo...* menu item is chosen in the Options menu:



You can select one of six tap tempo algorithms.

1. **Legacy** - this is the original algorithm used in Gig Performer 4.5 and older builds.
2. **Instant** - the tempo is determined by the last two taps.
3. **Arithmetic Moving Average** - the tempo is determined as the average of recent taps.
4. **Arithmetic Moving Average (Weighted)** - the tempo is determined as the average of recent taps, but it is biased towards more recent taps.
5. **Exponential Moving Average** - the tempo is determined as the average of recent taps, but it is biased exponentially towards more recent taps.
6. **Exponential Moving Average (Adaptive)** - the tempo is determined as the average of recent taps, but it is more responsive to sudden changes in tempo.

As for the new algorithms, the Instant algorithm is intended for very precise but fast detection and is mostly suitable when used mechanically. The other algorithms depend on how accurate one is with their tapping. Test these algorithms to determine which one works the best for you.

When you select the desired algorithm, check out the [Tempo and Time signature](#) chapter to see how to tap tempo.

## 17 Keyboard shortcuts

In this chapter find out more about keyboard shortcuts that you can use in Gig Performer. Besides the common keyboard shortcuts such as Ctrl+C, V, D (Cmd+C, V, D on Mac) for copy, paste, and duplicate respectively (works for plugins, panels, and widgets), check other useful shortcuts in the tables below.

### File management

MacOS	Windows	Function
Cmd+O	Ctrl+O	Opens a .gig file
Cmd+S	Ctrl+S	Saves the current .gig file
Shift+Cmd+S	Ctrl+Shift+S	Saves the current work to a .gig file with a new name
Shift+Cmd+I	Ctrl+Shift+I	Imports a single rackspace from a file
Shift+Cmd+E	Ctrl+Shift+E	Exports the current rackspace to a file

### Screen management

MacOS	Windows	Function
Cmd+T	Ctrl+T	Cycles between the Panels and Wiring views
Cmd+E	Ctrl+E	Toggles Panels view/Panels view Edit mode
Cmd+Shift+L	Ctrl+Shift+L	Activates the Setlist view
Cmd+Option+F	Ctrl+Alt+F	Toggles Full Screen
Cmd+,	Ctrl+,	Opens the General Options dialog
Shift+T	Shift+T	Activates the Tuner view

### Rackspace management

MacOS	Windows	Function
-------	---------	----------

Shift+Cmd+N	Ctrl+Shift+N	Adds a new rackspace
Shift+Cmd+D	Ctrl+Shift+D	Duplicates the selected rackspace
Shift+Cmd+Backspace	Ctrl+Shift+Backspace	Deletes the selected rackspace
Shift+Cmd+E	Ctrl+Shift+E	Exports the selected rackspace
Cmd+N	Ctrl+N	Adds a new variation to the selected rackspace
Cmd+Backspace	Ctrl+Backspace	Deletes the selected variation
Cmd+J	Ctrl+J	Opens the Quick Rackspace/Variation Finder
Cmd+G	Ctrl+G	Shows/Hides the Global rackspace

## Wiring and plugin management

MacOS	Windows	Function
Cmd+P	Ctrl+P	Opens the <i>Quick Plugin, Favorite or Preset Finder</i> (Gig Performer will switch to the Wiring View automatically if you are currently in the Panels View)
Shift+I	Shift+I	Zooms in the Wiring view surface
Shift+O	Shift+O	Zooms out the Wiring view surface
Shift+F	Shift+F	Zooms blocks to fit
Cmd+Shift+F	Ctrl+Shift+F	Fits blocks to the current view

## Song/Setlist management

MacOS	Windows	Function
Shift+Cmd+N	Ctrl+Shift+N	Adds a new song
Shift+Cmd+D	Ctrl+Shift+D	Duplicates the selected song
Shift+Cmd+Backspace	Ctrl+Shift+Backspace	Removes the selected song
Cmd+L	Ctrl+L	Locks/unlocks the selected song
Cmd+N	Ctrl+N	Adds a new song part to the selected song
Cmd+D	Ctrl+D	Duplicates the selected song part
Cmd+Backspace	Ctrl+Backspace	Removes the selected song part
Cmd+J	Ctrl+J	Opens a dialog that allows you to assign a rackspace/variation
Cmd+F	Ctrl+F	Opens the Quick Song Finder
Shift+Cmd+F	Cmd+Shift+F	Opens a dialog that allows you to add song(s) to the current setlist

### Gig Performer Window Management

MacOS	Windows	Function
Cmd+Shift+P	Ctrl+Shift+P	Opens the Plugin Manager
Cmd+R	Ctrl+R	Opens the Rig Manager
Cmd+M	Ctrl+M	Opens the Global MIDI Monitor
Cmd+H	Ctrl+H	Opens the Song Lyrics/Chord Editor
Cmd+Shift+1	Ctrl+Shift+1	Shows the current Rackspace script editor
Cmd+Shift+5	Ctrl+Shift+5	Shows the Global rackspace script editor
Cmd+Shift+2	Ctrl+Shift+2	Shows the Song script editor

Cmd+Shift+3	Ctrl+Shift+3	Shows the Gig script editor
Cmd+Shift+4	Ctrl+Shift+4	Shows the Log Window
Cmd+W	Ctrl+W	Closes the active plugin window
Cmd+Shift+W	Ctrl+Shift+W	Closes the active non-plugin window
Cmd+Option+W	Ctrl+Alt+W	Closes all plugin windows

To see in which dialogs the Undo feature is supported, check out the [Additional tips](#) chapter.



## 18 GPScript

Most users will find everything they need in Gig Performer's extensive feature set. Still, some users may require sophisticated custom functions that go beyond Gig Performer's main feature set. For those users, we have created a proprietary programming language called GPScript. GPScript enables you to perform incredible transformations and perform complex tasks. While any language requires some understanding of programming, GPScript is written to be accessible. If you can program spreadsheet macros or in a language such as VBA, you are likely to be comfortable with GPScript.

GPScript can be used to implement your own MIDI processing plugin filters using the [Scriptlet](#) plugin. While it has always been possible to create scripts for individual rackspaces, you can now create scripts at the song level and at the gig level. The Gig level script has some powerful features for efficiently rerouting incoming MIDI events, creating system-wide keyboard macro shortcuts and sending commands out to hardware when songs or rackspaces change.

Because GPScript will not be needed by most people and requires substantial documentation, we will not cover it in this manual in great detail. However, if you are interested, check out this simple example to get started: [How to control multiple parameters with a single widget](#).

Additionally, you can find an introduction to the GPScript language in this blog entry: [The GPScript programming language](#).

GPScript documentation can be found here (also accessible via the **Help** -> **Documentation**):

- [System function list](#)
- [Language manual](#)

For a discussion of GPScript and help getting it to do what you need, check out the [GPScript section](#) in the [Gig Performer Community](#).

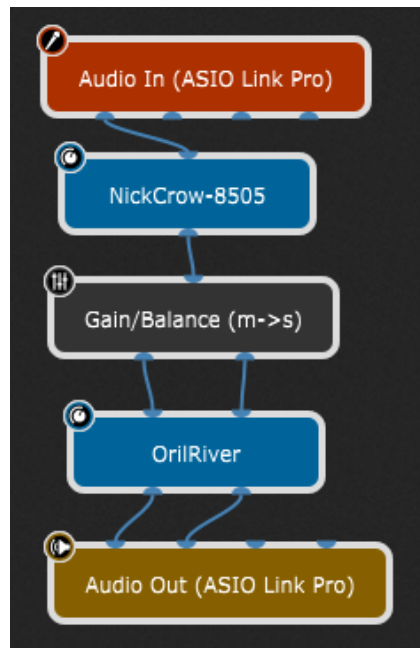
### 18.1 How to control multiple parameters with a single widget?

In Gig Performer, you can control multiple plugin parameters simultaneously:

- using [Widget Link groups](#),
- by assigning the same MIDI control to multiple widgets,
- using GPScript to control any number of parameters with a single widget.

In this chapter, we'll focus on a simple script allowing you to control multiple parameters with a single widget.

Start Gig Performer and create a new empty gig. First, click on the [Wiring view](#) and create a simple layout. In this example, besides built-in plugins, we are using two plugins - Nick Crow 8505 and OrilRiver:



Refer to [this blog article](#) for more information on how to get these plugins, insert them and connect them.

Let's say you want to control two parameters with a single widget: one parameter from the NickCrow plugin and the second parameter from the OrilRiver plugin. To use a plugin in scripting, you need to define a GPScript handle for it. Right-click on the NickCrow plugin to open its [contextual menu](#), then expand **OSC/GPScript Handle** and select **Set Handle...** :



**OSC/GPScript Handle name**

Please enter a new OSC/GPScript Handle name for this block

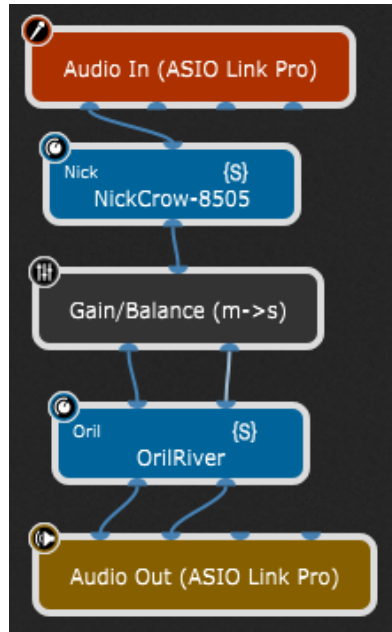
Nick

Enable OSC

Use in GPScript

OK Cancel

Type in a GPScript handle name (i.e. Nick) and check the **Use in GPScript** check box. Repeat the same steps for the OrilRiver plugin, set a handle name (i.e. Oril), and check the **Use in GPScript** check box. Now your layout looks like this:



The next step is to insert a widget and assign it a GPScript name so you can use it in scripting. Open the [Panels View](#) in Edit mode and insert the Plastic Blue Knob onto the panel (refer to [this chapter](#) for more information):



In the [Widget Properties Inspector](#), click on the Advanced tab and enter a GPScript name for this widget (i.e. Knob1).

The last information you need is a parameter number that you want to control with this widget. Click on the Wiring View and double-click on the Nick Crow plugin to open its plugin editor:



Move the desired control (i.e. the Gain knob) and you'll immediately see its parameter number (indicated with the red arrow). Repeat the same steps for the OrilRiver plugin and note the parameter number of, for example, the Room Size knob (indicated with the red arrow):

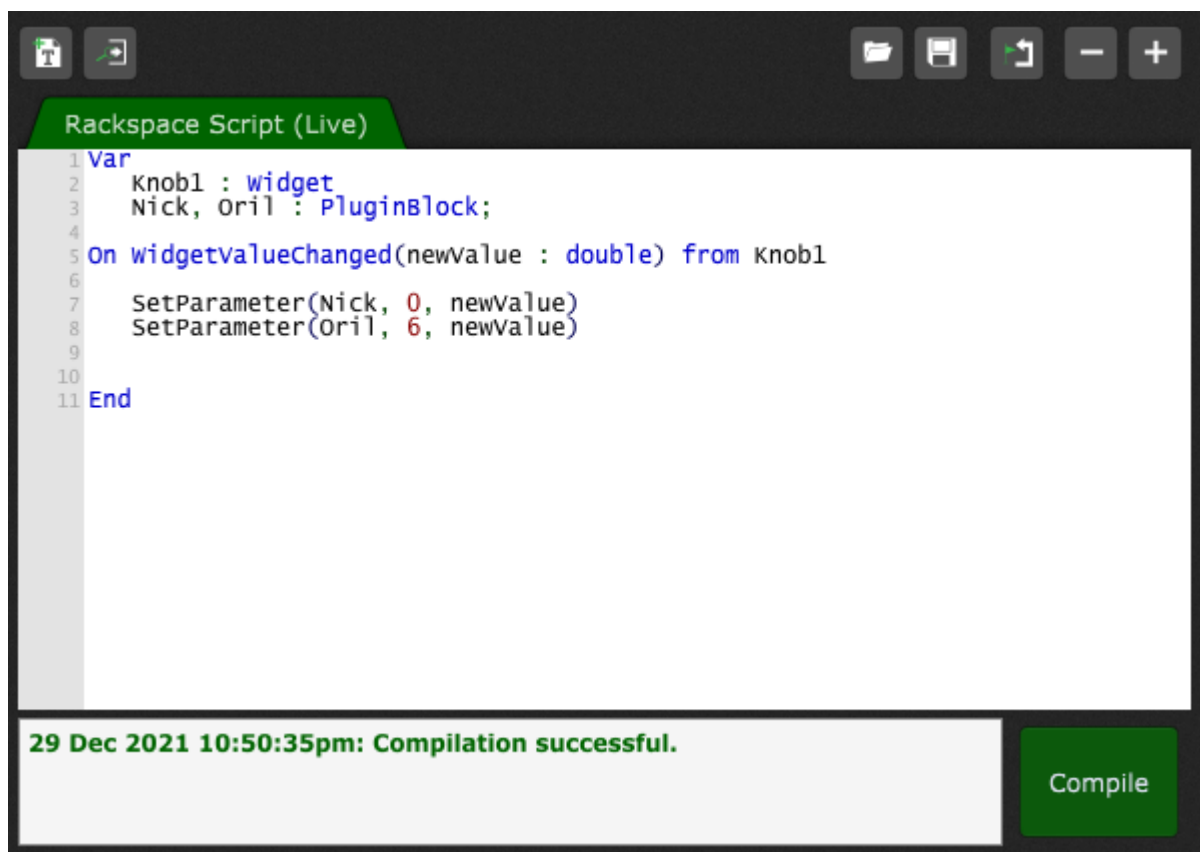


Click on the **Window** menu, select the **Show Current Rackspace Script Editor** menu item and insert this code:

```
Var
    Knob1 : Widget
```

```
Nick, Oril : PluginBlock;  
  
On WidgetValueChanged(newValue : double) from Knob1  
  
    SetParameter(Nick, 0, newValue)  
    SetParameter(Oril, 6, newValue)  
  
End
```

Now click on the **Compile** button:



If your script contains no errors, you'll receive a note that the compilation is successful.

Open the Panels view and move the widget; you'll notice how the chosen parameters change accordingly:



Let's take a look at the code.

In the **Var** section, you simply declare your variables based on the GPScript names you had previously given to them and specify their type (i.e. Knob1 : Widget or Nick, Oril : PluginBlock). If you want to control more parameters from different plugins, simply give those plugins the corresponding GPScript names and declare them in the Var section.

GPScript language is event-driven. The next section "On WidgetValueChanged(newValue : double) from Knob1" tells the compiler what event to monitor and on which widget. The **newValue** parameter is updated with the new value every time you move the knob (i.e. the widget value changes). Any function in GPScript that runs when a certain event occurs is called a callback (another example is *On Activate* which runs when a rackspace activates).

The section of code inside the callback simply tells the compiler what to do when the event is detected; in this example, we want to set the parameter in the plugin with the name "nick" and on the parameter number "0", with the value "newValue" (the current value of the knob widget).

This basic demonstration of GPScript provides you with virtually unlimited flexibility to create sophisticated setups for your live performance.

If you are interested in learning more, [join our Community](#) to see how users incorporate GPScript into their setups.

## 18.2 Local GP Port

The Local GP Port feature is a very advanced feature intended for the most sophisticated Gig Performer users.

The Local GP Port exists only inside Gig Performer. You can create MIDI In blocks that receive MIDI events from the Local GP Port, and widgets can be associated with MIDI events arriving at this port, just like any physical MIDI Input port. There is also a MIDI Out Local GP Port.

**Note:** this is not a virtual MIDI port visible to other applications. If you want to send audio to other DAWs or hosts, use a [GP Relayer](#) block.

The main purpose of this feature is to allow widgets to respond to incoming SysEx messages. GPScript lets you define a callback that will respond to incoming System Exclusive messages. If you use this callback, you can detect the changing bytes (one, two or more, depending on the particular message) from the SysEx message and generate standard MIDI channel messages out of them. If you inject such messages into the new Local GP Port, widgets can learn those messages and then control plugin parameters as usual.

**Note:** this is the reason the MIDI Monitor windows have the ability to display SysEx indices. It makes it easier for you to see which values in the incoming message need to be detected and captured.

MIDI events can be injected into the Local GP Port:

- from GPScript using the *InjectMidiEvent* function,
- or using a Local GP Port MIDI Out block.

To use this feature with GPScript:

1. Use GPScript (in a rackspace, scriptlet or the global Gig Script) to create a MIDI channel event (e.g. CC 42) and inject that message into the Local GP Port.
2. Open the Rig Manager and define a control that responds to that new CC event.
3. Have your widget learn the message in the usual fashion.

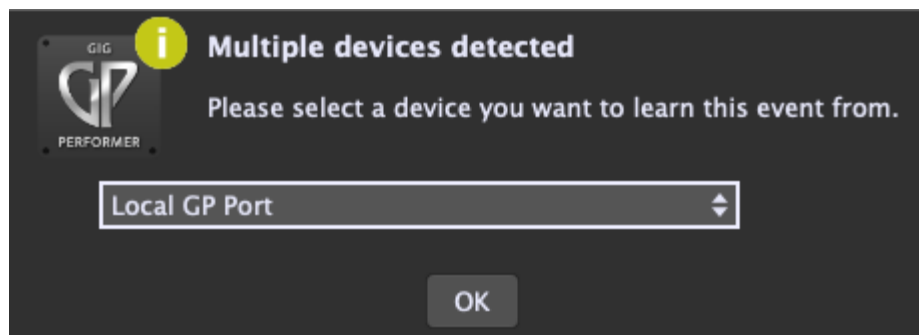
The MIDI Out (Local GP Port) is designed to receive messages from plugins such as the [MIDI File Player](#), [Envelope Follower](#) or other third-party plugins such as arpeggiators so that they can be mapped to widgets.

**Important:** do not connect any MIDI In (OMNI) device to this MIDI Out block to avoid creating a [MIDI feedback loop](#).

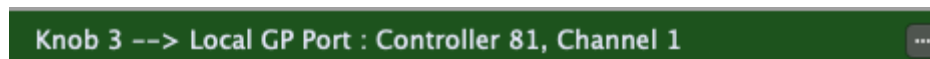
**Note:** if your trigger is another MIDI message coming from some other physical port - you will get a popup in the Rig Manager indicating that multiple devices were detected while learning your control:



Simply select the Local GP Port as your device:



The resulting entry in the Rig Manager:



Here is a simple script that injects everything that comes into a Scriplet plugin back into the Local GP Port:

```
On MidiEvent(m : MidiMessage)
    InjectMidiEvent("Local GP Port", m)
End
```



This script is used in [this blog article](#).

If you want to name the "Local GP Port" in the Rig Manager to e.g. GP1 - you can then use the following function:

```
InjectMidiEventViaRigManager("GP1", m)
```

Now simply drop a widget in and in the MIDI tab select the defined "Knob 3", for example.

## 19 Important workflow tools and tips

In this chapter, you will learn more about workflow tools and tips to help you get things done more quickly.

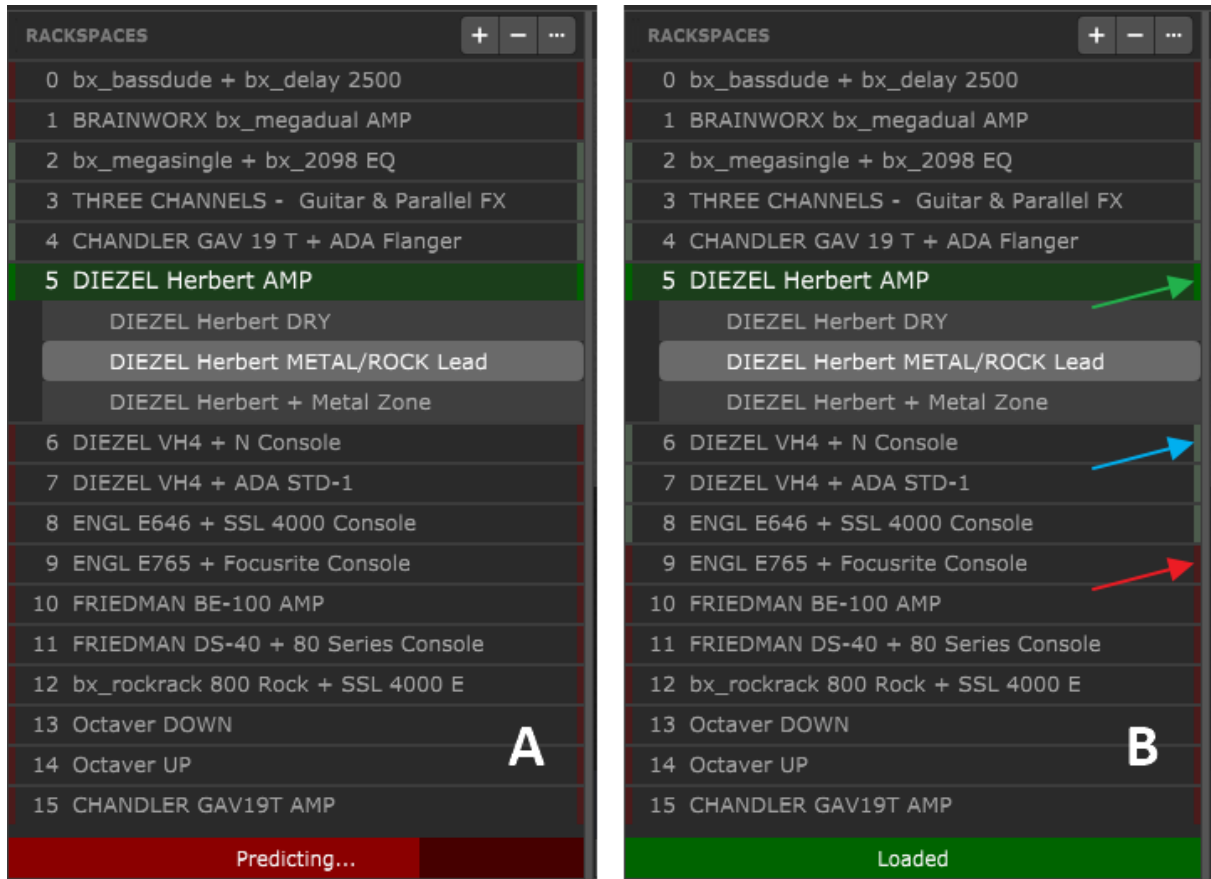
### 19.1 Predictive loading

One of the biggest problems for live performers is how much memory and CPU cycles are required to keep all the desired plugins available. In particular, plugins that use a significant number of samples can use many gigabytes of RAM to hold the required data.

In normal operation, Gig Performer loads all plugins for all rackspace and does some optimization to reduce the impact of hosting a large number of plugins simultaneously while still allowing you to switch quickly from one rackspace to another. However, this comes at the cost of RAM and CPU usage, which may create issues in systems with limited resources and/or significant audio hosting requirements. The possible alternative - loading just one rackspace (i.e. one set of plugins) at a time - would reduce RAM and CPU requirements considerably, but at the cost of waiting while each new rackspace is loaded.

Gig Performer's Predictive loading feature provides a creative solution: preload only those rackspace that will be needed in the immediate future (e.g. while playing the current song). Other rackspace are not loaded and will, therefore, not consume any memory. As an added benefit, startup time is significantly shortened since not all plugins need to be loaded. You may also see some reduced overall CPU utilization.

To activate this feature, open the [General Options](#) and turn on the **Predictive loading** toggle button. When this feature is activated, the **Limit to maximum of** field appears; here specify the maximum number of rackspace that will be pre-loaded around the currently selected rackspace; you can select an odd value between 1 and 49. For example, if you set **1**, then only the selected rackspace will be loaded; If you set **7**, that means that the current rackspace as well as the next three and previous three rackspace are also loaded and instantly accessible:



In the example above, each time you click on a rackspace the *Predicting...* status appears, as indicated with **(A)**; when seven rackspaces are loaded (the current rackspace plus three rackspaces before and three rackspaces after) the status changes to *Loaded*, as indicated with **(B)**. The current rackspace is indicated with the green arrow, predicted rackspaces are indicated with the blue arrow, and rackspaces that are not currently loaded are indicated with the red arrow.

The optimal use of this feature assumes you will be moving from your current rackspace to rackspaces that are close to it (within two, if the default value of 5 is being used). As you move to a new rackspace, Gig Performer will then load and unload other rackspaces so that there are always two available before and two after where you are. This happens in the background and does not affect your audio.

When using setlists, the Predictive loading feature prioritizes what gets loaded. The current song is always fully loaded, after that are loaded, in order: the next song, previous song, and any further songs until the limit imposed by the *Limit to maximum* setting is reached.

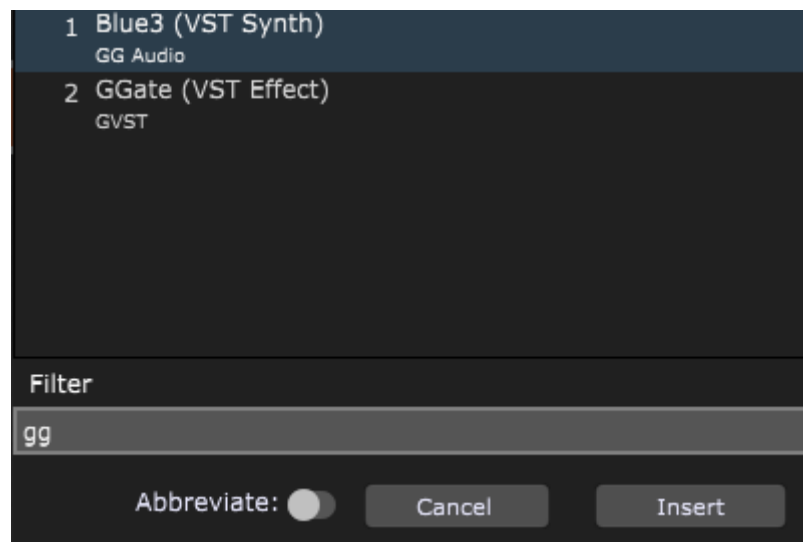
Predictive loading provides glitch-free instant switching among those rackspaces that are currently loaded, but with the tradeoff that you might not always be able to jump to an arbitrary rackspace without a short delay, depending on the particular plugin requirements. As a result, when using this option, you should configure your rackspaces in such a way that they are in rough order of when they will be needed in a performance, i.e. song intro -> first verse -> first chorus, etc. In that way, you can best ensure that you will never have to "jump" to another rackspace that is farther away than the specified number of loaded rackspaces (defined with the *Limit to maximum* field).

**Note:** rackspaces can easily be reordered by dragging them up and down using your mouse or trackpad; holding down the **Shift** key while doing so results in the dragged rackspace not being loaded, making it more efficient to do setlist reordering without having to wait for each rackspace to load.

More details about Predictive loading can be found on the Gig Performer website: [click here](#).

## 19.2 Quick Plugin, Favorite or Preset Finder

The **Ctrl+P** (or **Cmd+P** on Mac) keyboard shortcut provides a more efficient way to create a plugin than right-clicking, since it opens the **Quick Plugin, Favorite or Preset Finder** dialog (shown below), which allows you to type a partial plugin or company name, with the resultant list restricted to those that match:



Type in a partial string, such as "gg" to quickly find all plugins containing that name or their developers name. You can also type in e.g. "synth" to display all your synth plugins.

You can enter multiple substrings. For example, if you have the Arturia Modeling Collection installed, you could type "art dx" to restrict the list to the Arturia DX7 plugins. Using the substring "presets" or "favorites" will restrict the view to presets and favorites that you have created.

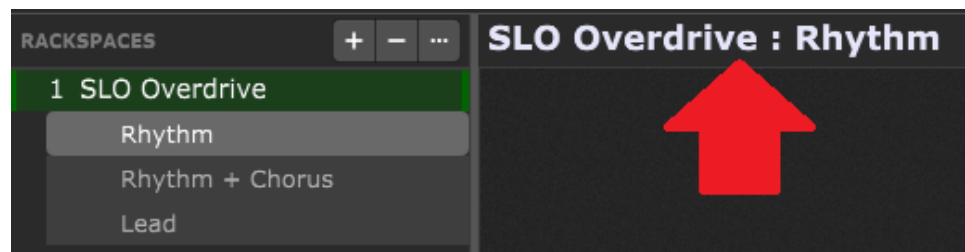
Turning on the **Abbreviate** toggle button removes some text from the view, such as favorites and presets, making the results easier to read.

Click on the **Insert** button to insert the selected plugin.

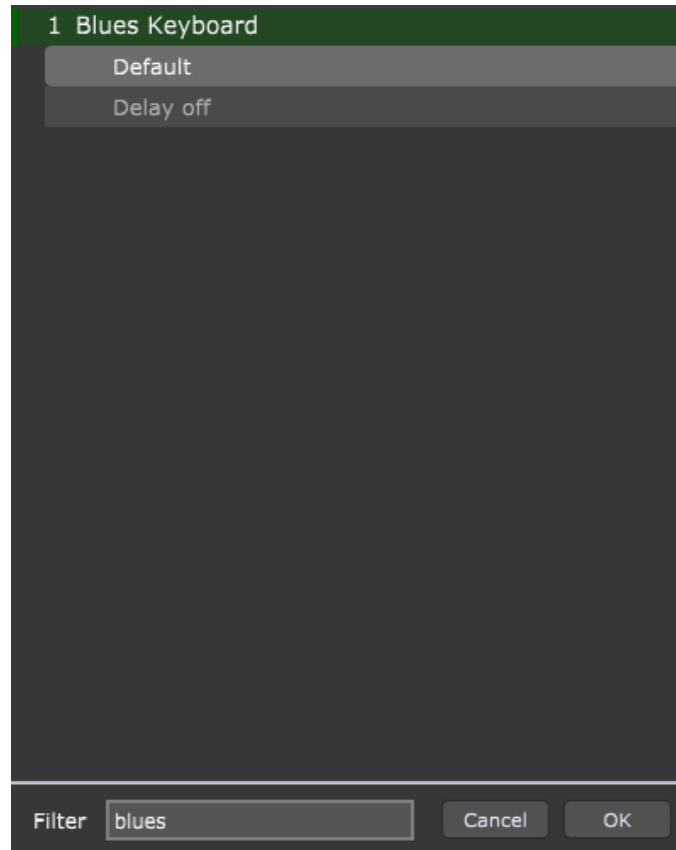
### 19.3 Quick Rackspace/Variation finder

The Quick Rackspace/Variation Finder is a fast way to switch to a specific rackspace and variation, great for jamming and improv performances or assembling setlists. It can be opened by using any of these four methods:

1. Double-click on the rackspace name:



2. Using the **Ctrl + J** (or **Cmd + J**) key combination,
3. Edit menu -> Quick Jump to Rackspace/Variation...
4. [More... button](#) -> Quick Jump to Rackspace/Variation...

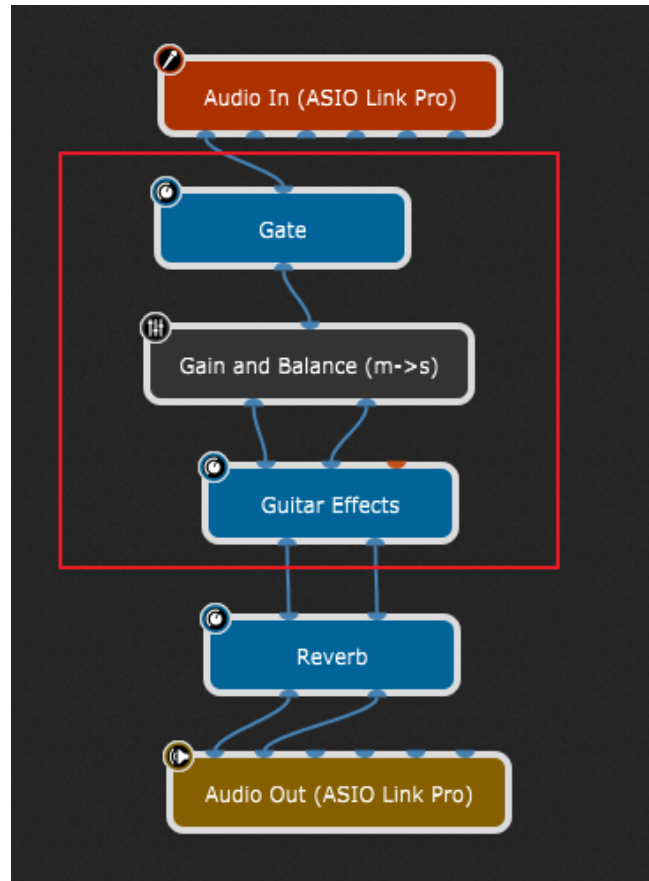


To switch to a different rackspace, open the Quick Jump to Rackspace/Variation... window, then click the variation you want to select, and click the OK button (or simply double-click the variation in the list).

Search terms can be entered in the Filter field to help find the variation you want, as indicated in the screenshot above.

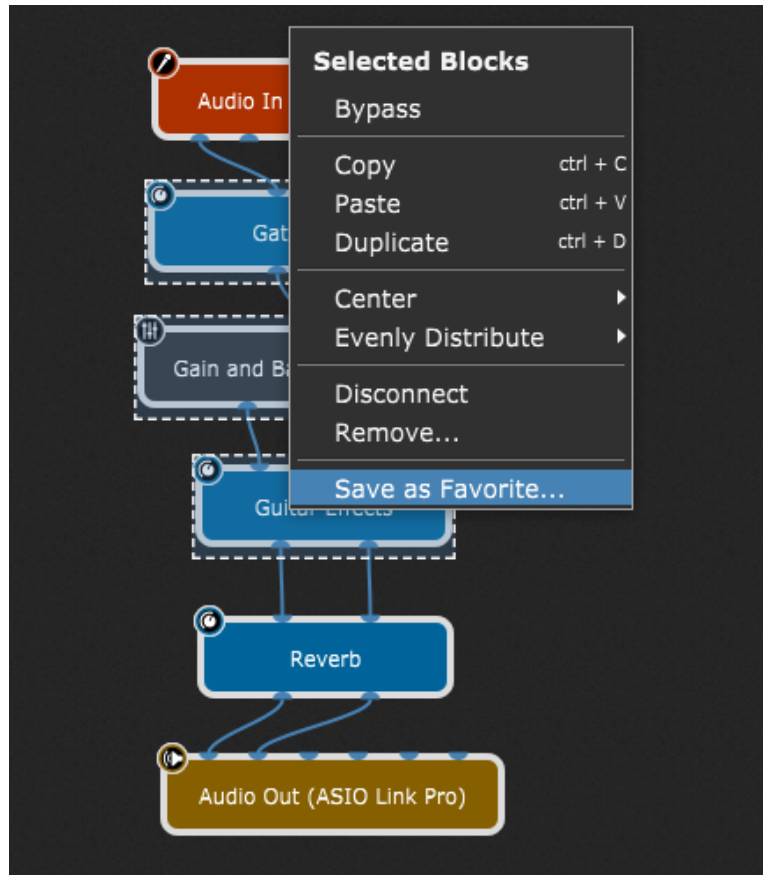
## 19.4 Save one or more plugins as favorites

Gig Performer enables you to select a plugin block or more plugin blocks and save the selection as a favorite. Let's take a look at this example below:



In this simple guitar layout, the audio signal first goes through the noise gate and is converted to a stereo signal, which is then forwarded to the guitar effects plugin. Suppose you want to re-use these three plugin blocks (indicated with the red rectangle) in other rackspaces or in new gig files.

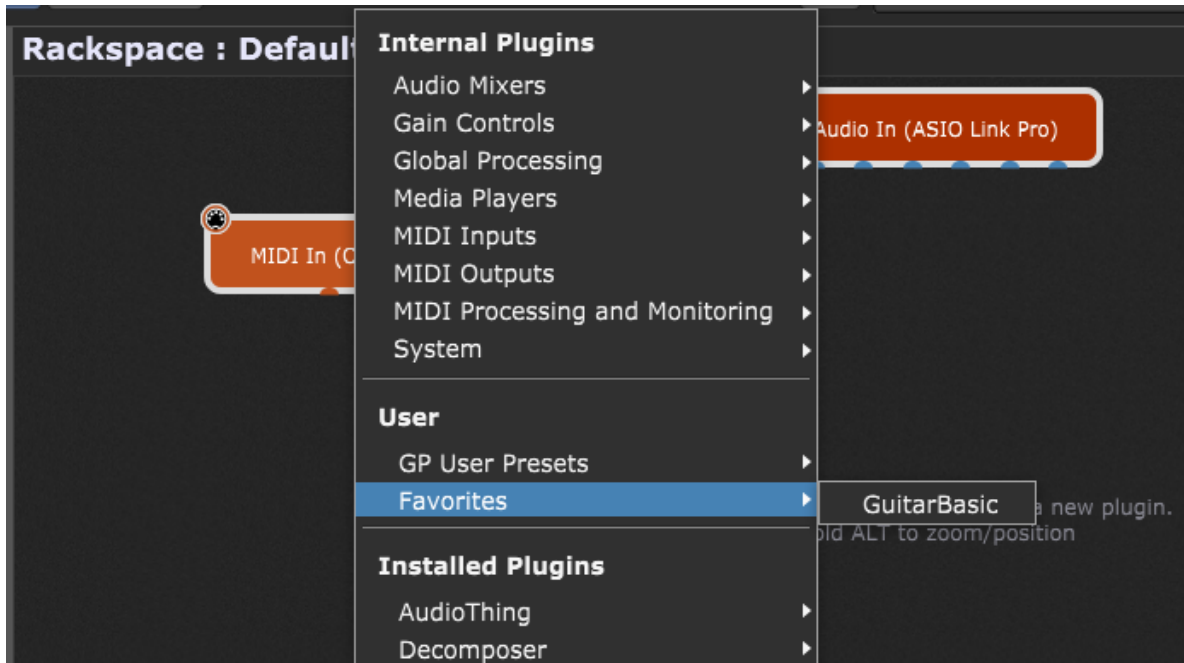
Simply select these plugins and right-mouse click on any selected plugin to open the [Selected Blocks contextual menu](#):



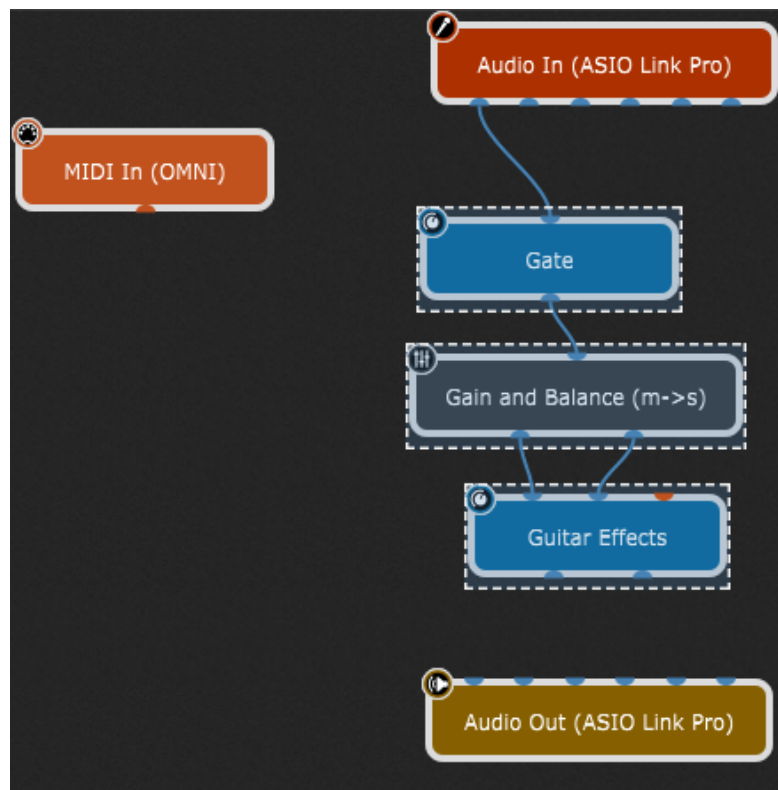
Select the **Save as Favorite...** menu item, type in a name (e.g. GuitarBasic) in the dialog that appears and click on the Save button.

Now let's create a new empty gig (File -> New -> New Empty Gig) to reuse the saved favorite:



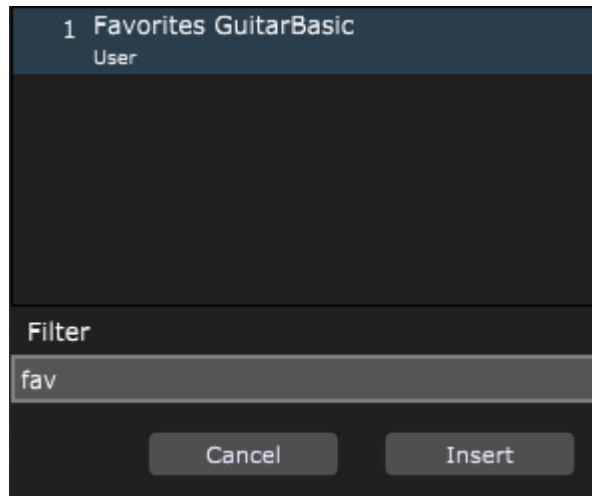


Right-click on the empty surface to open the [contextual menu](#), then expand Favorites and select your favorite file:



Three blocks are automatically inserted, connected to the audio input, and interconnected as you saved them. Additionally, all parameter values in these blocks are restored, so you can continue building your setup and connect it to the audio outputs.

**Tip:** you can also use the [Quick Plugin, Favorite or Preset Finder](#), and filter for favorites:



Simply type in "fav" to show all favorites, and pick the desired from the list.

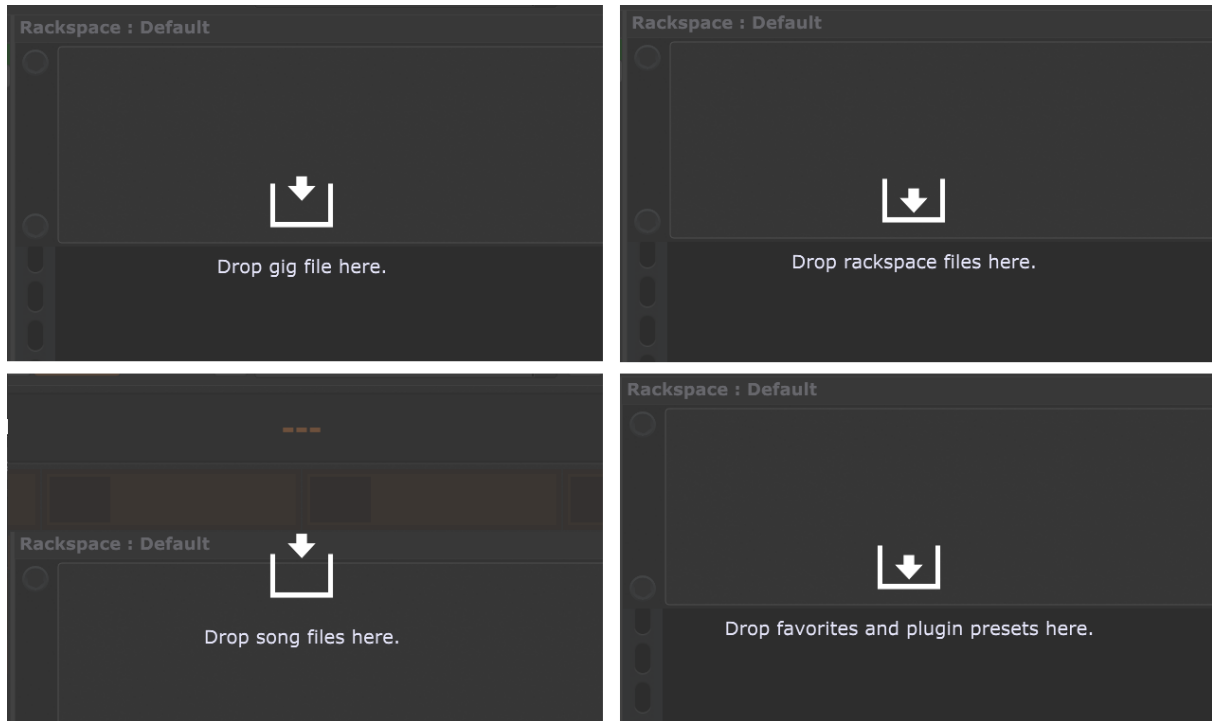
Favorites are a very valuable workflow tool, particularly useful for musicians who do session work and need to very quickly create certain sounds on demand. It is now easy for a musician to create a collection of favorite sounds, made from multiple interconnected blocks and just recall them when needed.

**Note:** to delete a favorite, click on the Window menu and then click on the *Open Content Folders* menu item; open the Favorites folder and then delete the desired favorite file.

You can insert favorites by using the drag-and-drop technique; check the [Drag and drop support](#) section to learn more.

## 19.5 Drag and drop support

Gig Performer supports inserting files using drag and drop:



The following file types are supported:

- Gig files,
- Rackspace files,
- Song files,
- User plugin presets,
- Favorites.

Simply click on a supported file and while holding the mouse drag it into the Gig Performer window. One of the four screens above will appear depending on the file type.

**Note:** you can load songs using drag and drop in the [Streaming Audio File Player](#) and the [MIDI File Player](#) plugins.

## 19.6 MIDI Assignment List window

Clicking on **Window** -> **MIDI Assignments** opens the MIDI Assignments List window:

Kind	Name	MIDI assignment
Global	Panic	CASIO CT-X5000 : Controller 23, Channel 1
Global	Play/Stop	CASIO CT-X5000 : Controller 47, Channel 1
Global	Tuner	CASIO CT-X5000 : Controller 9, Channel 1
Widget	Volume	CASIO CT-X5000 : Controller 7, Channel 1

This window displays the information from the Global MIDI Assignments section in the [Global MIDI Options](#), MIDI assignments in [Setlist Options](#), and [widget MIDI assignments](#). If you are wondering why some note is not playing or some MIDI CC message doesn't seem to be working as expected, check this assignment list to make sure that you didn't accidentally assign the message to a widget, thereby preventing it from being passed through to a MIDI In block.

**Note:** double-clicking a line containing a widget (see the Kind column) opens the [Widget Properties](#) window for that widget.

Widget MIDI assignments are always displayed for the currently active rackspace (not all rackspaces in a gig file).

**Note:** mappings to physical controllers made in the Widget Properties inspector override Global MIDI Assignments.

## 19.7 Widget Mappings List window

Clicking on **Window** -> **Widget Mappings** opens the Widget Mappings List window:

Caption	Rackspace	Type	Handle	Current Value	Scaled Value	Plugin Name	Plugin Caption	Parameter Name	Parameter Value	Widget Link	Radio Group
Channel 1-2 volume	Global	Knob		70.8	70.8	Audio Mixer (32ch)	Audio Mixer (32ch)	Channel 1-2 volume	0dB		
Room Size	Local	Knob		68.8	68.8	KeysOfThe70s	KeysOfThe70s	Room Size	7.19		
Reverb Mix	Local	Knob		61.0	61.0	KeysOfThe70s	KeysOfThe70s	Reverb Mix	0.61		
Reflection Fact.	Local	Knob		67.0	67.0	KeysOfThe70s	KeysOfThe70s	Reflection Fact.	0.67		
Reverb Tail	Local	Knob		43.0	43.0	KeysOfThe70s	KeysOfThe70s	Reverb Tail	0.43		
Reverb	Local	Button		0.00	0.00	KeysOfThe70s	KeysOfThe70s	Reverb	Off		
Mid	Local	Knob		15.8	15.8	KeysOfThe70s	KeysOfThe70s	Mid	1.04		
Low	Local	Knob		15.1	15.1	KeysOfThe70s	KeysOfThe70s	Low	1.00		
High	Local	Knob		15.3	15.3	KeysOfThe70s	KeysOfThe70s	High	1.01		
EQ	Local	Button		0.00	0.00	KeysOfThe70s	KeysOfThe70s	EQ	Off		
Output Vol.	Local	Knob		46.0	46.0	KeysOfThe70s	KeysOfThe70s	Output Vol.	2.30		
P I A N O	Local	Label		100.0	100.0						

This window allows you to easily review and/or troubleshoot not only widget mappings but also other useful information such as Widget Link, Radio groups or handles for the current rackspace and the Global rackspace.

Double-clicking an entry selects the corresponding widget in Edit mode.

## 19.8 Plugins In Use

Gig Performer allows you to see the list of used plugins for the loaded gig file. Click on **Window -> Plugins in Use:**

Plugin name	Manufacturer	Format	Version	Location	Rackspace
Blue3	GG Audio	VST	2.3.1.0	C:\Program Files...	Combo
ExaktLite	Sonicbits	VST3	1.4	C:\Program Files...	Combo
GGate	GVST	VST	1.2.0.0	D:\NEMANJA\PL...	GLOBAL RACKSPACE
KeysOfThe70s	Lostin70s	VST3	2.2.2	C:\Program Files...	Combo
					Piano
					Stage
					Vibraphone
MReverb	MeldaProduction	VST	14.1...	C:\Program Files...	Combo Vibraphone
S-GEAR 2	Scuffham Amps	VST3	2.9.6.0	C:\Program Files...	Stage
Sitala	Decomposer	VST	1.0.9.0	C:\Program Files...	Stage
TH-U-64	Overloud	VST	1.4.1...	C:\Program Files...	GLOBAL RACKSPACE
Youlean Loudness...	Youlean	VST	2.0.4.0	D:\NEMANJA\PL...	GLOBAL RACKSPACE

Copy

This list is sorted alphabetically by the *Plugin name* column. If a plugin is used in multiple rackspaces (like the KeysOfThe70s in the screenshot above), then the plugin will be listed only once, and the corresponding rackspaces will be listed below that entry. Plugins that are disabled in the [Plugin Manager](#) or whose files are missing are displayed in yellow.

Clicking on the **Copy** button will copy all entries to the Clipboard so you can easily paste the copied content into, for example, a spreadsheet application.

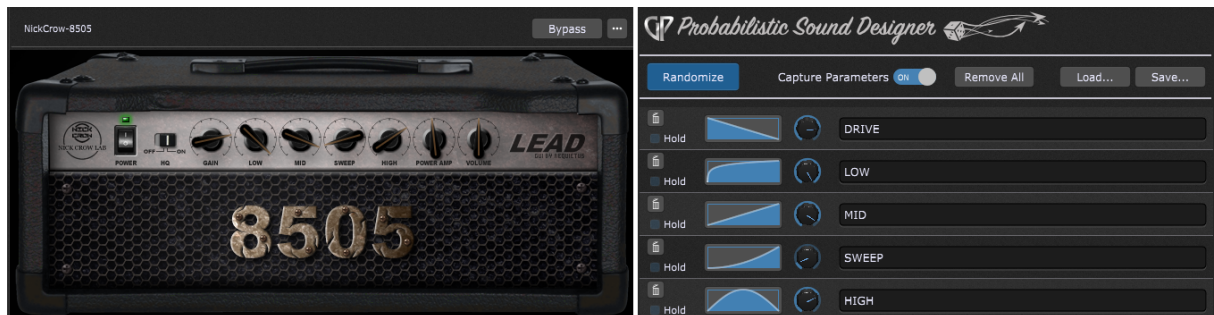
	A	B	C	D	E	F	G	H
1	<b>Name</b>	<b>Missing</b>	<b>Manufacturer</b>	<b>Format</b>	<b>Version</b>	<b>FileInfo</b>	<b>Rackspace</b>	
2	Blue3		GG Audio	VST	2.3.1.0	C:\Program Files\St Combo		
3	ExaktLite		Sonicbits	VST3	1.4	C:\Program Files\Cc Combo		
4	GGate		GVST	VST	1.2.0.0	D:\NEMANJA\PLUG GLOBAL RACKSPACE		
5	KeysOfThe70s		Lostin70s	VST3	2.2.2	C:\Program Files\Cc Combo		
6	KeysOfThe70s		Lostin70s	VST3	2.2.2	C:\Program Files\Cc Piano		
7	KeysOfThe70s		Lostin70s	VST3	2.2.2	C:\Program Files\Cc Stage		
8	KeysOfThe70s		Lostin70s	VST3	2.2.2	C:\Program Files\Cc Vibraphone		
9	MReverb		MeldaProduction	VST	14.16.0.0	C:\Program Files\Vs Combo		
10	MReverb		MeldaProduction	VST	14.16.0.0	C:\Program Files\Vs Vibraphone		
11	S-GEAR 2		Scuffham Amps	VST3	2.9.6.0	C:\Program Files\Cc Stage		
12	Sitala		Decomposer	VST	1.0.9.0	C:\Program Files\St Stage		
13	TH-U-64		Overloud	VST	1.4.10.0	C:\Program Files\Vs GLOBAL RACKSPACE		
14	Youlean Loudness Meter 2	X	Youlean	VST	2.0.4.0	D:\NEMANJA\PLUG GLOBAL RACKSPACE		

Missing or disabled plugins will have "X" in the *Missing* column.

**Note:** if you are using the [Predictive Loading](#) feature, make sure that you first **save** your gig file to view an accurate list of plugins that have been used.

## 19.9 Probabilistic Sound Designer

The Probabilistic Sound Designer (PSD) will work with any plugin and allows you to design random but related sounds by randomizing selected values of plugin parameters and using probabilities to determine how each parameter can be varied:

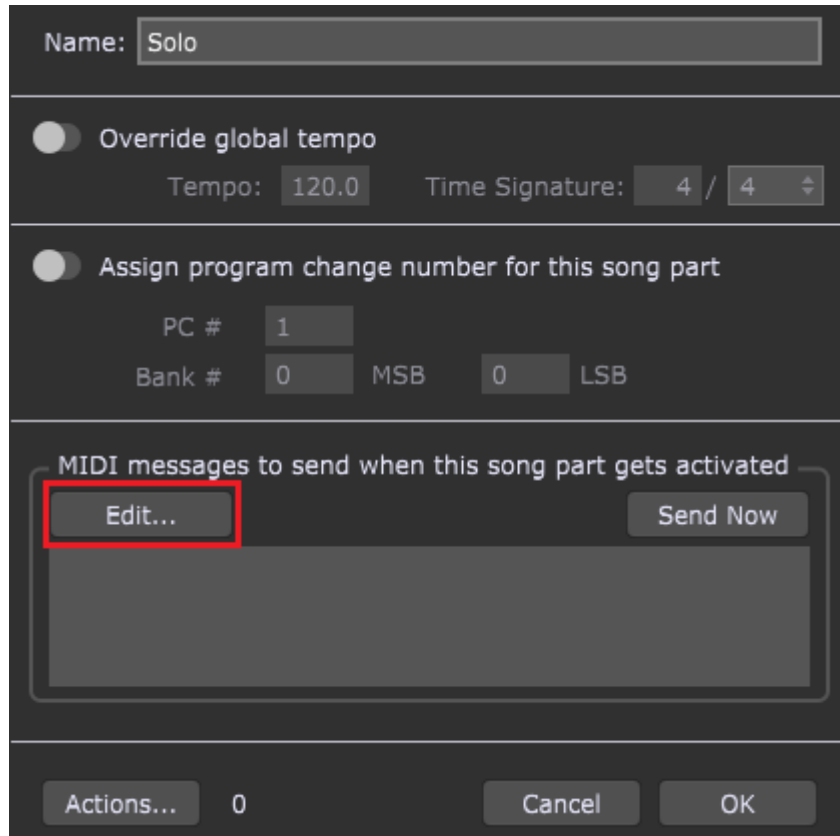


Simply move the desired controls (such as buttons or knobs) of your favorite plugin and the corresponding parameters will be automatically captured by the Probabilistic Sound Designer. Click on the *Randomize* button to create a new sound. Discover what amazing sounds your plugin can produce. Of course, you can save your created configuration along with the plugin preset so that you can later continue from where you left off.

Check [this chapter](#) to learn more.

## 19.10 MIDI Helper Tool

The MIDI Helper Tool allows you to easily create the desired MIDI messages interactively. To open this tool, click on the **Edit...** button, as indicated in the red rectangle.



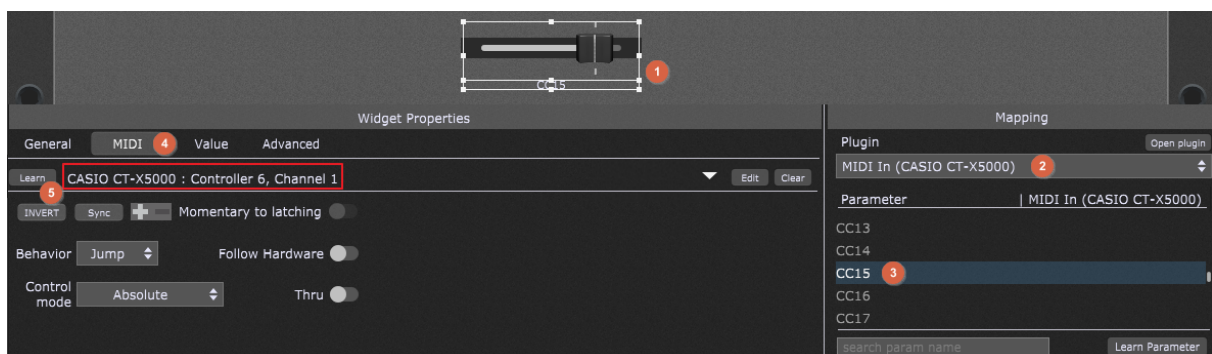
Refer to the [Song Part Properties](#) chapter for more information about the MIDI Helper Tool. This tool is also used in [MIDI Out](#) blocks to send arbitrary sequences of MIDI messages (including SysEx) when a rackspace is activated.

## 19.11 Remap MIDI messages without using the MIDI Filter plugin

Like other plugins, a [MIDI In](#) block has numerous parameters that widgets can control. These parameters include any MIDI note or continuous controller message. This makes it possible to remap MIDI messages coming from a physical controller without using the [MIDI Filter](#) plugin. Perform these steps to remap MIDI messages:

- First, make sure that your MIDI device is connected to your computer and that you have added the corresponding MIDI In block onto the surface in the Wiring view.

- In the Panels view Edit mode, create a new widget, e.g. a Black Slider (Horizontal) and click on it to select it **(1)**.
- In the widget mapping area, choose the MIDI In block corresponding to your desired MIDI device **(2)**.
- Choose a note or continuous controller from the Parameter list in the Mapping section **(3)**.
- In the widget properties area, click on the MIDI tab **(4)**, then click on the Learn button **(5)** and move the physical controller.



Note that this area indicated with the red rectangle changed from "(no MIDI In assignment yet)" to something else (depends on the MIDI device you used and what physical controller you moved).

Incoming messages from the physical controller (e.g. Controller 6, Channel 1) are now remapped to the note or continuous controller you chose (e.g. CC15).

Of course, this incoming CC could as easily be remapped to pitchbend, aftertouch, or mod wheel messages.

## 19.12 Additional tips

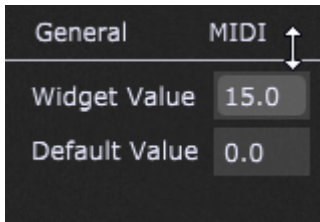
In this section you will learn more about useful tips that may help you to get things done quicker.

Before we start, make sure to check out this blog article: [Best practices for a rock-solid software-based setup](#).

### Entering values

You can easily set values in various fields by clicking on the field and moving the mouse up or down:





Additionally, if you also hold the **Shift** key, values are increased or decreased by 0.1. Note that the dragging speed is also taken into account; moving your mouse faster will make greater changes in values.

By default, clicking on the field opens a mini pad for entering values:



Click on the up/down arrows to change value by 1, hold the **Shift** key to change the value by 0.1, or simply type in the desired value.

Turning on the *Touch-friendly number editing* toggle button in [Display Options](#) shows a larger keypad instead of the mini pad:

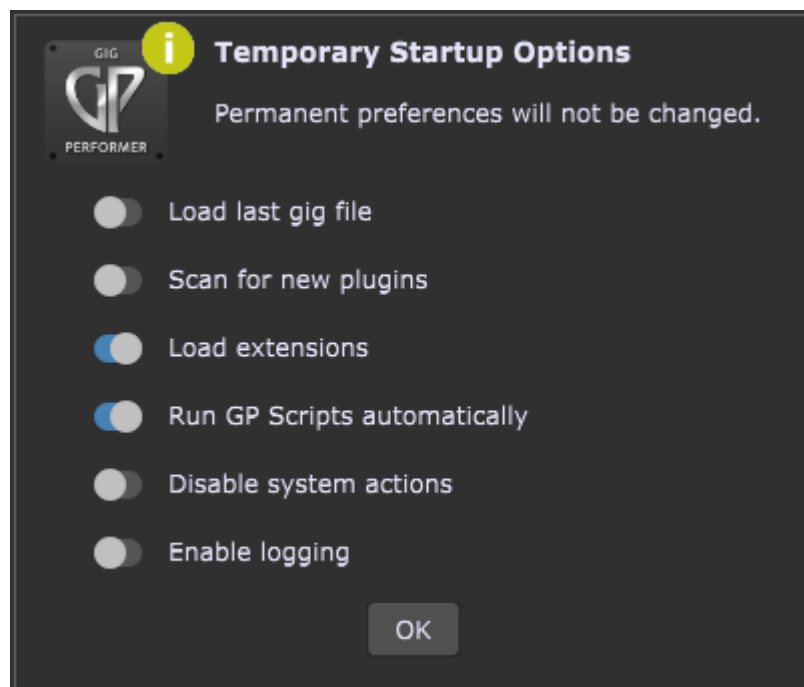


Use this option if you run Gig Performer on a touchscreen device such as a Microsoft Surface tablet.

**Note:** you can also directly enter values using MIDI numbers, by adding the letter "m" at the end of the value. For example, if you type in "64m", Gig Performer will interpret it as an exact value of 50.39 (roughly 50% of the maximum MIDI value of 127).

### Temporary Startup Options

Hold the **Shift** key down while Gig Performer is starting and it will open the Temporary Startup Options dialog:



This is a one-time menu, and permanent [options and preferences](#) will not be changed. For example, if *Load last gig file* is turned off, Gig Performer will not load the previous gig file, even if you have the toggle button "Reload the last gig on startup" turned on in the [General Options](#) window. Visit [this blog article](#) to learn more about these options.

### Dragging rackspace and songs

Holding the **Shift** key while dragging rackspace (or songs in the Setlists view) prevents the dragged rackspace/song from being activated (this can be very useful for adjusting your setlist order when [Predictive Loading](#) is enabled).

## Quick stereo connection

To make a quick stereo connection, you can either hold the **Shift** key down and drag a wire from an output port of one plugin to the input port of another plugin or hold the **Shift** key down while dragging a plugin until its output pins touch the input pins of another plugin, then release (unclick) the mouse. If there is already a connection, this operation will cause it to be removed instead.

## Fine control

Hold down the **Cmd** (macOS) or **Ctrl** (Windows) key for fine control while adjusting a widget.

## MIDI In plugin channel toggling

In the MIDI In plugin editors, if you hold the **Shift** key down while enabling or disabling individual MIDI channels by clicking on their respective check boxes, the other check boxes will be set to their opposite states. This is a convenient way to quickly enable a single channel while blocking all others.

## Global MIDI options and Setlist options learn buttons

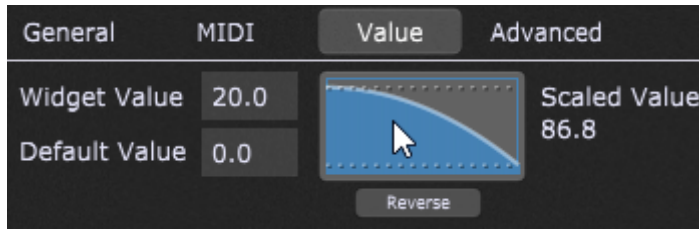
Holding the **Shift** key while clicking on the *Learn MIDI for selected item* buttons in the [Global MIDI Options](#) and [Setlist Options](#) dialog clears the MIDI assignment.

## Quickly assign multiple widgets to a Widget Link group

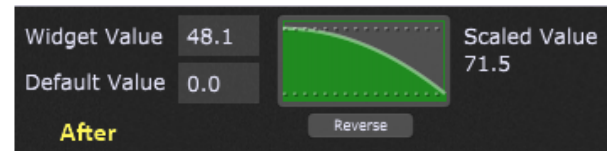
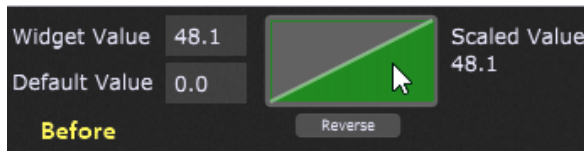
To quickly assign two or more selected widgets to a widget link group "N", select all widgets and press the keyboard key N while holding the **Alt** key (or **Option** on Mac).

## Copy and paste curves

If you hold the **Shift** key down while clicking on the scaling curve (of a widget, Probabilistic Sound Designer, or velocity curves in MIDI In blocks), the curve settings will be copied to the clipboard:



If you then hold down **Shift + Ctrl** (or **Shift + Cmd** on Mac) and click on another curve, the previously copied settings will replace the settings of that curve:

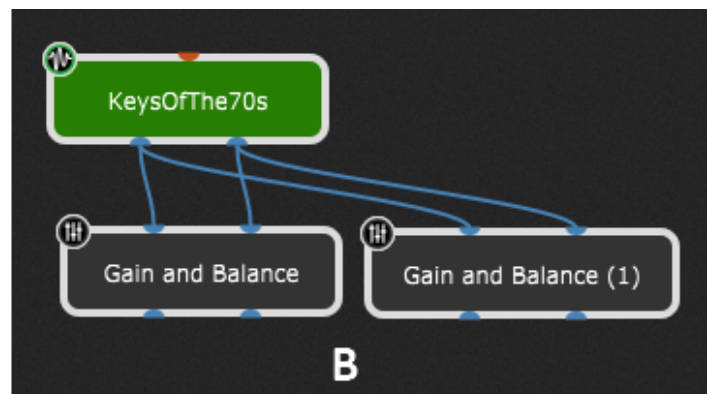
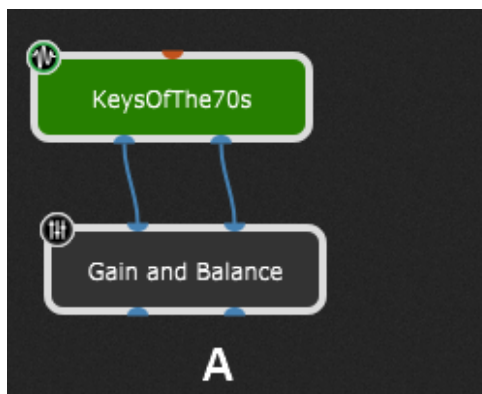


## Widget actions menu

If you click on a widget while holding the **Shift** key in the Panels view, the Widget actions menu will appear; check the [Widget contextual menu](#) chapter to learn more.

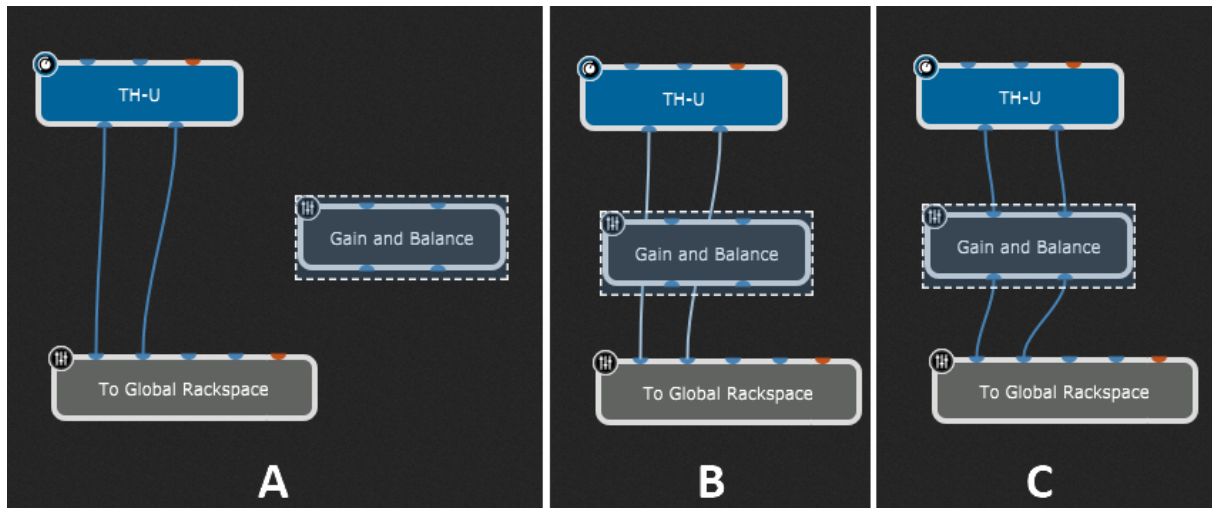
## Duplicate a plugin with inputs

If you right-click on a plugin while holding the **Shift** key in the Wiring view, you'll notice the *Duplicate with inputs* menu item. Clicking on this menu item will preserve connections with the plugin inputs. For example, if you duplicate with inputs the Gain and Balance plugin block in **(A)**, then the duplicate will preserve the connections, as indicated with **(B)**:



## Insert a plugin block into audio or MIDI path

Plugin blocks can be automatically inserted into an Audio/MIDI path by holding down the **Shift** key while dragging the block over the connection wires. Simply hold **Shift** and click on the desired plugin block **(A)**, then drag it onto connection wires until they light up **(B)**, and then release the click **(C)**.



Hold **Shift** and drag an already inserted block to remove that block from the path.

### Undo facility

The Undo feature is supported in the following operations:

- Works for widget movement, widget resizing and widget caption resizing.
- Scaling curve can be undone.
- Randomize parameters operation in the Probabilistic Sound Designer can be undone.
- Works for the three widget color attributes (fill, outline, text) and the font.
- Widget Undo also works when you paste attributes or fonts into multiple widgets.
- Pressing the Backspace key while the mouse is over a wire will delete the wire; use the Undo feature to reconnect the wire.
- Undo is supported in Chord Maker and MIDI Rechannelizer.

## 20 Extensions

This chapter is dedicated to Gig Performer extensions. You will learn about extensions and how to enable or disable them.

Please refer to [this blog article](#) to check out what third-party extensions are currently available for Gig Performer.

### 20.1 What are extensions?

[Extensions](#) are basically libraries (.dll files on Windows and .dylib files on macOS) independently developed that Gig Performer can load to provide additional functionality, i.e., additional support for a hardware device, some new GUI functionality or even additional GPsScript functions. For the first, an extension would make your hardware device usable directly from within Gig Performer as if it was already built-in.

For example, if you had a keyboard with a display and the keyboard's manufacturer created an extension for Gig Performer (with our SDK), Gig Performer would be able to communicate directly with your keyboard, and show various parameters, such as Song name, Rackspace name or Tempo on the display (or assign buttons and knobs with functions in Gig Performer).

Additionally, extensions may also provide panels and menu items in Gig Performer that may help to use them better.

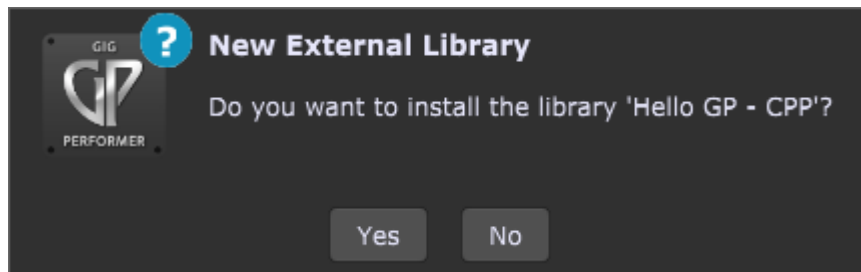
Gig Performer exposes an API (application programming interface) that allows third-party developers to create libraries that can be used to interface with Gig Performer. The primary purpose of the API is to enable the creation of libraries to manage control surfaces and other hardware that one might want to use with Gig Performer, but it is also possible to create 3rd party GUI tools as well.

Extensions are developed using Gig Performer's SDK (software development kit) which is freely available under the GPL-3.0 license and [can be downloaded from GitHub](#). Visit that link to learn more about SDK and fundamentals.

## 20.2 Enabling/disabling extensions

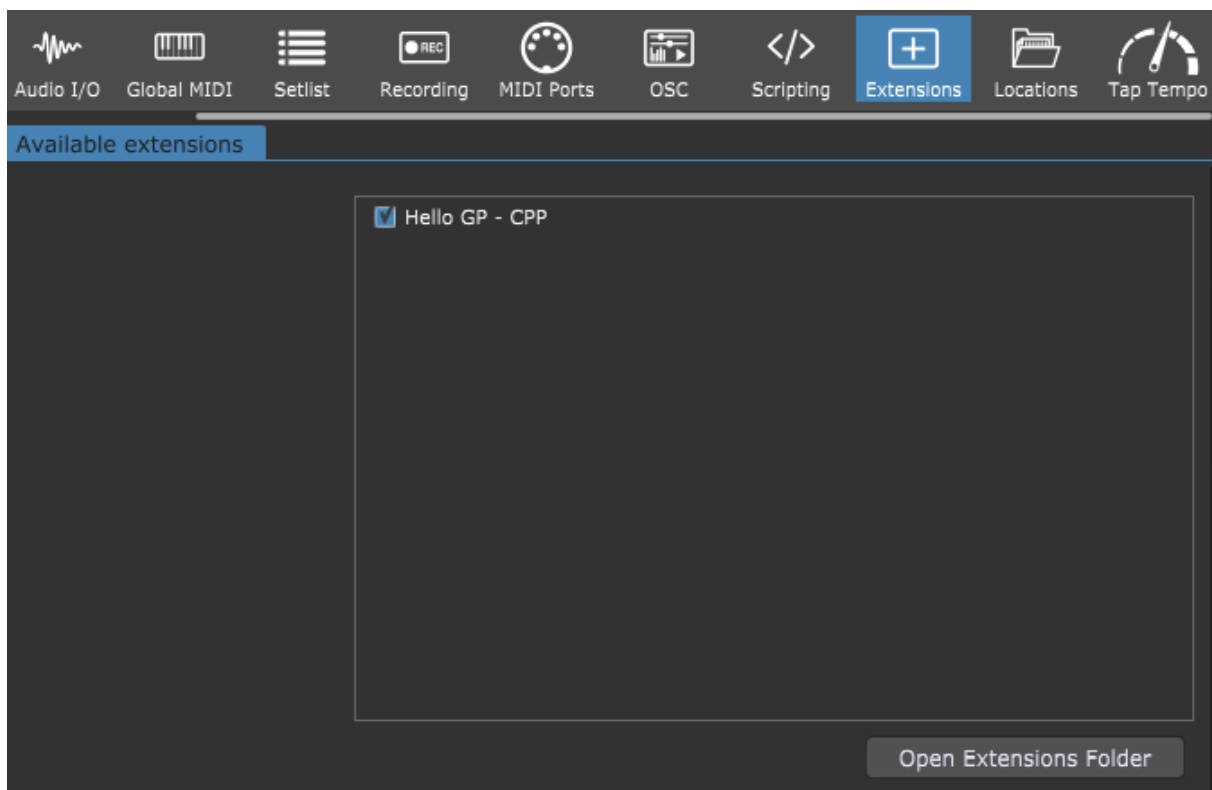
If you obtained a Gig Performer extension, first make sure to place it in the Extensions folder (refer to the [File locations in Gig Performer](#) chapter for more information). If it is supplied as an archive (i.e. a ZIP archive), extract it directly into the Extensions folder.

Afterward, launch Gig Performer:

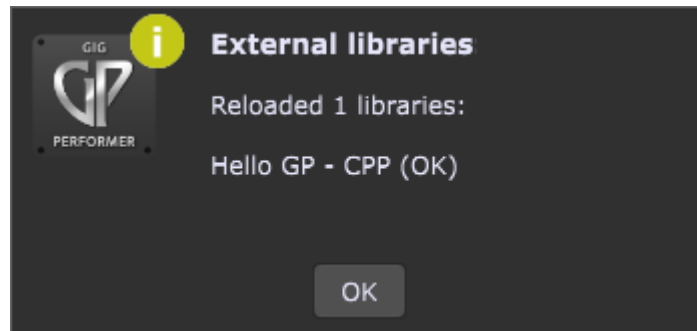


The *New External Library* dialog will appear, allowing you to install the found external library.

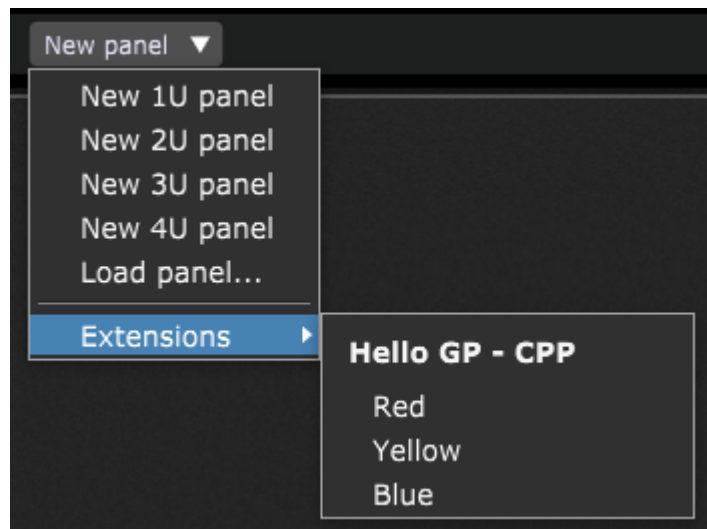
Installed extensions are enabled by default. To manage installed extensions open the [Extensions Options](#):



If you want to reload your enabled extensions, click **Options** -> **Reload Third Party Libraries**:



If an extension provides one or more custom panels, they will be displayed under the **New panel** drop-down list in the **Extensions** sub-menu (which is otherwise hidden if there aren't extensions that provide additional panels):



Expand the Extensions sub-menu and select the desired panel. Alternatively, open the panel contextual menu, expand the Extensions sub-menu (which is hidden if there aren't extensions that provide additional panels) and select the desired panel.

**Note:** if you suspect that an extension is preventing Gig Performer from starting normally, you can temporarily disable extensions in the [Temporary Startup Options](#). Check the [Additional tips](#) chapter to learn more.



## 21 Optimizing your computer for a glitch-free audio performance

In this chapter you can find out more about latency, sample rate, buffer size and what settings are acceptable for performing live. Also, you will learn how to optimize your operating system (Windows or macOS) to ensure maximum performance and stability.

Additionally, you may want to optimize your plugins and make clever choices in their configuration. To learn more, please refer to this blog article [Clever ways to optimize your plugin usage](#).

In [this blog article](#) we have compiled best practices for a rock-solid setup.

### 21.1 Latency explained

In this chapter we will introduce [some basic terms](#) such as audio latency, buffer size and sample rate and suggest recommendations for live use.

**Audio latency** is simply the amount of time that passes between the sound being generated and then perceived by your brain. Basically, it is a delay.

For example, if you are 10ft away from the speakers, and since the speed of sound is approximately 1,000 ft/s in air, it means that it takes  $10 \text{ ft} : 1000 \text{ ft/s} = 0.1 \text{ seconds}$  (or 10 milliseconds) for sound to travel from the speakers to your ears. The latency here is about 10 ms.

**Buffer size** is basically the number of samples that will be collected before your audio plugins get to process them. Your audio interface is an analog-to-digital as well as digital-to-analog converter. It takes any audio input, converts that into digital form (numbers) and then converts those numbers back to analog audio on the output side.

The **Sample rate** determines how many samples your audio interface will capture every second and perform the above-mentioned conversions. A common sampling frequency for live use is 44.1 KHz.

For example, if your buffer size is **256** and your sampling rate is **44.1 KHz** (44,100 times per second, as Hz means cycles per second) then your latency will be  $256/44,100 \text{ seconds}$  which is 0.0058 seconds or **5.8 ms**.

If your buffer size is **256** and the sample rate is **96 KHz** you will get  $256/96,000 = 2.7 \text{ ms}$  latency.

You can experiment with this: If you change the buffer size to **128** and leave the sampling frequency at **44.1 KHz** – your latency will be **2.9 ms** and so on.

These values directly affect the performance of your PC, as smaller latency values require your computer to respond more quickly to process all those samples in time without producing any glitches.

It has been shown that people can perceive differences between 3 ms – 10 ms, and that our brain cannot distinguish anything below 3 ms.

Therefore, for live performance, many musicians use a buffer size of 256 or 128 and a sample rate of 44.1 KHz. We are not discussing here a recording scenario, which should be done slightly differently.

Generally speaking, using higher buffer sizes or lower sample rates lets your computer run at lower temperatures and allows you to run more intensive plugins in parallel without crackles or pops, but the trade-off will be higher latency or lower quality respectively.

Another factor is the additional latency introduced by your audio interface. Not all audio interfaces are created equal. Some have low internal latency for both A/D and D/A conversions while others add more latency than perhaps you might like, so you have to lower your buffer size to get the overall desired latency.

To find out how much extra latency your interface introduces - you can use Gig Performer's built-in [Latency Measurement Tool](#).

**Note:** Gig Performer also provides a feature to see how much latency every new plugin adds to your setup. Simply hover over your plugin block to show a tooltip and look for the Latency information. Find out more in the [Plugins](#) chapter.

The point is to create sufficient CPU headroom to ensure a glitch-free live performance. You can also set smaller buffer sizes if that does not impact your live performance, but at the cost of less CPU capacity for processing more plugins.

## 21.2 Optimizing your Windows PC for maximum performance and stability

To get the most from your Windows operating system for digital audio processing, many Windows settings and features can be configured to achieve better performance and stability.

Deskew Technologies, LLC provides you with a free e-book [The Ultimate Guide to Optimize your Windows PC for the Stage](#), that will help you to avoid issues like crackles, dropouts, audio glitches, high latency and configure your PC to be a great audio platform.

## 21.3 Optimizing your MacOS

MacOS has real-time audio and MIDI functionality built-in. Multiple applications can use audio and MIDI at the same time. You can also aggregate multiple audio interfaces into a single unified "large" audio interface.

We have compiled a list of tips to fully prepare your Mac for your live performance. Visit this blog article to learn more: [Optimize your Mac for a gig](#).

## 22 Join our community

Many music professionals worldwide rely on Gig Performer as their audio plugin host of choice for live performance. The official community for Gig Performer users is located at: <https://community.gigperformer.com/>

We encourage you to join our growing community and [share your experiences](#) with us. Besides community moderators and our active members, Gig Performer developers are also actively engaged in the community discussions. If you have a problem with Gig Performer, your audio setup, or just need some advice - no problem: many gigging musicians with ton of live performing experience will try to help you, suggesting tips or possible [solutions](#):

Rank13Jul 14

Lightbulb moment!

I'm not at the computer to test this but I think it should work.

- The audio player has a parameter for the timeline.
- Use widget Groups to link a widget for the audio player timeline with a widget for a mixer pan control.
- When the audio player is at the start this should make the pan control fully left.
- As the audio player starts it will then adjust the pan control and when the player finishes the pan control should be fully to the right. Right?

✔ Solution
7 ❤️
🔗
⋮
↩ Reply

---

David van EJul 14

Clever idea 💡 😊

❤️
🔗
⋮
↩ Reply

---

William EJul 14

This is brilliant! 🧐 👍

❤️
🔗
⋮
↩ Reply

---

HermanJul 14

I tried it, works perfect! Good to know 😊

📄 [AudioPlayerAutoPan.gig](#) (5.2 KB)

❤️
🔗
⋮
↩ Reply

---

IngemarJul 14

Hey that is fantastic. Nothing to buy. Great work and many thanks to Rank13.

1 ❤️
🔗
⋮
↩ Reply

The Community is organized and split into different categories; whether you want to e.g. discuss Gig Performer on [Windows](#) or [macOS](#), [plugins](#), [hardware](#), [OSC](#) or customizing your

setup using [scripts](#), simply select the corresponding category and create a new thread. If you want to find a specific issue i.e. "virtual audio ports latency", use the integrated search option.

Useful information in the Community is organized and indexed regularly; for example, check out our [complete Resources thread](#), where you can find all the tutorials and other material to help you improve your setup. Save your time, learn the best practices, and share your experiences with us at [community.gigperformer.com](https://community.gigperformer.com).

## 23 A word from the Gig Performer creators

Since the first release of Gig Performer® in 2016, we have tried to stay true to the main goals we set for ourselves before writing a single line of code. Those goals included stability, flexibility, performance, powerful features and ease of use for musicians, including ourselves, rather than for technicians.

- We wanted to create our "sounds" without being distracted by figuring out routings, buses, inserts, sends, and other constraints imposed by a channel strip paradigm. That drove our decision to use a visual metaphor for connecting plugins together.
- We wanted to be able to get up on stage and *not be stressed* out by whether pushing a particular button or turning a knob would crash the system.
- We wanted to ensure that Gig Performer was suitable for *all kinds of musicians*, and not biased towards keyboard players.

David is primarily a keyboard player and Nebojsa [was](#) primarily a guitarist. This allowed for two very different perspectives on how a plugin host should work to meet such disparate needs. The combined requirements resulted in a system that can be used by almost any kind of musician who wants to work with technology.

We feel that we have managed to stay true to these goals in Gig Performer while providing numerous new and important features. This iteration of Gig Performer is packed with new functionality and powerful improvements.

As touring musicians, we deeply understand of what it means to go on stage and perform in front of a live audience. We have tried to make Gig Performer® transparent and invisible in your performance. Ultimately, it all comes down to your artistic skills and expressions, and the technology around it is simply a tool that allows you to express yourself. We know deeply that on stage, everything "has to work."

We are confident that you will Own the Stage® with Gig Performer®.

We could not have done this without our **awesome** beta group. You may recognize some of these names / handles from our forums.

@pianopaul – Thaddäus: There's no way we could have found some of the bugs you did.

@schamass – Erik: Nothing really escapes your watchful eye.

@rank13 – Kieran: Thank you for all the user interface improvements.

@David-san – David: Thank you for suggesting things and always pushing back when needed.

@keyman – Antonio: There's no way we could have moved forward so fast without your input.

@jazzundso – Lukas: Thanks for your brilliant insights.

@jpt – John: Thank you for all your contributions.

@edm11 – Ed: Thank you for your valuable feedback and insights.

@speed12 – Mark: You've been helping us for a long time. Thanks for your support.

@simon – Simon: Always great feedback.

@LeeHarvey – Markus: Thanks for your honest and direct comments.

@tonycore – Tony: Thanks for all your efforts in bringing people aware of Gig Performer.

@brandon : Great insights.

@elucid – Chris: Thanks for testing!

@chris\_g – Chris: Thank you for the great templates.

@KCummings – Keith: Nice templates – thanks!

@Frank1119 – Frank: Thank you for your insights.

Rob Neal: Thank you for a wonderful design of a sustain pedal.

The Gig Performer team.

## 24 Appendix

In this chapter you can review the license agreement and find out more information about the basic terminology used in Gig Performer, file extensions and many other topics that are organized as subchapters.

### 24.1 License agreement

DESKEW TECHNOLOGIES, LLC - END USER LICENSE AGREEMENT FOR GIG PERFORMER

(“THE SOFTWARE”) BY CLICKING THE ACCEPT BUTTON OR DOWNLOADING THE SOFTWARE, YOU AGREE TO BE BOUND BY AND BECOME A PARTY TO THIS AGREEMENT. DESKEW TECHNOLOGIES, LLC’S ACCEPTANCE AND FULFILLMENT OF YOUR ORDER ARE EXPRESSLY CONDITIONED ON YOUR AGREEMENT TO THESE TERMS. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, YOU MAY NOT DOWNLOAD, INSTALL, EVALUATE OR USE THE SOFTWARE. CAREFULLY READ THE FOLLOWING LEGAL AGREEMENT (“AGREEMENT”) FOR THE LICENSE OF SOFTWARE YOU ARE ABOUT TO DOWNLOAD OR INSTALL, PROVIDED TO YOU UNDER THIS AGREEMENT AND THE ACCOMPANYING DOCUMENTATION (THE “DOCUMENTATION”). “DESKEW” MEANS DESKEW TECHNOLOGIES LLC. “YOU” MEANS THE INDIVIDUAL PERSON INSTALLING OR USING THE SOFTWARE ON HIS OR HER OWN BEHALF; OR, IF THE SOFTWARE IS BEING DOWNLOADED OR INSTALLED ON BEHALF OF AN ORGANIZATION, SUCH AS AN EMPLOYER, “YOU” MEANS THE ORGANIZATION FOR WHICH THE SOFTWARE IS DOWNLOADED OR INSTALLED, AND THE PERSON INSTALLING OR USING THE SOFTWARE REPRESENTS THAT HE OR SHE HAS THE AUTHORITY TO DO SO ON YOUR BEHALF. YOU HEREBY EXPRESSLY CONSENT TO DESKEW’S PROCESSING OF PERSONAL DATA YOU PROVIDE TO DESKEW (WHICH MAY BE COLLECTED BY DESKEW OR ITS DISTRIBUTORS) ACCORDING TO DESKEW’S CURRENT PRIVACY POLICY WHICH IS INCORPORATED INTO THIS AGREEMENT BY REFERENCE (SEE <https://gigperformer.com/privacy/>). IF “YOU” ARE AN ORGANIZATION, YOU WILL ENSURE THAT EACH MEMBER OF YOUR ORGANIZATION (INCLUDING EMPLOYEES AND CONTRACTORS) ABOUT WHOM PERSONAL DATA MAY BE PROVIDED TO DESKEW HAS GIVEN HIS OR HER EXPRESS CONSENT TO DESKEW’S PROCESSING OF SUCH PERSONAL DATA. PERSONAL DATA WILL BE PROCESSED BY DESKEW OR ITS DISTRIBUTORS IN THE COUNTRY WHERE IT WAS COLLECTED, OR IN THE LOCATION OF DESKEW OR ITS DISTRIBUTORS; UNITED STATES LAWS REGARDING PROCESSING OF PERSONAL DATA MAY BE LESS STRINGENT THAN THE LAWS IN YOUR JURISDICTION.

1. **License Grant.** Subject to the terms and conditions of this Agreement, DESKEW hereby grants you a non-exclusive, non-transferable right to reproduce and use a reasonable



number of copies of the Software and the accompanying Documentation solely for your internal evaluation purposes during the term of this Agreement (as set forth in Section 3 below). You may install each authorized copy of the Software only on the computer, workstation, personal digital assistant, pager, “smart phone”, or other electronic device for which the Software was designed.

**2. Ownership Rights.** The Software is protected by United States copyright laws and international treaty provisions. DESKEW and its suppliers retain all right, title, and interest in and to the Software, including all copyrights, patents, trade secret rights, trademarks, and other intellectual property rights therein. Your possession, installation, or evaluation of the Software does not transfer to you any title to the intellectual property in the Software. All rights in the Software and Documentation not expressly granted to you in this Agreement are reserved by DESKEW and its suppliers.

**3. Restrictions.** You may not rent, lease, loan, resell, or otherwise redistribute the Software or Documentation to any third party. You may not permit any third party to benefit from the use or functionality of the Software via a timesharing, service bureau, or other arrangement. You may not transfer (by assignment, sublicense, operation of law, or otherwise) any of the rights granted to you under this Agreement; and any purported transfer will be null and void and result in immediate termination of this Agreement. You may not reverse engineer, decompile, or disassemble the Software, except to the extent (if at all) the foregoing restriction is expressly prohibited by applicable law. You may not modify, or create derivative works based upon, the Software or Documentation, in whole or in part. You may not copy the Software or Documentation, except as expressly permitted in Section 1 above. You may not remove any proprietary notices or labels on the Software or Documentation. The Software may contain a usage tracking mechanism for purposes of monitoring license compliance. You agree not to (a) disable or block the operation of this tracking mechanism and (b) delete tracking information that may be stored with the Software. DESKEW reserves the right to review such usage tracking information for the purpose of verifying your compliance with this Agreement.

**4. Warranty Disclaimer.** THE SOFTWARE IS PROVIDED “AS IS.” TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, DESKEW DISCLAIMS ALL WARRANTIES, WHETHER EXPRESSOR IMPLIED OR STATUTORY, WITH RESPECT TO THE SOFTWARE AND THE DOCUMENTATION, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON INFRINGEMENT. Without limiting the generality of the foregoing, the Software is not fault-tolerant and is not designed or intended for use in hazardous environments requiring fail-safe performance, such as the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, weapons systems, direct life-support

machines, and any other application in which the failure of the Software could lead directly to death, personal injury, or severe physical or property damage. You assume full responsibility for the installation of, use of, and results obtained from the Software. Without limiting the foregoing, DESKEW makes no warranty that the Software will be error-free.

**5. Limitation of Liability and Indemnification.** IN NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER IN TORT, IN CONTRACT, OR OTHERWISE, WILL DESKEW OR ITS SUPPLIERS BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY DAMAGES WHATSOEVER, INCLUDING DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY NATURE ARISING FROM OR RELATING TO THIS AGREEMENT OR THE SOFTWARE OR DOCUMENTATION, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, OR COMPUTER FAILURE OR MALFUNCTION, EVEN IF DESKEW HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THESE LIMITATIONS OF LIABILITY WILL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION.

**6. United States Government.** The Software and Documentation are deemed to be “commercial computer software” and “commercial computer software documentation,” respectively, pursuant to DFAR Section 227.7202 and FAR Section 12.212, as applicable. Any use, modification, reproduction, release performance, display or disclosure of the Software and Documentation by the United States Government will be governed solely by the terms of this Agreement and will be prohibited except to the extent expressly permitted by this Agreement.

**7. Export Controls.** You are advised that the Software is subject to the U.S. Export Administration Regulations. You will not directly or indirectly export, import, or transfer Software contrary to U.S. or other applicable laws, and will not cause, approve, or assist others to do so. You represent and agree that (a) neither the United States Bureau of Export Administration nor any other federal agency has suspended, revoked or denied your export privileges, and (b) you are not located in or a resident of any other country to which the United States has embargoed goods.

**8. Miscellaneous.** This Agreement is governed by the laws of the United States and the State of New York, without regard to conflict of laws principles. The application of the United Nations Convention of Contracts for the International Sale of Goods is expressly excluded. Any dispute between the parties arising out of or related to this Agreement shall be settled by final, binding arbitration utilizing the dispute resolution procedures of the American Arbitration Association (AAA) in San Francisco, CA. Notwithstanding the foregoing, DESKEW may bring suit in any appropriate forum for any breach of Sections 1,

2, 5 or 10 or otherwise for infringement or misappropriation of its intellectual property rights. BY AGREEING TO ARBITRATION, THE PARTIES UNDERSTAND THAT THEY ARE WAIVING CERTAIN RIGHTS AND PROTECTIONS WHICH MAY OTHERWISE BE AVAILABLE IN COURT, INCLUDING, WITHOUT LIMITATION, THE RIGHT TO A JURY TRIAL. DESKEW will be entitled to immediate injunctive relief (in addition to all other remedies it may have) in the event of any breach, or threatened breach, of Section 2 or Section 5 by Licensee. No right or remedy of DESKEW under this Agreement will be deemed waived unless such waiver is in writing and signed by an authorized representative of DESKEW. If any provision of this Agreement is held invalid, such provision will be deemed modified so that it is valid and enforceable to the maximum extent allowed under applicable law and the remainder of this Agreement will continue in full force and effect. The parties confirm that it is their wish that this Agreement has been written in the English language only and if this Agreement is translated into any other language, the English version will prevail. This Agreement is the entire agreement between the parties and supersedes any other written or oral communications or understandings with respect to the Software. This Agreement may not be modified except by a written document signed by authorized representatives of each party that expressly states their intention to amend this Agreement.

## 24.2 Basic terminology

In this chapter we present an alphabetical list of basic terms used in Gig Performer's world.

### **Control Change**

Most commonly referred to as a "CC". A CC message consists of a controller number and a value between 0-127. Expression pedals, knobs, faders and buttons typically send these types of messages. Synth modules might also send these messages out when you adjust certain parameters.

### **Curves**

Also known as scale curves, curves are used to adjust how widgets move as a function of incoming MIDI messages. Curves are also used to define velocity mappings for keyboards.

### **Extensions**

Extensions are basically external libraries that Gig Performer can load and provide the additional functionality, e.g. support for a hardware device such as Stream Deck. [Click here](#) to see the list of available extensions for Gig Performer.

**Gig**

A Gig Performer file (gig file or "gig") containing all the data you have created - rackspaces, variations, and all widgets and widget groups.

**Global rackspace**

A special always running rackspace that can be used to manage common plugins needed by most or all rackspaces. For example, you might have a compressor or final EQ that you want all rackspaces to use. Or you might want to run a looper plugin that needs to stay running regardless of which rackspace is running. If you use the same instrument (an acoustic piano perhaps) in almost all your rackspaces, you might insert that instrument in the Global rackspace instead, making your other rackspaces easier to manage.

**Handle**

In the Gig Performer universe, a handle is a name that you can assign to a plugin or a widget to use them in Timeline or Song Part actions, Gig Performer extensions, Gig Performer's proprietary programming language GScript and OSC. Historically, handles were used exclusively for advanced use cases with both GScript and OSC. Starting with Gig Performer 4.0 and the public availability of the GP-SDK, allowing 3rd party extensions to be developed, developers created extensions that interacted with plugins and widgets through their handles. To learn more about handles, check [this blog article](#).

**Host automation**

The old approach of controlling plugin parameters is through the direct MIDI controller mapping to these parameters. However, the modern approach of controlling plugin parameters is through the plugin's host automation support. This means that a host (Gig Performer) can automate (manage) plugin's parameters through widgets.

**Instance**

Multiple copies of Gig Performer can be launched and run simultaneously on a single computer. Each of these independent copies, running in its own window, is called an "instance" of Gig Performer. This is useful for applications like giving different band members the ability to change the rackspace affecting their particular instrument without changing the rackspace affecting other instruments.

**MIDI**

An acronym for **M**usical **I**nstrument **D**igital **I**nterface—a standard protocol for the interchange of musical information between computers, controllers, and hardware-based electronic musical instruments such as synthesizers and samplers. Visit [this blog article](#) to learn how Gig Performer enables you to have the most flexible MIDI processing.

**MIDI controller**

A physical device such as a keyboard or pedalboard typically connected to your computer via USB.

**OSC**

Acronym for **Open Sound Control**—a communications protocol that is optimized for modern networking technology.

**Panel**

In Gig Performer, an area (visually similar to hardware rack-mounted sound modules and signal processors) containing widgets used to control the plugins in the rackspace containing the panel. A rackspace can contain multiple panels, each containing one or more widgets.

**Panels View**

The Gig Performer screen containing the panels and widgets used to control your sounds during performance.

**Pin**

Plugins are connected by connecting virtual wires to their pins (also called plugin ports). Plugins may have one or more input/output pins. MIDI pins are colored orange and audio pins are colored blue. See the chapter [Wiring view](#) for more information on plugin pins.

**Plugin**

Also referred to as a plugin block or just "block". A package of functionality that can be employed by any compatible audio host software. There is a tremendous variety of plugins available in the audio/music marketplace, but many popular plugins are software emulations of physical instruments or hardware devices. Originally developed for use within DAW (**D**igital **A**udio **W**orkstation) applications, some plugins (called "virtual instruments") generate audio, for example, emulations of musical instruments and synthesizers, while effects plugins alter incoming audio signals, such as from a microphone, electric guitar, or virtual instrument, offering processes like equalization, filtering, compression, limiting, reverb, delay, flanging, chorusing, phasing, and the like.

**Tip:** Don't mix plugin formats from the same company. ([click here](#) to learn more)

**Port**

An alternate name for a MIDI input or output device.

**Probabilistic Sound Designer**

A tool in Gig Performer that is available for all your plugins to help you create new sounds.

**Program Change**

Also referred to as a "patch change" (or "PC", when talking about MIDI messages; don't confuse this with "computer"). These messages are commonly used to prompt hardware or software to change a preset (or "patch"). If General MIDI is being used, the program change number indicates a particular sound. Since this standard isn't obligatory, manufacturers may list patches in any order they wish. See [this blog article](#) to learn how to manage program change events in Gig Performer.

**Rackspace**

A collection of interconnected plugins, along with one or more panels containing widgets to control plugin parameters in real time. A rackspace can actually be pretty much anything you like: It could represent a single sound or all the elements you need for an entire song - or even just one part of a song. You can switch from one rackspace to another instantaneously - and with no interruption of sound - either from your laptop, with up/down pedals, or via MIDI program change messages. [Click here](#) to learn more.

**Radio group**

Allows you to group multiple widgets where only one widget in the group is turned on at a time. Supported widgets are buttons, LEDs, switches, and the Sustain pedal. You can define up to 99 radio groups. A widget cannot be a member in both radio group and a widget link group. To learn more about this feature, visit the chapter [How to add a widget to a Radio Group](#).

**Rig**

In Gig Performer, a "Rig" is a collection of three sets of information: a set of physical MIDI controllers, a set of virtual controllers that can be attached to widgets, and a map that relates the two sets, thereby connecting your physical controller, through widgets, to parameter controls.

**Rig Manager**

A mechanism that allows you to quickly change your connected MIDI hardware so your rackspaces are not dependent on specific devices. The Rig Manager is also used to help you distinguish multiple identical MIDI controllers.

**Scriptlet**

A proprietary plugin in Gig Performer that executes GScript code to implement

specialized MIDI transformations. [Click here](#) to see the list of community-created scriptlets.

### **Setlist**

A collection of songs arranged in a desired order. Setlists are assembled from the library of available songs; you might have one setlist for a set you perform regularly and other setlists for individual performances requiring different songs or a different order. A song in a setlist is made active simply by clicking on it.

### **Song**

A collection of song parts arranged in a desired order. Songs can execute actions when they are made active, such as sending out a MIDI program change message, overriding the current tempo with the song tempo, or opening a ChordPro file showing chords and lyrics. A library of songs can be built up, making them available to be used in setlists.

### **Song Part**

An individual section of a song, such as a verse, chorus, bridge, or interlude. A song part consists of a reference to a rackspace along with some other properties. Selecting a song part makes its associated rackspace variation active, and can perform other actions like overriding the current tempo or sending out a custom string of MIDI messages. As the lowest level building block of setlists, song parts can be easily accessed by MIDI program change commands from an external controller, the large song part tiles in the Setlist/Song view, or by song part tiles in the Gig Performer Lemur template (or another custom OSC template).

### **Variation**

These are essentially rackspace presets, all containing the same plugins and interconnections, but with different widget settings. For example, you can create a rackspace to add phasing to a piano sound, and then add several variations, each with different degrees of phase intensity. Since variations actually store widget settings, every parameter you want stored in a variation must be assigned to a widget on a rack panel. As with rackspaces, you can switch from one variation to another instantaneously - and with no interruption of sound - either from your laptop, with up/down pedals, or via MIDI program change messages.

### **Widget**

A virtual object (such as a knob, slider, button, or meter) used to control and/or display plugin settings. Widgets can be operated either with your laptop's mouse or trackpad, or remotely from a MIDI controller. With the use of OSC, widgets can also be operated from

iOS / Android tablets or smartphones.

### Linked widgets group

A collection of widgets that operate ganged together, where changing one widget in the group changes all of the others in the group, as well. This is a powerful method for controlling multiple plugin parameters simultaneously. For example, you could create several slider widgets that control the volume of various plugins and then raise or lower the volume of all of them by moving any one of them. In addition, individual widget scaling allows simple implementation of effects such as cross-fading. For more information, see the [Widget Properties Inspector](#) chapter.

### Wiring View

The Wiring view is where you create your plugins and connect them together.

## 24.3 File extensions used in Gig Performer

In this section find out more about file extensions that are used in Gig Performer.

File extension	Description
<b>gig</b>	a gig file contains all the data you have created: rackspace, variations, and all widgets/widget groups
<b>gpaudiosettings</b>	exported Gig Performer audio settings are stored in this format
<b>gpc</b>	this extension indicates a compiled GPCScript file
<b>gpfav</b>	used for storing the plugin or selection of plugins saved as favorites
<b>gppanel</b>	a Gig Performer panel template file extension
<b>gpp_audiounit</b>	saved presets of AU plugins are stored in this format
<b>gpp_internal</b>	saved presets of built-in plugins are stored in this format
<b>gpp_vst</b>	saved presets of VST plugins are stored in this format
<b>gpp_vst3</b>	saved presets of VST3 plugins are stored in this format
<b>gppsd_audiounit</b>	Probabilistic Sound Designer file extension addressing AU plugins
<b>gppsd_vst</b>	Probabilistic Sound Designer file extension addressing VST plugins
<b>gppsd_vst3</b>	Probabilistic Sound Designer file extension addressing VST3 plugins



<b>gpscale</b>	stores the widget's curve scaled value
<b>gpsettings</b>	exported Gig Performer settings are stored in this format
<b>gpsong</b>	songs are stored in this format
<b>rackspace</b>	this format is used to store the exported rackspace
<b>rigsetup</b>	the Rig Manager setup is exported to this file format
<b>userkeywords</b>	exported user keywords from the Plugin Manager are stored in this format

## 24.4 File locations in Gig Performer

In this section find out more about default file locations that are used in Gig Performer.

MacOS	Windows	Description
/Applications	C:\Program Files\Gig Performer 5	Installation folder
/Users/Shared/Gig Performer/Extensions	C:\Users\Public\Documents\Gig Performer\Extensions	third-party extensions folder
/Users/Username/Documents/Gig Performer/Favorites	C:\Users\Username\Documents\Gig Performer\Favorites	Favorites folder
/Users/Username/Documents/Gig Performer/Presets	C:\Users\Username\Documents\Gig Performer\Presets	Presets folder
/Users/Username/Documents/Gig Performer/PSD	C:\Users\Username\Documents\Gig Performer\PSD	Probabilistic Sound Designer folder
/Users/Username/Documents/Gig Performer Song Lyrics-Chords	C:\Users\Username\Documents\Gig Performer Song Lyrics-Chords	Root folder for ChordPro files
/Users/Username/Documents/Gig Performer Recordings	C:\Users\Username\Documents\Gig Performer Recordings	Default master folder for the Recorder. Each set of recordings will be in time stamped sub folders

/Users/Username/Documents /Gig Performer/ScalingCurves	C: \Users\Username\Documents\ Gig Performer\ScalingCurves	Saved scaling curves
/Users/Username/Documents /Gig Performer/Scripts	C: \Users\Username\Documents\ Gig Performer\Scripts	Scripts folder
/Users/Username/Documents /Gig Performer/Rigs	C: \Users\Username\Documents\ Gig Performer\Rigs	Exported rigs folder
/Users/Username/Documents /Gig Performer/AudioSettings	C: \Users\Username\Documents\ Gig Performer\AudioSettings	Exported audio settings folder
/Users/Username/Documents /Gig Performer/GlobalOptions	C: \Users\Username\Documents\ Gig Performer\GlobalOptions	Exported all options folder
/Users/Username/Documents /Gig Performer/CachedAudioFiles	C: \Users\Username\Documents\ Gig Performer\CachedAudioFiles	Default folder where cached audio files are stored

## 24.5 Cross-Platform Usage

Files created by Gig Performer can be opened by both macOS and Windows computers. If the computer you're using to access the file is missing some of the plugins that were present when the original file was created, Gig Performer will create a "chameleon" plugin that mimics all the properties of the original plugin except the actual processing. This will allow you to modify the file, save it and later re-open it on a computer that has all required plugins installed.

Similarly, if the audio interface connected to the current system does not have enough inputs or outputs to accommodate the setup utilized by the file, Gig Performer will prompt you to create "Virtual channels" that produce no input or output but retain all connections. Doing so will allow you to modify the file, save it and later reopen it on a computer that is connected to the original audio interface.

If you plan to use Gig Performer on both macOS and Windows, you may want to consider just using VST plugins on macOS rather than AU plugins. This will make it easier for you to use the same gigs on both platforms, assuming of course you have the plugins themselves on both platforms.

For more information, please refer to this blog article [Cross-platform usage and moving from one platform to another](#).

## 24.6 Troubleshooting

In this chapter we present you with the most common troubleshooting tips. Also make sure to check out our dedicated blog article [Tips to troubleshoot your computer-based setup](#).

### No sound

Test your audio system to make sure it is receiving signal from your connected audio interface.

Make sure one or more plugins have connections from their audio output(s) to one or more audio input(s) on your connected audio interface.

If you have multiple audio interfaces connected to your system, make sure the correct one is selected in [Audio Options](#).

If you are running Gig Performer on Mac OS X 10.14 or above, you will be prompted to allow access to the "microphone"; check the [Before you start](#) chapter to learn more.

### Gig Performer isn't finding my plugins

Normally Gig Performer scans only certain "standard" folders for plugins, such as (in macOS systems) Library/Audio/Plug-ins/VST or (in Windows systems) C:\Program Files\VstPlugins. However, some plugin and DAW manufacturers install their plugins in different folders. If Gig Performer isn't finding your plugins, that's the likely cause. To instruct it to scan different folders, open the [Plugin Manager](#) and click on the *Manage* button in the bottom left-hand corner, then select *Set VST folders for scanning...* and *Set VST3 folders for scanning...* In these dialogs you can easily add or delete folders to the list of those scanned by Gig Performer.

If you are a Mac user and your Mac uses the new M1, M2, or newer processors, please check [this blog article](#) to learn more how to resolve possible problems with missing plugins.

### Audio interface does not appear in the Wiring view

In most cases, Gig Performer will automatically detect all audio interfaces connected to your system. If it does not, check to make sure that the interface is connected and

powered on, and relaunch Gig Performer. If your audio interface still does not appear, take the following steps:

- **MacOS:** start up the Audio MIDI Setup app in the Utilities folder and make sure that your interface appears in the list of input and output devices on the left-hand side of the Audio Devices window. If it does not appear, consult the documentation for your interface.
- **Windows:** go to Control Panel -> Hardware and Sound -> Sound and make sure that your interface appears in the list of playback and recording devices. If it does not appear, consult the documentation for your interface.

### **MIDI controller does not appear in the Wiring view**

In most cases, Gig Performer will automatically detect all MIDI controllers connected to your system via USB. If it does not, check to make sure that the controller is powered on, and relaunch Gig Performer. If your MIDI controller still does not appear, take the following steps:

- **MacOS:** go to Utilities -> Audio MIDI Setup and select Show MIDI Studio from the Window menu, then make sure that your controller appears in the window and is not grayed out. If it is grayed out or does not appear, consult the documentation for your controller.
- **Windows:** press the key combination **Win key + X** and then select the Device Manager; make sure that your controller appears in the "Sound, video and game controllers" category. If it appears with a yellow question mark or exclamation point, right-click and choose the Update Driver Software option. If it does not appear, consult the documentation for your controller.

### **Widget not working**

Double-click on the widget in Edit mode to open the [Widget Properties inspector](#) and make sure that it is assigned to a plugin and a plugin parameter.

**Note:** If your question is not answered here, please check our [Gig Performer Resources](#). If your problem is still not addressed, feel free to open a new thread in our [Community forums](#) or reach out to us through our [support page](#).

## 24.7 OSC messages

In this section learn more about Gig Performer OSC command messages. The messages can be sent to Gig Performer which will then respond appropriately, perhaps also returning an OSC message or bundle with information.

### Non-realtime OSC messages

Address	Arguments	Description
/GigPerformer/GetSetLists		Returns a list of available setlists
/GigPerformer/SelectSetList	Integer	Switch to a specific set list
/SetList/GetSongList		Returns a list of available songs in the current setlist
/SetList/GetSongParts		Returns a list of available song parts for the current song
/Song/GetSongParts		Returns a list of available song parts for the current song
/Window/ToFront		Bring the main GP window to the front
/Window/ToBack		Send the main GP window to the back
/Window/FullScreen	Integer(0, 1)	Make GP be fullscreen
/Actions/PanelView		Switch to the Panel view
/Actions/WiringView		Switch to the Wiring view
/Actions/SetlistView		Switch to the Setlist view
/Actions/ShowGlobalRackspace	Integer (0, 1)	Show/hide the Global rackspace
/Actions/ToggleGlobalRackspace		Toggle the Global rackspace

/GigPerformer/SwitchToProgramNumber	Integer(program number)	Switch to a specific rackspace using program number
/GigPerformer/SwitchToProgramNumber	Integer(MSB), Integer(LSB), Integer(Program number)	Switch to a specific rackspace using bank select values and program number
/Tap		Emulate tap tempo
/SetMasterFader	Double (0.0 .. 1.0)	Set the master fader volume
/SetGlobalTranspose	Integer	Set the global transpose value
/GetGlobalTranspose		Returns the current global transpose value
/RecordInputs	Integer(0, 1)	Enables or disables recording
/GetRecordStatus		Returns whether recording is enabled
/GetAbletonLinkStatus		Returns whether Ableton Link is enabled
/SetAbletonLinkStatus	Integer(0, 1)	Enable or disable Ableton Link
/SetBPM	Double(bpm)	Set the global BPM
/AdjustBPM	Double(offset)	Adds the offset to the current global BPM
/GetCurrentBPM		Returns the global BPM
/Play	Integer(0, 1)	Enable or disable the global playhead
/Panic		Emulate MIDI Panic
/RackSpace/ListOSCPlugins		Returns the list of OSC enabled plugins
/RackSpace/ListOSCWidgets		Returns the list of OSC enabled widgets
/GigPerformer/ListAllRackspaces		Returns a list of all rackspaces

/GigPerformer/ListAllVariations		Returns the list of all variations of the current rackspace
/Refresh		Ask Gig Performer to resend status messages (useful if OSC connection is flakey)
/TunerView/SetValue	Integer(0, 1)	Switch in or out of Tuner View
/SongPart/TakeSnapshot		Take a snapshot of the current widget overrides in a song part
/SongPart/ClearSnapshot		Clear the snapshot, thereby resetting widgets to the original rackspace values

### Realtime OSC messages

Address	Arguments	Description
/SelectSongByIndex		
/GigPerformer/SwitchToSong	Integer	Switch to another song based on its index
/SelectSongPart		Deprecated - use /Song/SwitchToPart
/Song/SwitchToPart	Integer	Switch to another song part
/Song/MoveDown		
/Song/MoveUp		
/GigPerformer/NextSong		
/GigPerformer/PrevSong		
/Song/NextPart		
/Song/PrevPart		
/RackSpace/SetVariation	Integer	Switch to the (zero based) variation

/GigPerformer/SwitchToRack	Integer [Integer]	Switch to the (zero based) rackspace and optionally to a variation in that rackspace
/GigPerformer/SwitchToRackSpace	Integer [Integer]	Switch to the (zero based) rackspace and optionally to a variation in that rackspace
/GigPerformer/SwitchToRackSpace	String [Integer]	Switch to the named rackspace and optionally to a variation in that rackspace
/GigPerformer/MoveDown		
/GigPerformer/MoveUp		
/GigPerformer/NextRackSpace		Switch to the next RackSpace
/GigPerformer/PrevRackSpace		Switch to the previous RackSpace
/RackSpace/NextVariation		Switch to the next variation
/RackSpace/PrevVariation		Switch to the previous variation

**MIDI messages - replace "oscHandle" with the handle of the MIDI In Block**

Address	Arguments	Description
/oscHandle/Midi	1 to 3 MIDI bytes	Send an arbitrary MIDI message - (NB sysex currently not supported)
/oscHandle/NoteOn	Integer(Note number), Integer(Velocity), Integer(1..16)	Send a MIDI Note On message on the given channel
/oscHandle/NoteOff	Integer(Note number), Integer(Release), Integer(1..16)	Send a MIDI Note Off message on the given channel



/oscHandle/Note	Integer(Note number), Integer(Velocity), Integer(1..16), Integer(delay in ms)	Send a Note On message and then later send a matching Note Off message
/oscHandle/CC	Integer(Controller number), Integer(value), Integer(1..16)	Send a CC message
/oscHandle/PP	Integer(Note number), Integer(pressure), Integer(1..16)	Send a polyphonic aftertouch message
/oscHandle/PB	Integer(pitchbend amount), Integer(1..16)	Send a pitchbend message
/oscHandle/AT		
/oscHandle/PC		
/oscHandle/AT	Integer(aftertouch amount), Integer(1..16)	Send an aftertouch (channel pressure) message
/oscHandle/PC	Integer(0..127)	Send a program change message

**Widget control - replace "oscHandle" with the handle of the widget**

Address	Arguments	Description
/oscHandle/SetValue	Double(0.0 .. 1.0)	Set the value of a widget
/oscHandle/SetResetValue	Double(0.0 .. 1.0)	Set the default reset value for a widget
/oscHandle/GetResetValue		Returns the default reset value for a widget
/oscHandle/Reset		Set the widget to its default value (same as double-clicking on widget)
/oscHandle/SetCaption	String	Set the caption of a widget
/oscHandle/GetValue		Returns the current value of the widget

/oscHandle/GetCaption		Returns the current widget caption
-----------------------	--	------------------------------------

**Plugin parameter control - replace "oscHandle" with the handle of a plugin block**

Address	Arguments	Description
/oscHandle/SetParam	Integer(Parameter Number), Double(0.0 .. 1.0)	Set the parameter value of the specified parameter
/oscHandle/SetPreset	Integer	Switch to a different plugin preset (NB some plugins do not support this)
/OSCHandle/LoadGPPreset	<preset name>	Loads the GP preset specified by the preset name
/oscHandle/GetParamCount		Returns the number of parameters in a plugin
/oscHandle/GetParam	Integer(Parameter Number)	Get the value of the given parameter number
/oscHandle/Bypass	Integer(0, 1)	The value 1 will bypass the plugin - the value 0 will unbypass it

**Note:** some of these commands send back a lot of information. You may want to use a tool such as OSCShark or Max to view the response messages.