



Gig Performer

User Guide

v. 1.5.2

April 2017

Written by Howard Massey
for *On The Right Wavelength* professional consulting services
otrw@earthlink.net

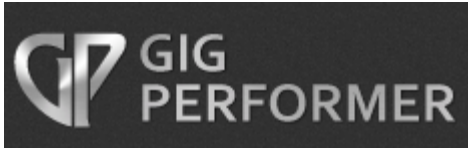
Copyright © 2017 Deskew Technologies, LLC

All rights reserved.

Table of Contents

INTRODUCTION	<u>1</u>
Gig Performer Features.....	<u>2</u>
Basic Terminology.....	<u>4</u>
Before You Start.....	<u>5</u>
A Word About Cross-Platform Usage.....	<u>6</u>
MIDI TUTORIAL	<u>7</u>
AUDIO TUTORIAL	<u>30</u>
OSC TUTORIAL	<u>45</u>
REFERENCE	<u>51</u>
Screens.....	<u>51</u>
Connections View.....	<u>51</u>
Performer View	<u>53</u>
Tuner View.....	<u>54</u>
Edit Mode screen.....	<u>55</u>
Main Menus	<u>56</u>
File	<u>56</u>
Edit.....	<u>58</u>
Options.....	<u>59</u>
General preferences.....	<u>60</u>
Audio I/O.....	<u>62</u>
OSC	<u>63</u>
Global MIDI.....	<u>66</u>
Advanced.....	<u>68</u>
Instances	<u>68</u>
Predictive Loading.....	<u>69</u>
Window	<u>71</u>
Plugin manager	<u>71</u>
MIDI Monitor	<u>73</u>
Help	<u>74</u>

Other Menus and Dialogs.....	75
Widget Properties inspector (Edit mode)	75
Rackspace Properties	78
Variation Properties.....	79
“More” menu.....	80
Supplied Plugins	81
Gain Control.....	81
Gain and Balance	82
Gain and Balance (m->s)	83
MIDI In (OMNI) / MIDI In (OSC).....	84
MIDI Filter	89
MIDI Monitor.....	91
MIDI Out.....	92
Keystroke Shortcuts.....	94
Quick plugin finder	95
Troubleshooting	97
APPENDIX A: OSC	99
APPENDIX B:LICENSE AGREEMENT	106



INTRODUCTION

Welcome to Gig Performer—the world’s most versatile plugin host application for live performance. Gig Performer lets you take complete control of your entire rig and provides instant access to all your instrument and effects plugins—no DAW software necessary.

Whether you’re a keyboard player, a guitarist, a vocalist, a drummer, or a live sound engineer, Gig Performer has something 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 pedalboard, guitarists can freely switch between multiple amp/cabinet models and digital effects in real time without any audio glitching.
- Singers can easily add compression, EQ and other desired effects to their vocals.
- Live sound engineers can use Gig Performer 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.
- You can even run multiple Gig Performer instances simultaneously, allowing different members of a band to access different plugin setups independently!

Sound complicated? It’s not. Thanks to our intuitive interface, you’ll find Gig Performer remarkably easy to use. In fact, you’ll be up and running in no time. Beneath the surface, however, are many advanced features, all covered in the pages of this User Guide—we’ve even included a number of brief [tutorials](#) to get you started. So, without further ado, let’s begin...

Gig Performer Features

- Supports all VST plugins (OSX and Windows) as well as AU (OSX only)
- Allows you to play multiple instruments, layer sounds, and add effects easily
- Easy user interface: simply connect any plugin to any other plugin in any configuration—no confusing busses, sends or inserts
- Automatic mixing of signals when multiple outputs are connected to a single input
- Deactivate any plugin at any time to remove sounds or bypass effects
- Build fully customizable rackspaces consisting of one or more front panels
- 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 via a wide range of widgets such as knobs, sliders, buttons, labels, and meters
- Learn mode makes it easy to assign a MIDI control to a widget and a widget to a plugin parameter
- Widget scaling allows full motion while restricting the parameter range
- LEDs show input/output levels on each channel as well as audio and MIDI activity
- Built-in fullscreen chromatic tuner
- Create multiple independent running instances for different members of the band or for when you want to control some plugins while leaving others alone
- Runs on both Mac (OSX) and Windows platforms so you can create, share, and modify your files on any computer, even if some plugins are missing
- Minimal CPU usage

Gig Performer User Guide v 1.5.2

- Support for multiple audio interfaces
- 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
- OSC support for remote control via tablets and smartphones
- Support for MaxMSP and Bidule for the development of complex interactive control algorithms

Be sure to check <http://www.gigperformer.com/features-and-benefits.html> for a full list of current features.

Basic Terminology

Connections View: The Gig Performer screen in which you can connect plugins to one another. A collection of interconnected plugins is also sometimes known as a *plugin architecture*.

Gig: A Gig Performer file containing all the data you have created—rackspaces, variations, and all widgets and widget groups. (See below)

Instance: An iteration of an app. Gig Performer supports multiple instances, allowing you to launch it several times in one computer, with each instance operating simultaneously but independently. For example, you can use instances to give different band members the ability to change the rackspace affecting their particular instrument without changing the rackspaces affecting different instruments. [Click here](#) for more information.

MIDI: An acronym for Musical Instrument Digital Interface—a standard protocol for the interchange of musical information between computers, controllers, and hardware-based electronic musical instruments such as synthesizers and samplers. (See below)

MIDI controller: A physical device such as a keyboard or pedalboard typically connected to your computer via USB.

OSC: An acronym for Open Sound Control—a communications protocol that is optimized for modern networking technology. [Click here](#) for more information.

Panel: In Gig Performer, a virtual screen space (visually similar to hardware rack-mounted sound modules and signal processors) containing all the widgets needed to control the plugins in a rackspace. A rackspace can contain multiple panels, each containing one or more widgets.

Performer View: The Gig Performer screen containing the panels and widgets used to control your sounds during performance.

Plugin: A software emulation of a physical instrument or hardware device. Originally developed for use within DAW (Digital Audio Workstation) applications, plugins come in two varieties. *Sound generating plugins*, as their name implies, generate audio. These include emulations of musical instruments and synthesizers. *Effects plugins* alter incoming audio signal, such as from a microphone or electric guitar, enabling processes like equalization, filtering, compression, limiting, and the addition of reverb, delay, flanging, chorusing, phasing, and the like.

Gig Performer User Guide v 1.5.2

Rackspace: A collection of interconnected plugins, along with one or more panels containing widgets to control various 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.

Variation: These are essentially rackspace presets. Variations contain the same plugins and interconnections but with different parameter settings. For example, you can create a rackspace to add phasing to a piano sound, then add several variations, each with different degrees of phase intensity. 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.

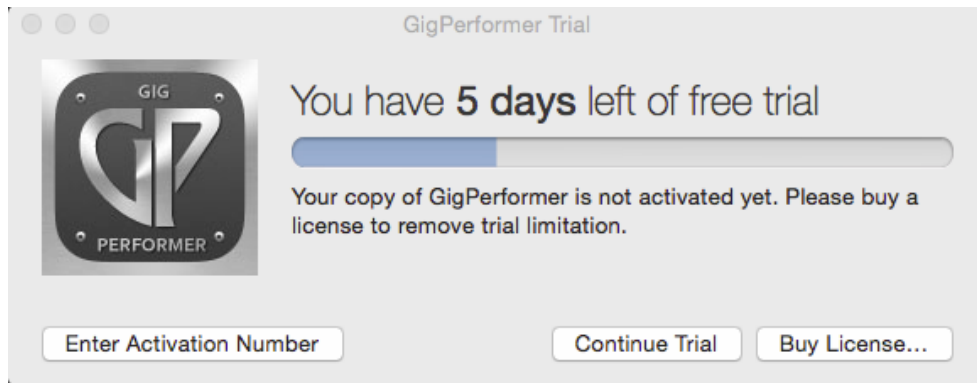
Widget: A generic term for a virtual object (such as a knob, slider, button, or meter) that is used to control and/or display plugin settings. Widgets can be “moved” either with your laptop’s mouse or trackpad, or remotely from a MIDI controller. With the use of OSC, widgets can also be “moved” from iOS / Android tablets or smartphones.

Widget group: A collection of widgets that control 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 them by moving any one of them. In addition, individual widget scaling allows simple implementation of effects such as cross-fading.

Before You Start

Before launching Gig Performer, **make sure that all audio and MIDI interfaces connected to your system are powered on.** If no audio interface is connected, Gig Performer will use your computer's built-in input and output. If you have more than one interface connected, you'll be able to choose among them after the app launches. Note that you can use different interfaces for input and output if desired.

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

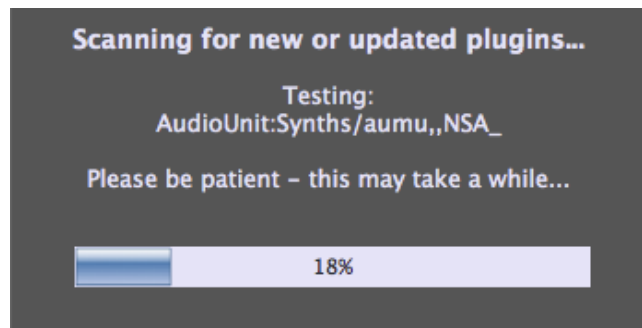


If you have purchased a license, click on Enter Activation Number and type or paste in the activation code emailed to you. (Make sure your anti-spam blocker doesn't prevent the confirmation email with your license from being received).

If you have not yet purchased a license, click on Continue Trial to access a fully operational trial version of Gig Performer for a period of 14 days. (The dialog shows you how many days you have remaining.). You can purchase an activation code at any time by clicking on the Buy License... button.

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

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



Gig Performer User Guide v 1.5.2

When running Gig Performer on Mac (OSX) systems it will scan for both VST (Virtual Studio Technology) and Audio Unit (AU) plugins; when running on Windows systems it will look for all VST plugins. **Note:** If you are a Windows user, you will have to tell Gig Performer what folders to search, since there are no default locations where all plugins are guaranteed to be installed. To do so, go to the Windows menu, then select “Plugin manager.” In the resultant window, click on Manage, then choose the “Set VST folders for scanning” option. We suggest you begin by having Gig Performer search in the folder C:\Program Files\VstPlugins. If you have additional plugins stored in other folder locations (as some plugin and DAW manufacturers automatically create), the Plugin manager allows you to easily add other folders to scan. [Click here](#) for more information.)

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 later disable this scan completely with an [option](#) in the Preferences dialog.)

Gig Performer does not include any synth or effects plugins because there are so many available plugins out there—both free and commercial—we’re sure that most users will want to use their own favorites. ([Click here](#) for a list of some of our favorite free ones.) However, [it does come with a number of specialized plugins](#) to handle basic MIDI and audio operations.

A Word About Cross-Platform Usage

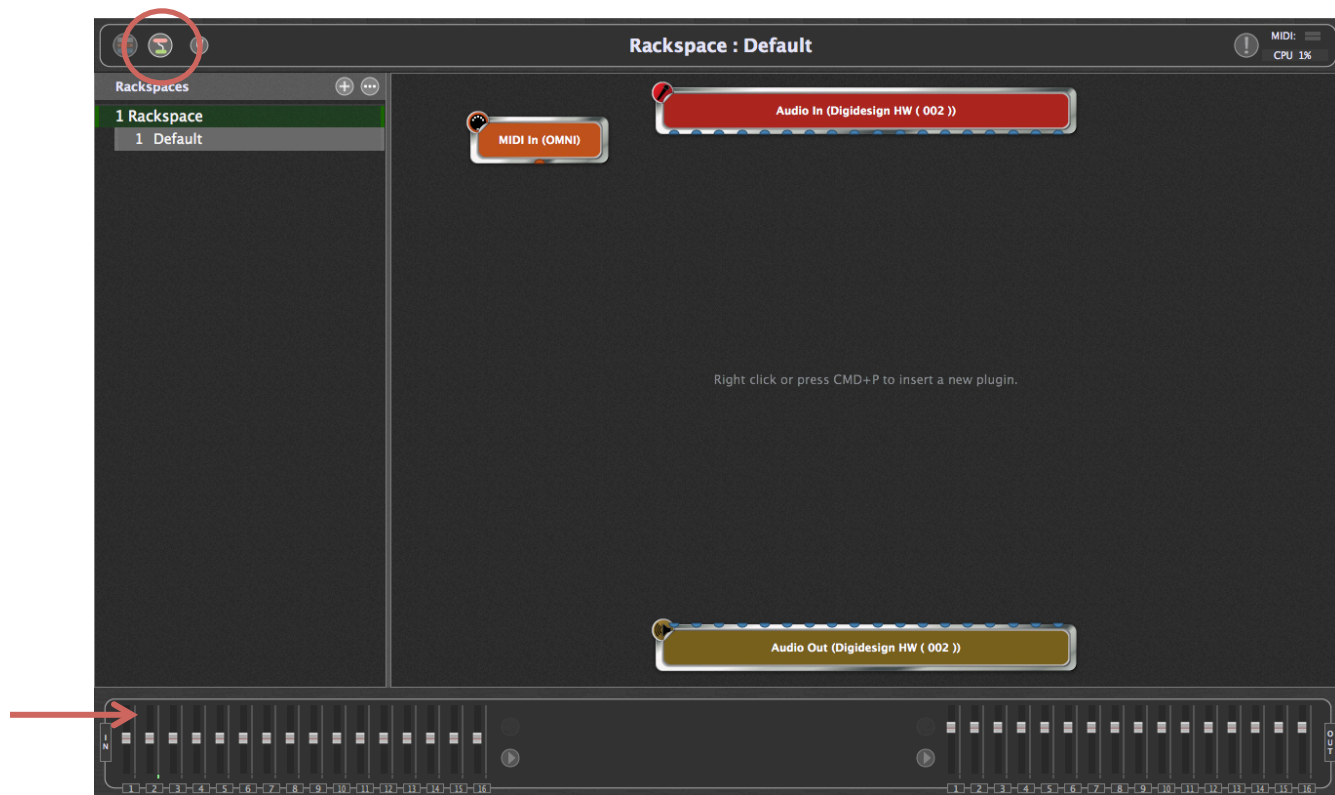
Files created by Gig Performer can be opened by both Mac (OSX) and Windows computers. If the computer you’re using to access the file is missing some of the plugins that were present when the original file was created, Gig Performer will create a “chameleon” plugin that mimics all the properties of the 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.

MIDI TUTORIAL

This tutorial will show you how to use Gig Performer to play plugins from your MIDI controller.

1. **Launch Gig Performer**, then click on the Connections View icon (circled in red) to enter Connections View:

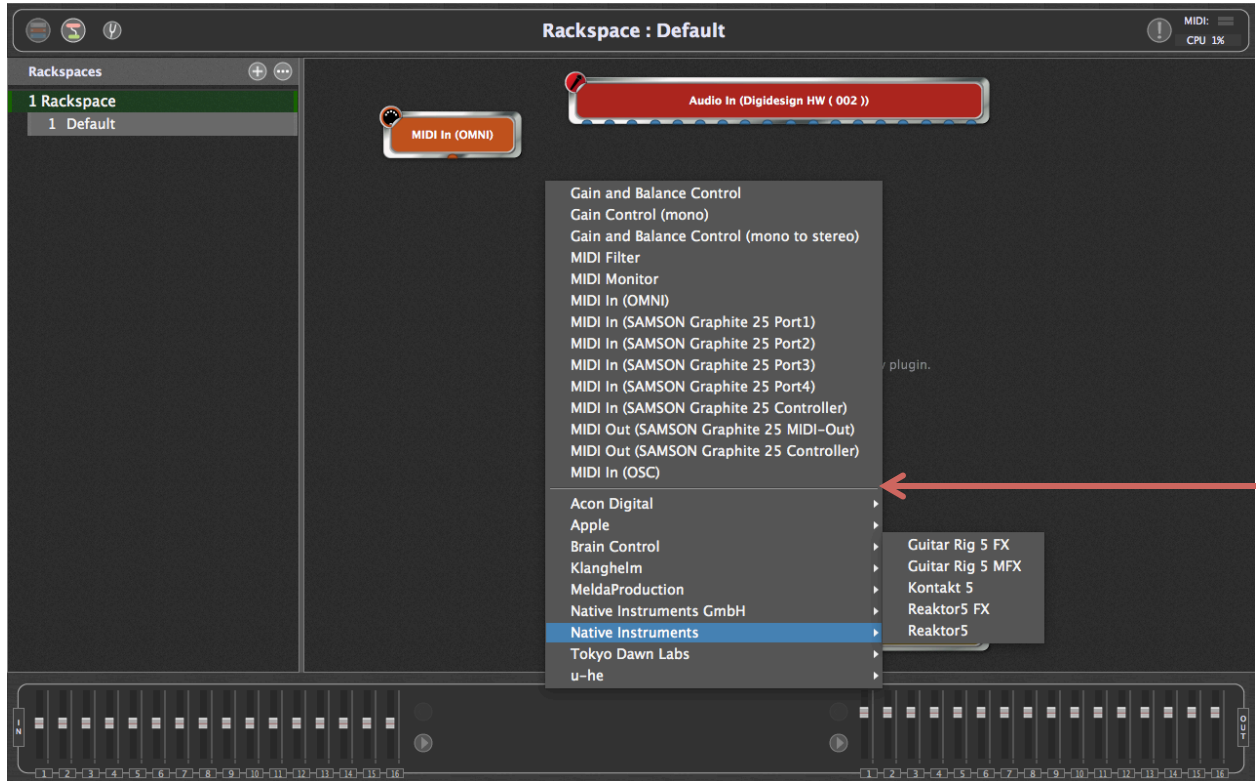


This screen allows you to insert plugins and connect them to other plugins any way you like. (**Note:** Your screen may look slightly different from the one shown above, since the audio input and output blocks will reflect the name of your connected audio interface as well as the number of inputs and outputs it provides.)

The orange block at top left represents the MIDI input to your system, typically from a MIDI keyboard or other MIDI controller. The red block at the top represents the input of your audio interface, typically from microphones, instruments connected via DI (such as electric guitars), and line-level sources. The brown block at the bottom represents the output of your audio interface, and the panel beneath it (indicated by the red arrow) provides metering and virtual faders for each of your audio interface's inputs and outputs. (Flickering in the input meters indicates presence of signal.) If no audio interface is connected, Gig Performer will use your computer's built-in input and output. If an audio interface is connected but does not appear, see the "Troubleshooting" section in this User Guide.

Gig Performer User Guide v 1.5.2

- Choose a plugin.** Right-click (or two-finger tap if you're using a trackpad) anywhere in the main screen to bring up the plugin selector menu, which allows you to choose the plugin you want to access. The built-in Gig Performer plugins (including MIDI In and MIDI Out blocks for each connected MIDI controller) appear above the line separator (indicated by the red arrow), while all other plugins appear under the separator, arranged alphabetically by the name of the plugin manufacturer:



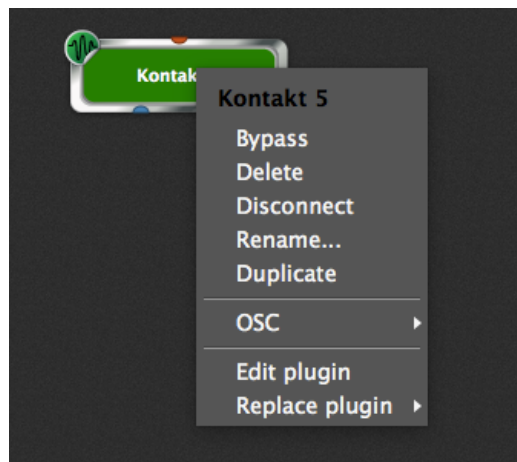
TIP: You can also insert a plug-in by using the CMD-P keyboard shortcut. This option is actually preferable since it opens a “Quick plugin finder” window that allows you to search for a particular plugin by name. [Click here](#) for more information.

As soon as you select a plugin block (or double-click on it), Gig Performer opens that plugin's editor window, allowing you to alter the sound. (You can disable the opening on selection action with an option in the [Preferences dialog](#).) Let's start with a Hammond organ plugin. For this example, we have selected “C3 Blues” from the Native Instruments' Kontakt 5 Player (you can choose your favorite). Its editor window is shown on the following page:

Gig Performer User Guide v 1.5.2



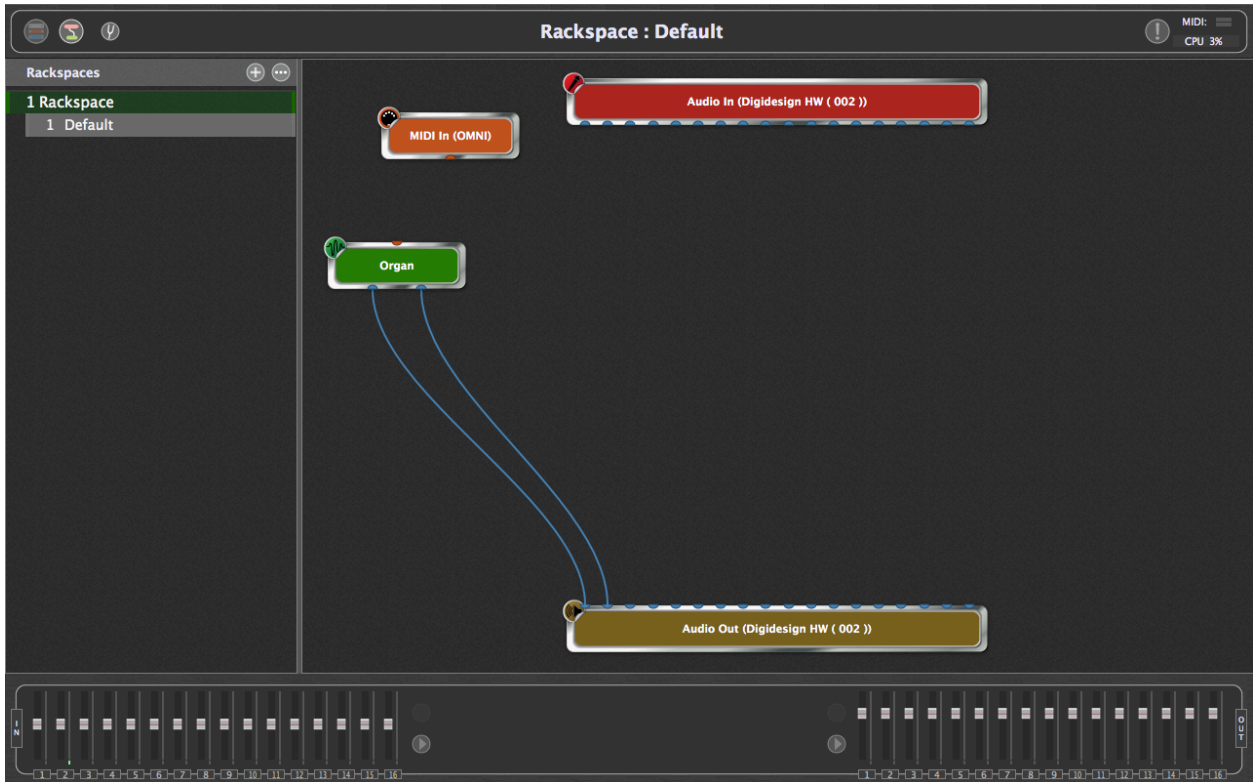
- 3. Give your plugin a custom name.** Close the plugin editor window to return to Connections View. Sound generating plugins such as the one you selected appear as green blocks. (Effects processing plugins, as we will see in the next tutorial, appear as blue blocks.) By default, the name of the plugin is displayed although you can easily rename it with something that represents the actual sound—very useful for distinguishing between multiple copies of the same plugin. Accordingly, right-click on the green block and select the Rename... option, then type in the name “Organ.”



Gig Performer User Guide v 1.5.2

- 4. Set the audio output.** Every plugin block has several pins both above and below it. These pins are used to make interconnections—both MIDI and audio—to other plugin blocks. Orange pins are for MIDI signals and blue pins are for audio signals. Pins at the top provide input to the plugin, while pins at the bottom send output from the plugin. If you connect the outputs of multiple plugins to the input of another plugin, the signals will be mixed automatically.

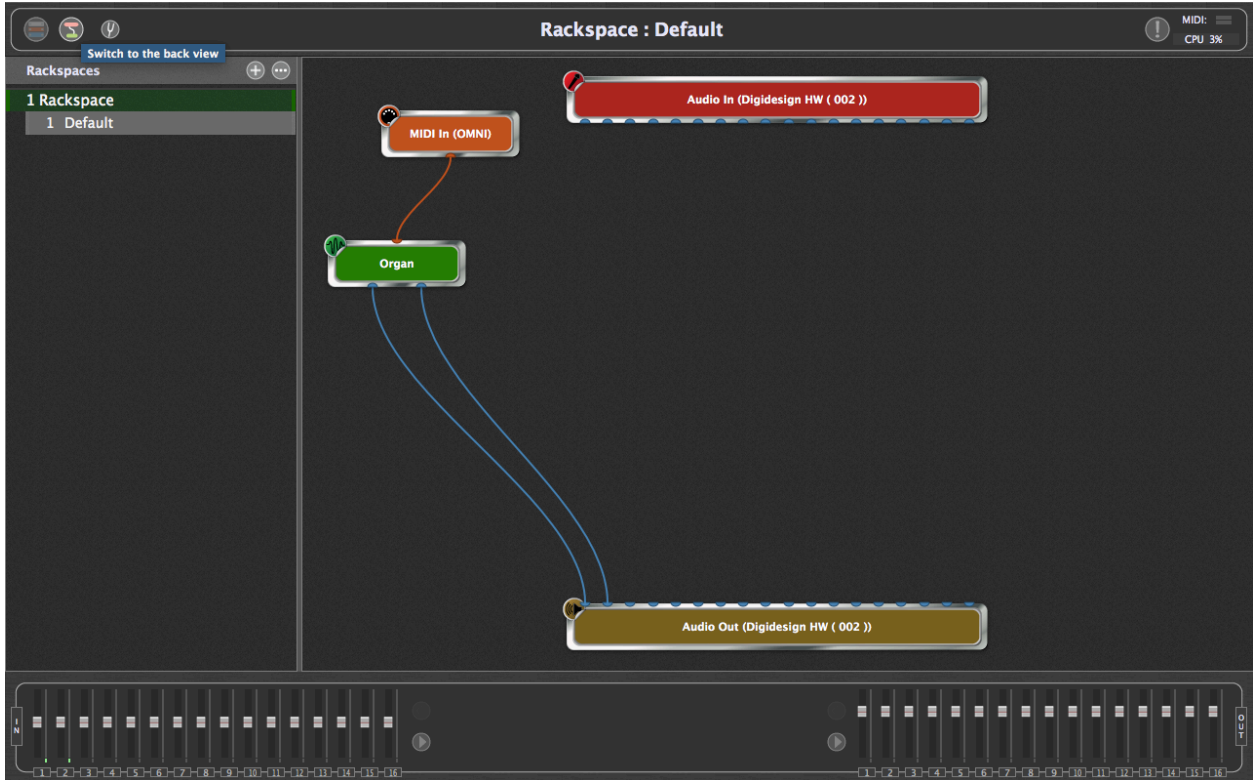
The Kontakt 5 Player creates a stereo audio signal, so there are two blue pins on the bottom. To be able to hear its sound, we need to connect its output to two inputs of the audio interface. To make these connections, simply click on an output pin and, while holding down the mouse button (or keeping your finger on the trackpad), drag the cursor over the desired input pin and then release (unclick) the mouse. ([Click here](#) for a shortcut method of making stereo connections.) Audio connections between plugins are represented by blue lines.



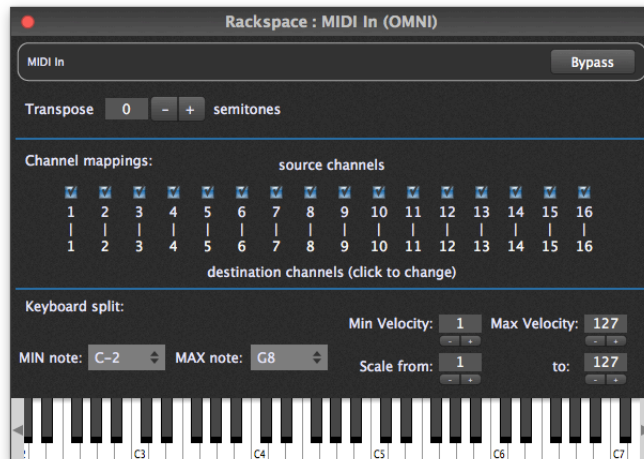
TIP: Connections and plugin blocks can be moved anywhere in the screen by clicking and dragging on them. To remove (disconnect) a connection, right-click on it and select “Remove this connection” or simply click and drag the connection until it detaches from a plugin and then let go. To remove a plugin block, right-click on it and select “Delete.” To remove all connections from a plugin block, right-click on the block and select “Disconnect.”

Gig Performer User Guide v 1.5.2

5. **Set the MIDI input and play your sound.** Using the same click and drag technique, make a connection between the orange output pin on the bottom of the “MIDI In (OMNI)” block and the orange input pin on the top of your plugin. MIDI connections between plugins are represented by orange lines.

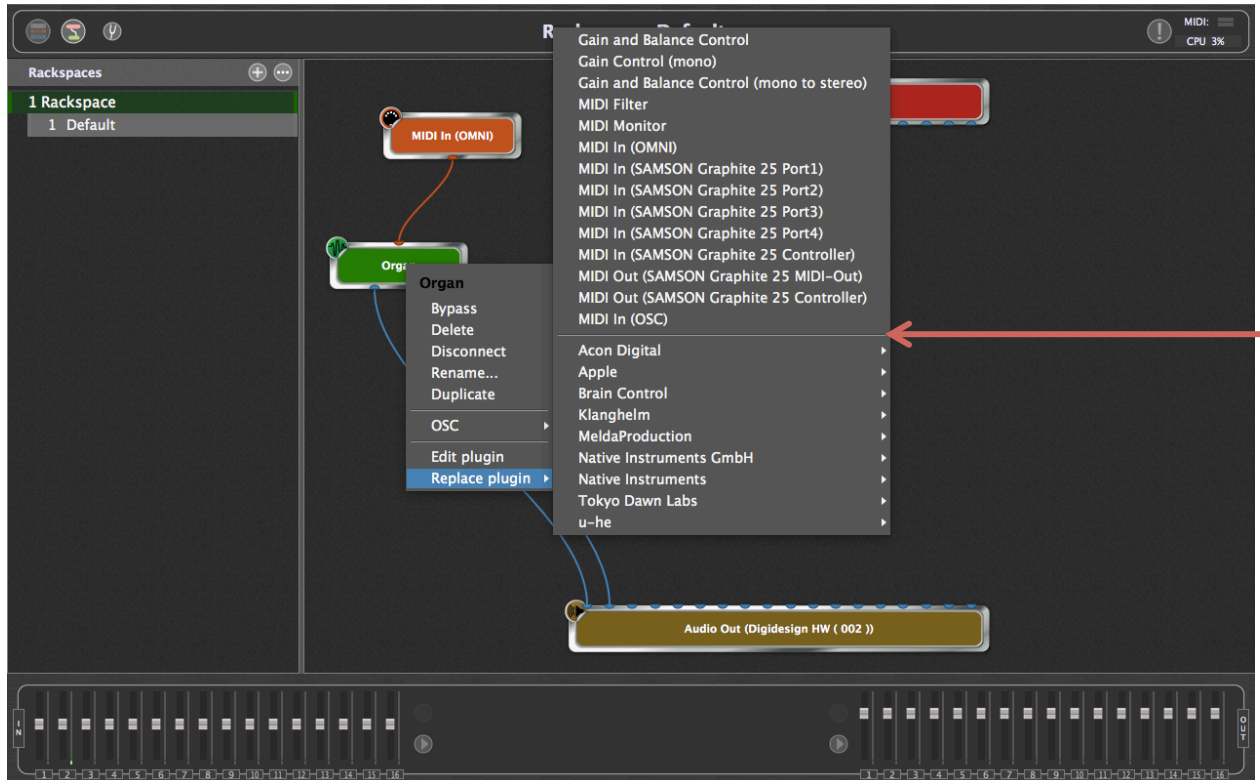


The “MIDI In (OMNI)” plugin is included with Gig Performer and is the default MIDI input until you specify a controller-specific one ([see next page](#)). In addition to passing MIDI signal generated by any connected MIDI controller, it also provides a virtual keyboard so you can play your plugins even if you are not using an external MIDI controller.



Gig Performer User Guide v 1.5.2

At this point, if you play a few notes on your controller (assuming your audio system is turned up), you should now be hearing an organ sound coming from your connected audio interface. Alternatively, you can double-click on the “MIDI In (OMNI)” block to open its editor, then “play” the organ sound by clicking notes on the virtual keyboard. You can also opt to substitute [a plugin representing your particular MIDI controller](#) (thus allowing the plugin to be played only by that controller and/or via a specific MIDI port offered by that controller) by right-clicking on the “MIDI In (OMNI)” block and selecting the “Replace Plugin” option.

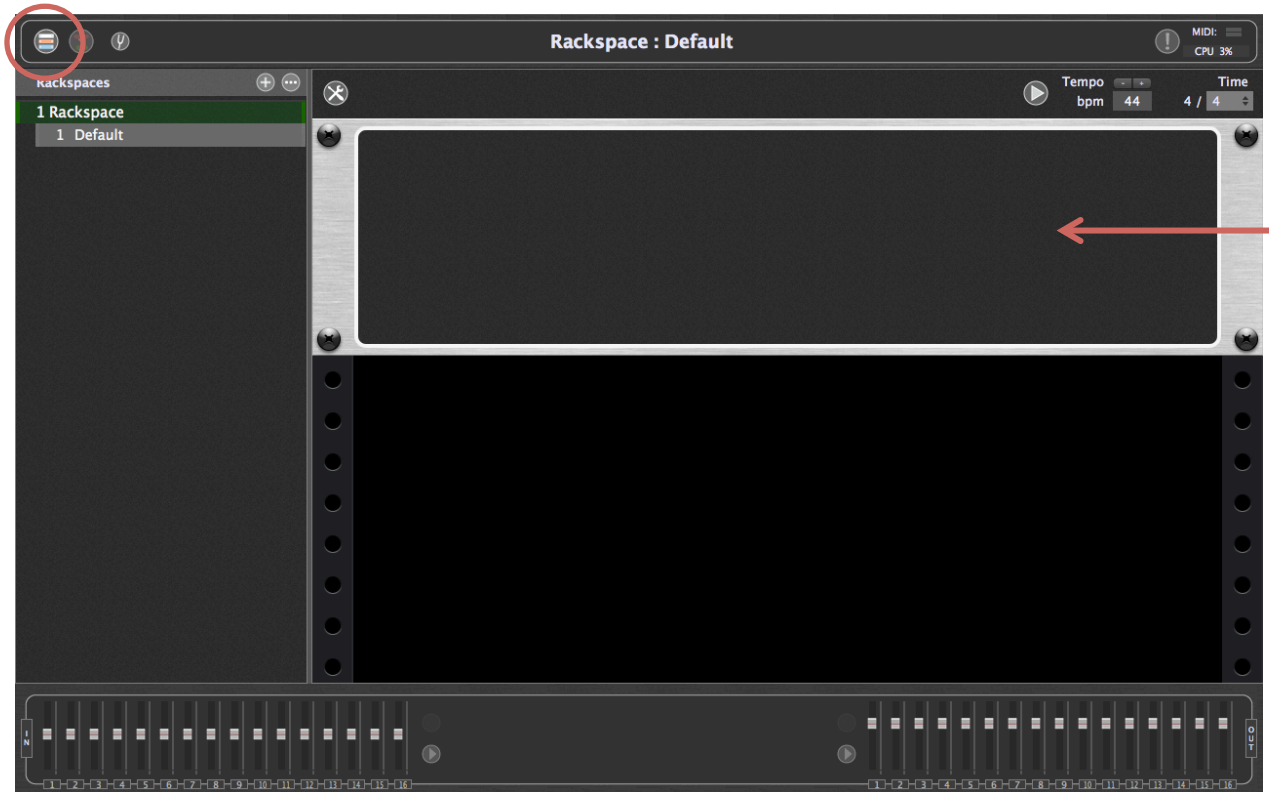


Again, all plugins that represent connected MIDI controllers (such as the “SAMSON Graphite plugins shown here), along with the built-in Gig Performer plugins such as “MIDI In (OMNI),” appear above the line separator (indicated by the red arrow).

For more information on the “MIDI In (OMNI)” plugin, see the [Supplied Plugins](#) section in this User Guide.

Gig Performer User Guide v 1.5.2

- Go to Performer View.** Click on the top left icon (circled in red) to switch Gig Performer to Performer View.

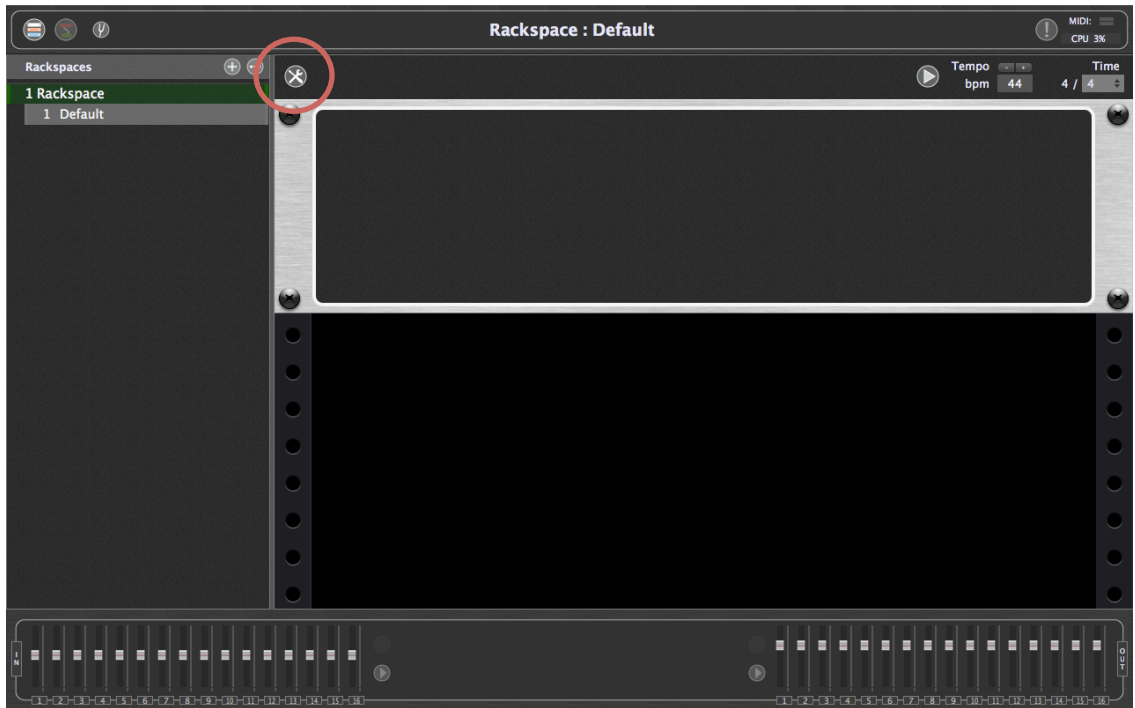


This screen, which is designed to look like a rack of one or more devices, allows you to create widgets (virtual knobs, sliders, buttons, etc.) that can be used to control various aspects of the sound created by your plugins. These widgets can be adjusted manually from your laptop and/or via physical controls (such as real knobs, sliders, and buttons) on your MIDI controller; with the use of an OSC app, they can also be adjusted remotely from iOS / Android tablets or smartphones. (See the [OSC tutorial](#) in this user guide for more information.)

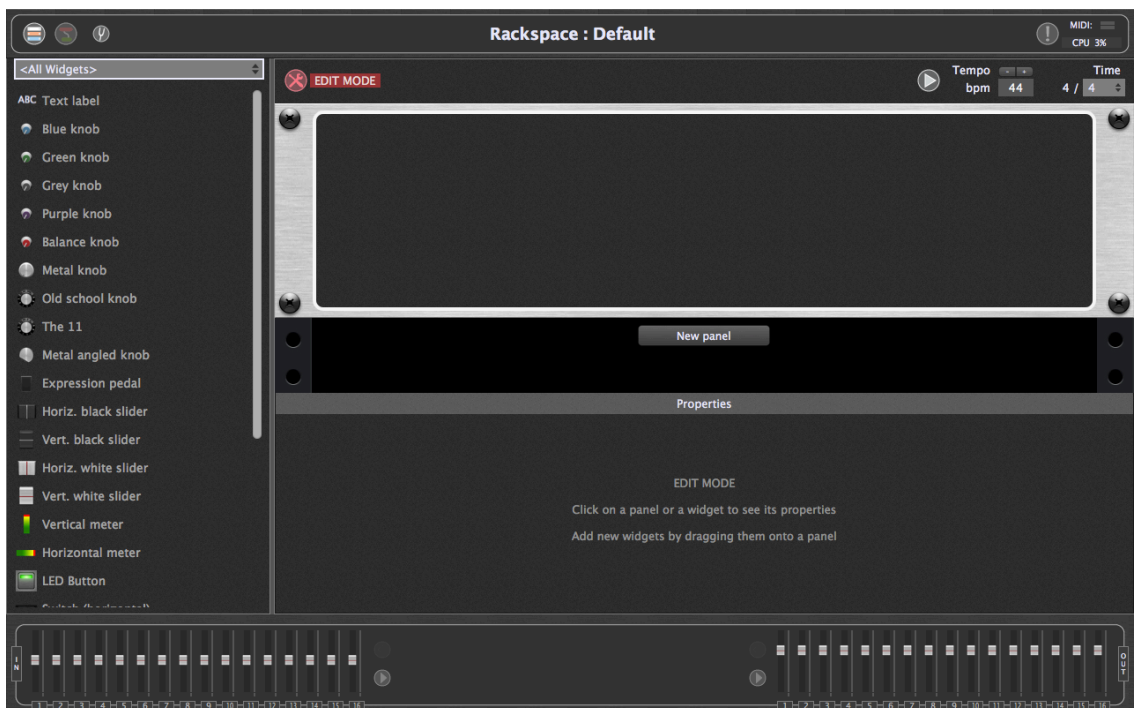
When you switch to Performer View for the first time, you will see a Rackspace containing a single empty panel (indicated by the red arrow) into which you can insert widgets. (Extra panels can be added to group your widgets as desired; more about this [shortly](#)). You can create multiple Rackspaces—each representing a collection of interconnected plugins—and you can switch from one rackspace to another manually from your laptop or via MIDI program change messages (with optional bank switching messages), or by using UP/DOWN commands from a MIDI pedalboard controller. You can also switch rackspaces with OSC commands (see [Appendix A](#) in this User Guide for more information).

Gig Performer User Guide v 1.5.2

7. **Create a widget.** Click on the Widget Layout Designer icon (circled in red) to enter Edit mode.

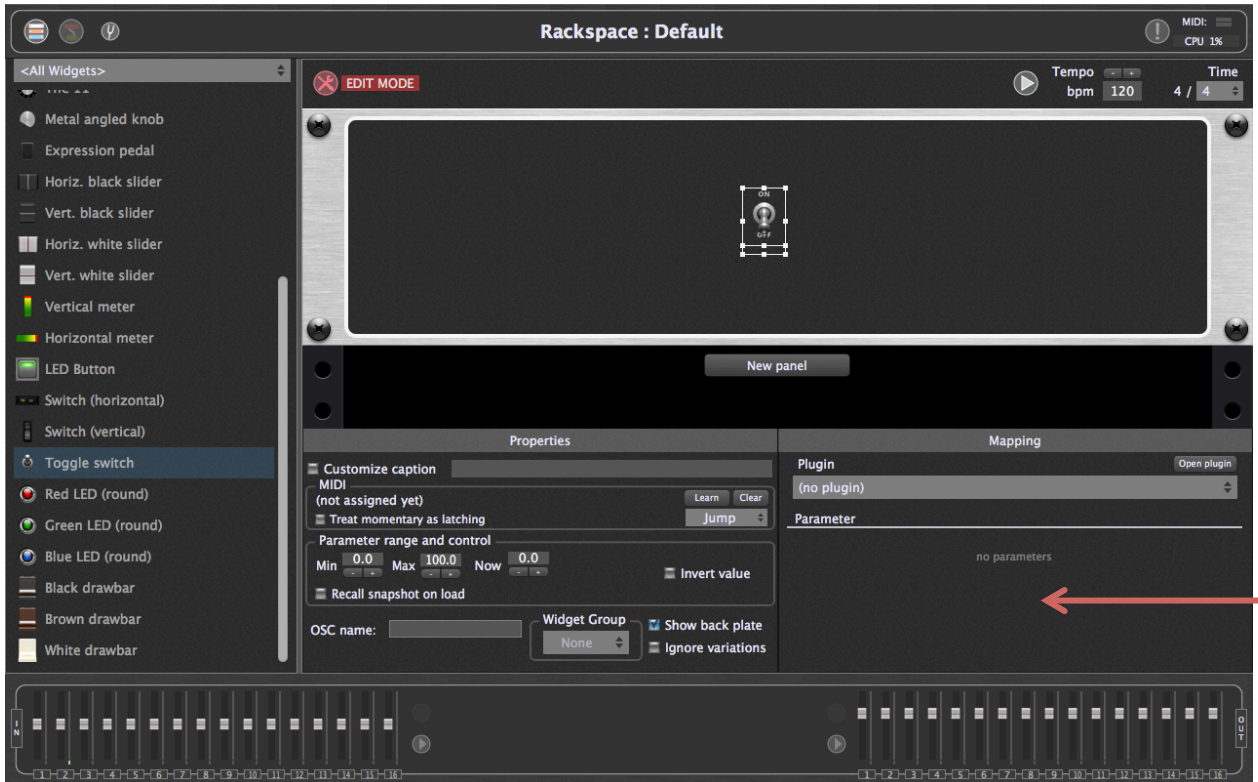


In this screen, the list of rackspace on the left is replaced by a listing of Gig Performer widgets, including virtual knobs and sliders of different shapes and colors, along with text labels, meters, buttons, and toggle switches.



Gig Performer User Guide v 1.5.2

Widgets can be placed anywhere in a panel by simply dragging them to the desired location. To start, scroll down the list of widgets and drag a toggle switch from the left column somewhere into the panel. It will appear highlighted within a white rectangle and you can resize it by dragging any of the corner points. (The smaller section directly beneath the white rectangle displays a caption for the widget.)

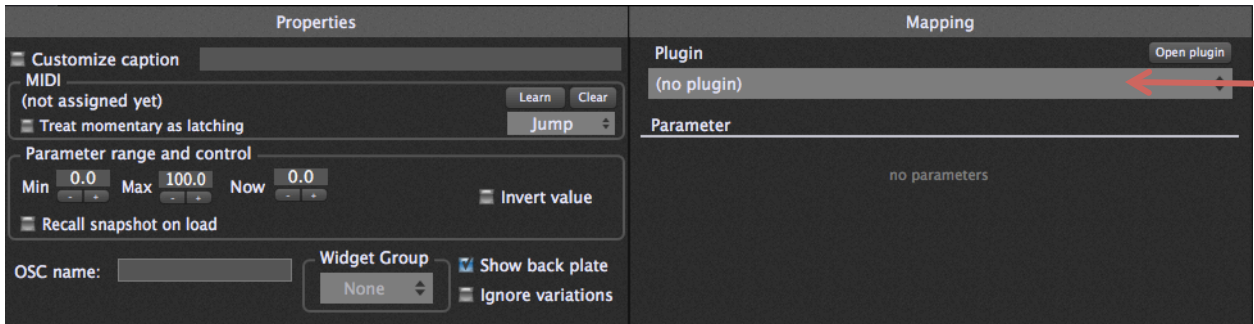


When a widget is highlighted, a Widget Properties “inspector” area appears below, as indicated by the red arrow. (The “New Panel” button above it allows you to create additional panels within the rackspace if desired.) The inspector is divided into two sections: Properties and Mapping. The Properties section on the left allows you to associate the widget with a physical button, knob or slider on a connected MIDI controller; it also allows you to optionally assign it an [OSC name](#). The Mapping section on the right allows you to assign the widget to control a particular plugin parameter. Together, this enables you to use the buttons, pads, sliders and knobs on your controller to change parameters in real time.

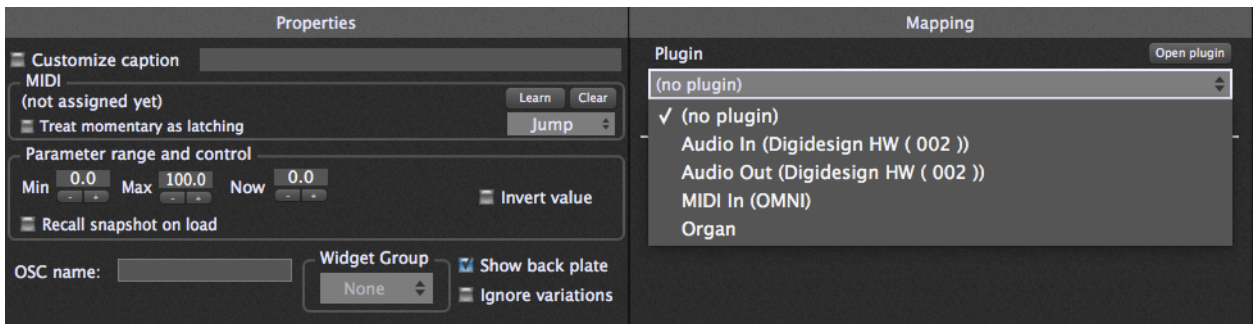
Note: There are no technical differences between the various knob and slider widgets—only cosmetic differences. They all do the same thing: they generate values when you move them. But their different appearances allow you to easily group related items or have certain kinds of parameters (for example, all volume parameters) stand out clearly.

Gig Performer User Guide v 1.5.2

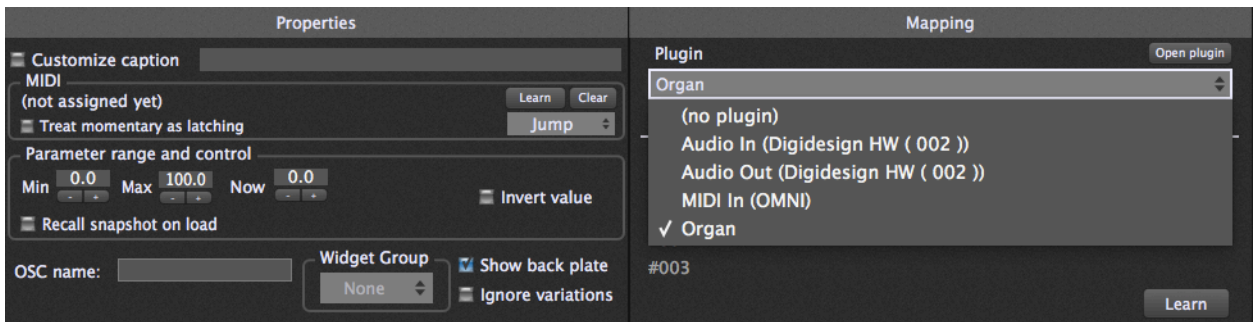
8. **Assign the widget to a plugin.** We're going to use this toggle switch to select fast or slow Leslie speeds for our Hammond organ sound. (Pretty much every Hammond organ plugin offers such a parameter). Start by telling Gig Performer which plugin you want to apply the widget to. This is accomplished by clicking on the plugin list dropdown menu beneath the word "Plugin" in the mapping section of the inspector (indicated by the red arrow).



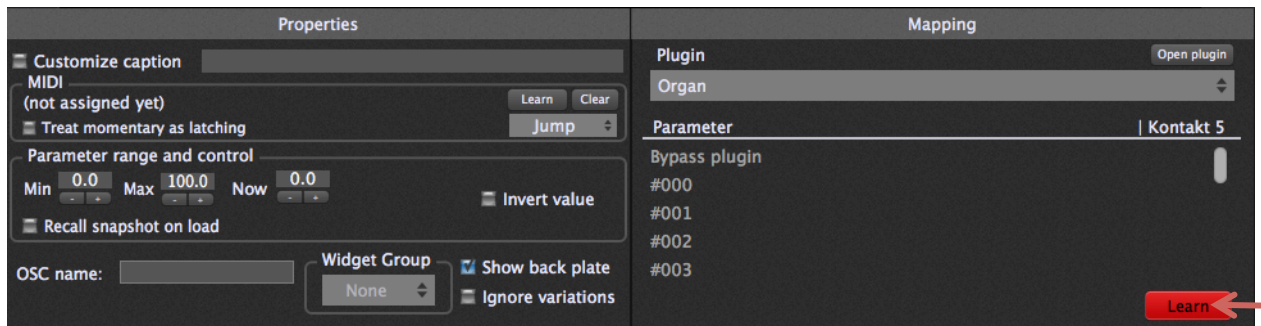
The default selection when you first create a widget is always "(no plugin)," as indicated by the checkmark to the left.



Scroll down the list of plugins—currently, there are just four of them: your audio interface input and output, the built-in MIDI In (Omni) plugin, and your Organ plugin—and select Organ.



9. **Assign the widget to a plugin parameter.** Next, we have to tell Gig Performer which parameter we want to have the widget control. This is done in the Parameter area of the mapping section of the Widget Properties inspector, immediately below the plugin list menu. As soon as you select a plugin, this area is populated with a list of available parameters that can be controlled via MIDI. This always begins with the “Bypass plugin” option, enabling you to use a widget to simply bring the selected plugin in or out of the signal chain. The remainder of the controllable parameters are plug-in specific. Many plugins provide a list that is easily readable (i.e., “volume,” “overdrive,” “reverb level,” etc.), in which case all you need to do is to select the parameter you want the widget to affect, and you’re done with this part of the process. However, other plugins (like the Kontakt 5 Player) simply offer a list of parameter numbers, leaving you to decipher which number affects what aspect of the sound. Even if that’s the case, no problem, thanks to Gig Performer’s a handy Learn mode, which enables it to instantly recognize and learn the appropriate parameter. To use Learn mode, simply click on the “Learn” button in the right-hand Plugin section (as indicated by the arrow) so that it turns red.

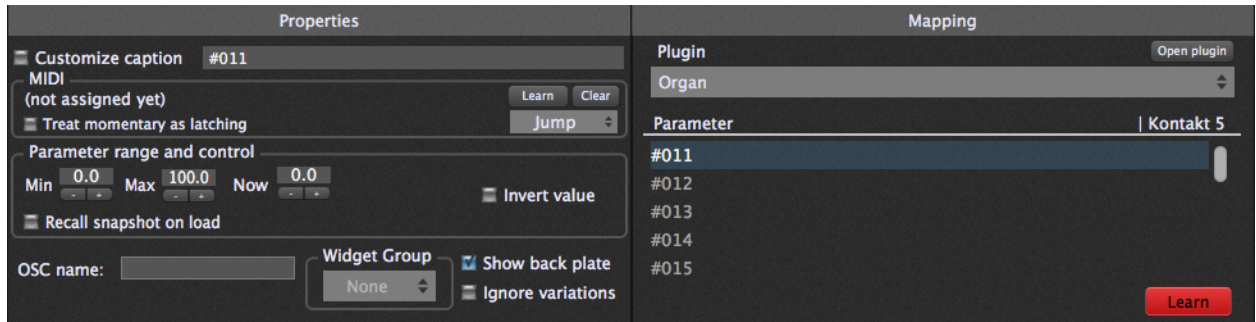


Doing so immediately opens the selected plugin’s editor window. Now all you have to do is use your mouse to move the control you want the widget to affect, and Gig Performer instantly learns the parameter. In the case of the “C3 Blues” instrument, the parameter is called “R Speed,” circled in yellow in this illustration.



Gig Performer User Guide v 1.5.2

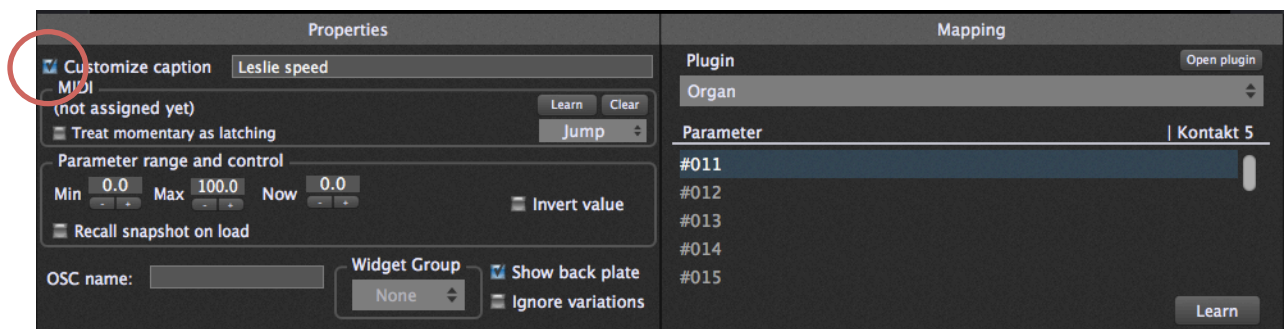
- 10. Use the widget to control the sound.** If you now close your plugin's editor screen, the mapping section of the Widget Properties inspector will display the parameter assigned to your widget (in this case, the cryptically named "#011").



At this point, if you exit Edit mode by clicking on the Widget Layout Designer icon in red (above your rackspace), you will see a panel with your toggle switch, in this case also cryptically named "#011. (We'll show you how to change the name in a moment.) More importantly, if you click on the toggle switch with your mouse or trackpad, it will flip between the "ON" and "OFF" positions, and your organ sound will switch between a slow Leslie and a fast Leslie speed. (Try it!) When you're done, disable Learn mode by clicking on the "Learn" button again so that it no longer lights red.

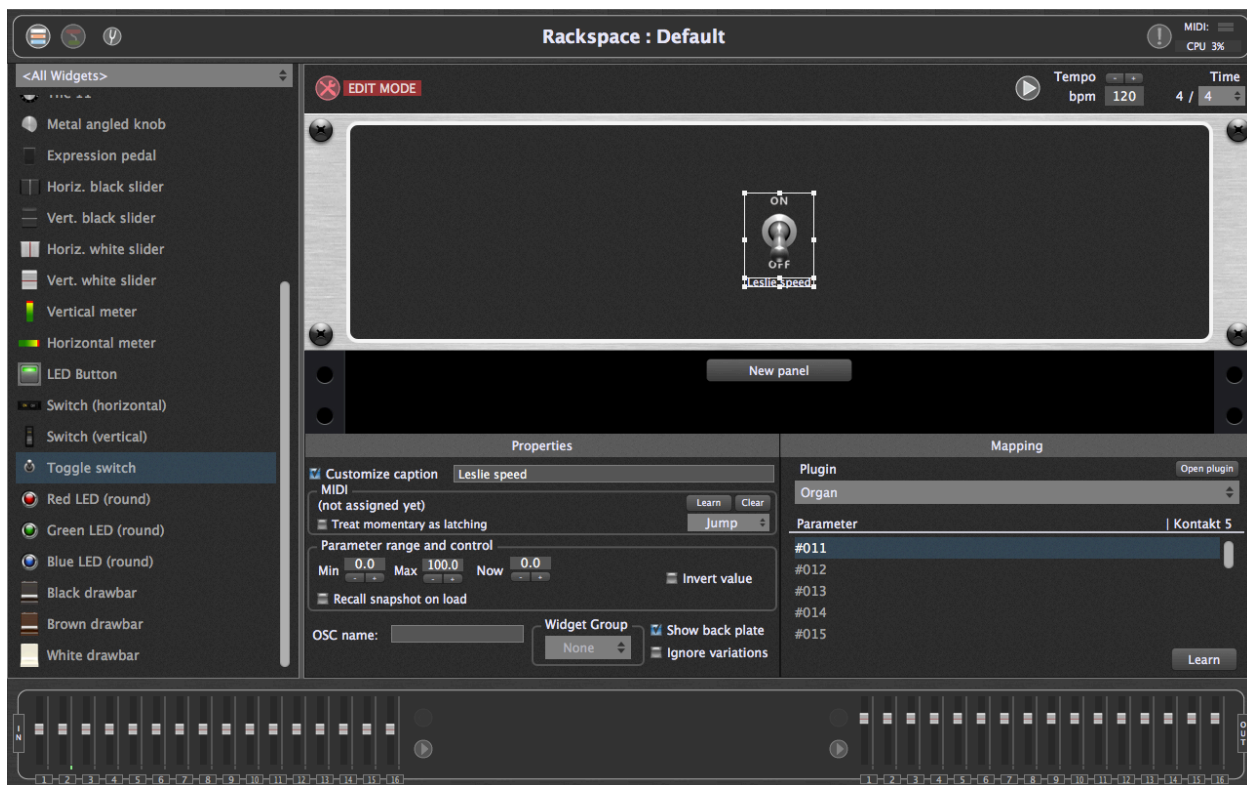
If for some reason a widget works the opposite way you'd like it to (for example, if the "ON" position of a toggle switch gives you a slow Leslie speed instead of the fast speed), you can easily change that in Edit mode. Just click on the widget to open the Inspector, then click on the "Invert value" button in the Properties section on the left; alternatively, you can set the MIN value higher than the MAX value. [Click here](#) for more information.

Edit mode also gives you the ability to rename the caption below a widget. To do so, simply click on the "Customize caption" button (circled in red) in the Properties section and type in a new name. Let's enter "Leslie speed" as the caption:



Gig Performer User Guide v 1.5.2

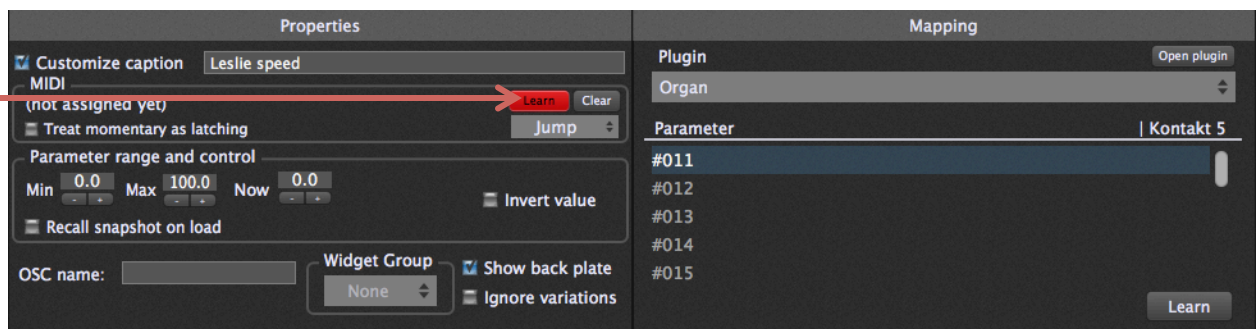
As you may notice, a name of more than a few characters may not display properly within the default caption space. No problem! As mentioned earlier, the white rectangle that appears around the selected widget (and caption below) in Edit mode allows you to resize it (and the accompanying caption) by dragging any of the corner points. Here we've made it considerably larger so as to accommodate the new "Leslie speed" caption:



Tip: Not sure how a widget movement is affecting your sound? It's easy to see for yourself. Simply double-click on the widget while in Edit mode and the associated plugin editor will open. When you exit Edit mode, the plugin editor remains open. Now all you have to do is move the widget and you can see the assigned parameter changing—a knob turning, a slider moving, or a switch/button going on or off—in real time.

11. Associate a widget with a physical device. The real power of Gig Performer is that it gives you the ability to associate widgets with physical devices like the knobs, sliders, pads, buttons, and switches on your MIDI controller, as well as any pedals connected to it. You can even control widgets by playing specific notes, or by pressing down on keys harder.

Once again, this process is made easy by a handy Learn mode, which enables Gig Performer to instantly recognize which physical device you want to use to control the selected widget. To enter this mode, simply click on the “Learn” button in the left-hand Properties section of the inspector (indicated by the arrow) so that it turns red.



Now all you have to do is touch or move any switch, button, knob, slider, or pad on your MIDI controller, play any note on your keyboard, or step on any connected pedal. Gig Performer instantly learns the MIDI message transmitted by that device, and the toggle switch widget moves at the same time. Try it!

If your keyboard controller has pads, they likely transmit MIDI note-on / note-off messages (unlike switches and buttons, which typically transmit MIDI controller messages). For that reason, when using notes or a pad to control an on-off widget such as the toggle switch, you may need to click on the “Treat momentary as latching” button in the MIDI section of the Widget Properties inspector for them to work correctly. [Click here](#) for more information.

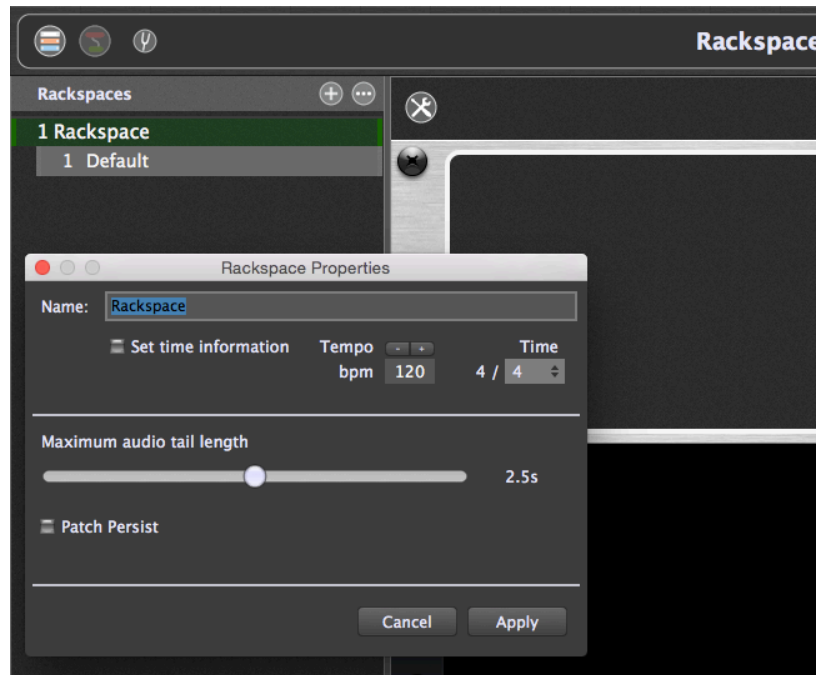
When you’re satisfied with the physical controller you’ve selected, click the “Learn” button again so that it no longer lights red.

Tip: While in “Learn” mode, you can click on different widgets and keep associating them with plugin parameters or physical controls—useful if you want to create all your widgets first and then associate them with parameters and physical controls later.

Important note: While you can use continuous physical controllers like knobs or sliders to control on-off widgets such as buttons or toggle switches (and vice versa: physical on-off controllers such as switches, buttons, or pads can be used to control continuous widgets such as knobs or sliders), this is not recommended practice!

Important note: Using keys to control widgets requires some extra care. When you first press a key while Gig Performer is in Learn mode, it will recognize and learn a Note On event. However, if you release the key while Learn is still enabled, Gig Performer will recognize and learn a Note Off message. In addition, if your keyboard controller generates aftertouch (channel pressure) messages, they will occur as you push the key harder in-between the Note On and Note Off messages. If Learn mode is still enabled while a note is being held, Gig Performer may well recognize and learn one of those values instead of the desired Note On or Note Off message.

- 12. Name your rackspace.** Double-clicking on the rackspace name (currently the default name “Rackspace”) in the upper left-hand corner of the screen brings up the Rackspace Properties dialog:



There are a number of options provided here, all of which are discussed in the [Reference section](#) of this User Guide. For now, simply type in a new name—let’s use “Organ with Leslie speed control”—and click on Apply, which will close the dialog.

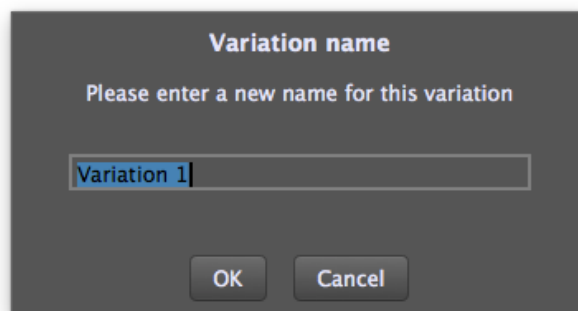
Gig Performer User Guide v 1.5.2

13. Create a variation. As discussed in the [Basic Terminology](#) section of this User Guide, a variation is essentially a rackspace preset which contains the same plugins and interconnections but with different parameter settings. Their usage allows you to create complex arrangements of plugins in a single rackspace without having to constantly duplicate rackspaces to craft different sounds—a great way to save on memory! For example, you could have multiple plugins followed by gain controls, with different plugins muted in different variations. Or you could connect multiple effects (i.e., phasing, flanging, tremelo, reverb, echo, etc.) and have each individually turned on/off or adjusted in different variations. This saves you having to create a separate rackspace for a plugin with phasing, and another for the same plugin with flanging, etc.

As an example, let's create one variation of our "Organ with Leslie speed control" rackspace that is set to a slow Leslie speed, and another that is set to a fast Leslie speed. You can freely switch between variations instantaneously—and with no interruption of sound—either manually, with up/down pedals, or via MIDI program change messages.

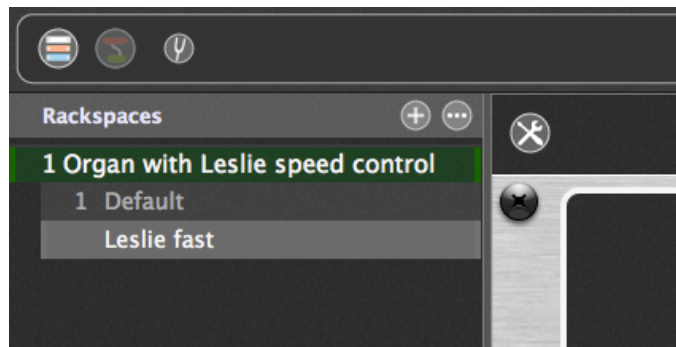
Variations contain only the value of widgets, as opposed to information about the plugins being used and their interconnections. However, since those widgets are connected to plugin parameters, whenever the widgets change, the parameter values will also change.

To create a new variation, either go to Edit | New variation or type CMD-N. A dialog will open asking you to type in a new name. Let's call this variation "Leslie Fast."

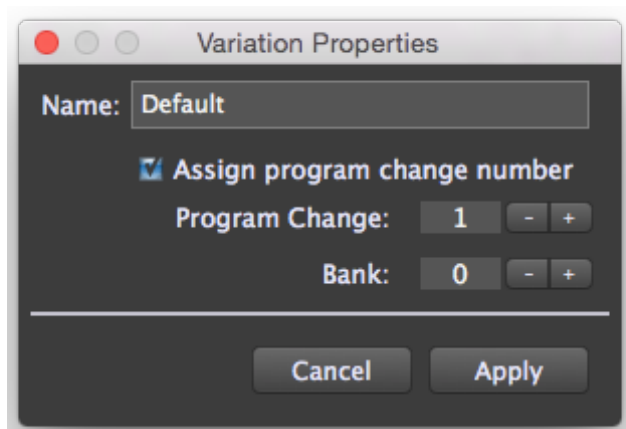


Gig Performer User Guide v 1.5.2

You’ve probably noticed that there’s already a variation for the “Organ with Leslie speed control” rackspace you created, with the name “Default”:



Every rackspace automatically comes with at least one variation, initially named “Default.” (The reason is that, technically, only variations—not the rackspaces themselves—can be selected with MIDI program change messages.) Now that we’ve created a second variation, let’s go ahead and rename the first one. To do so, double-click on the name, which will bring up the Variation Properties window:

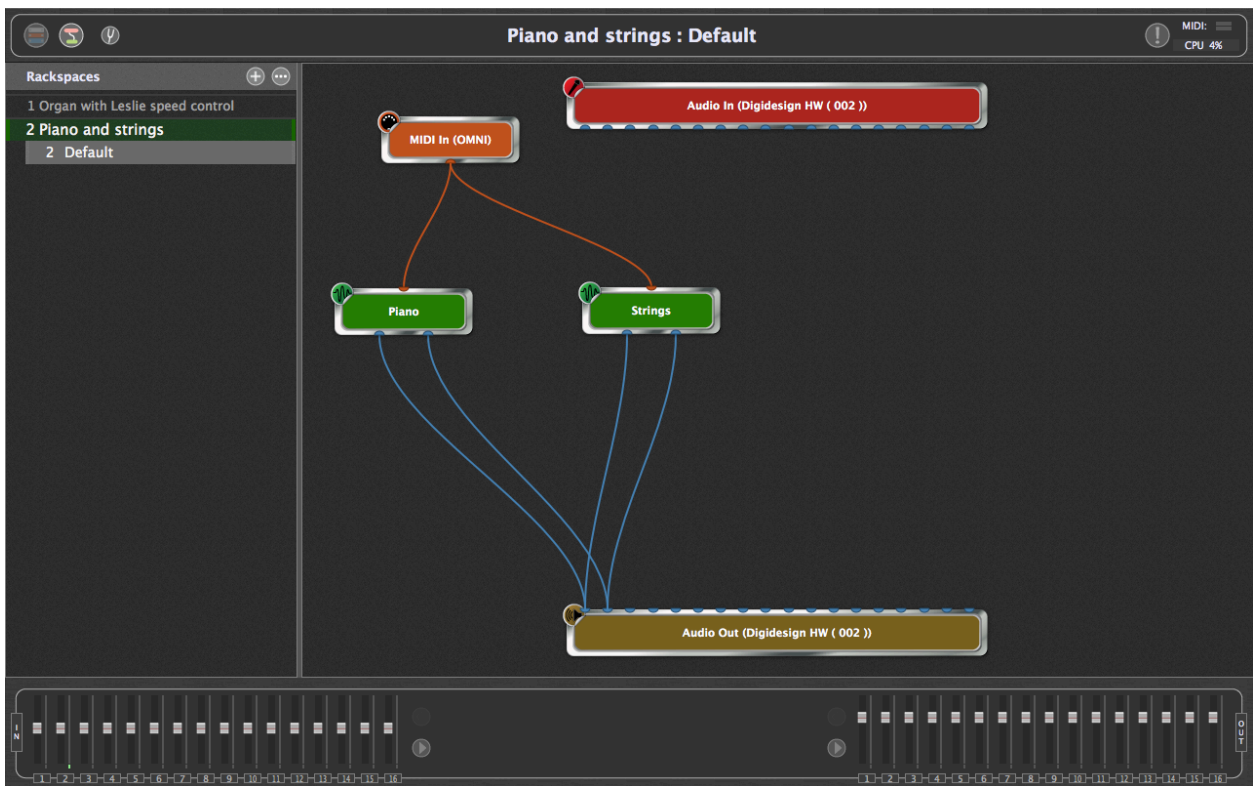


Enter the name “Leslie slow” in the Name field and click on Apply, which will close the dialog.

Now all that remains is to make sure the toggle switch widget is set to the “Off” position—something you can do either manually from your laptop or by moving the associated physical device on your MIDI controller (as set in step 11 above) and that it is set to the “On” position when you click on the “Leslie fast” variation. For more information about switching between variations with MIDI pedals or a MIDI controller, [click here](#).

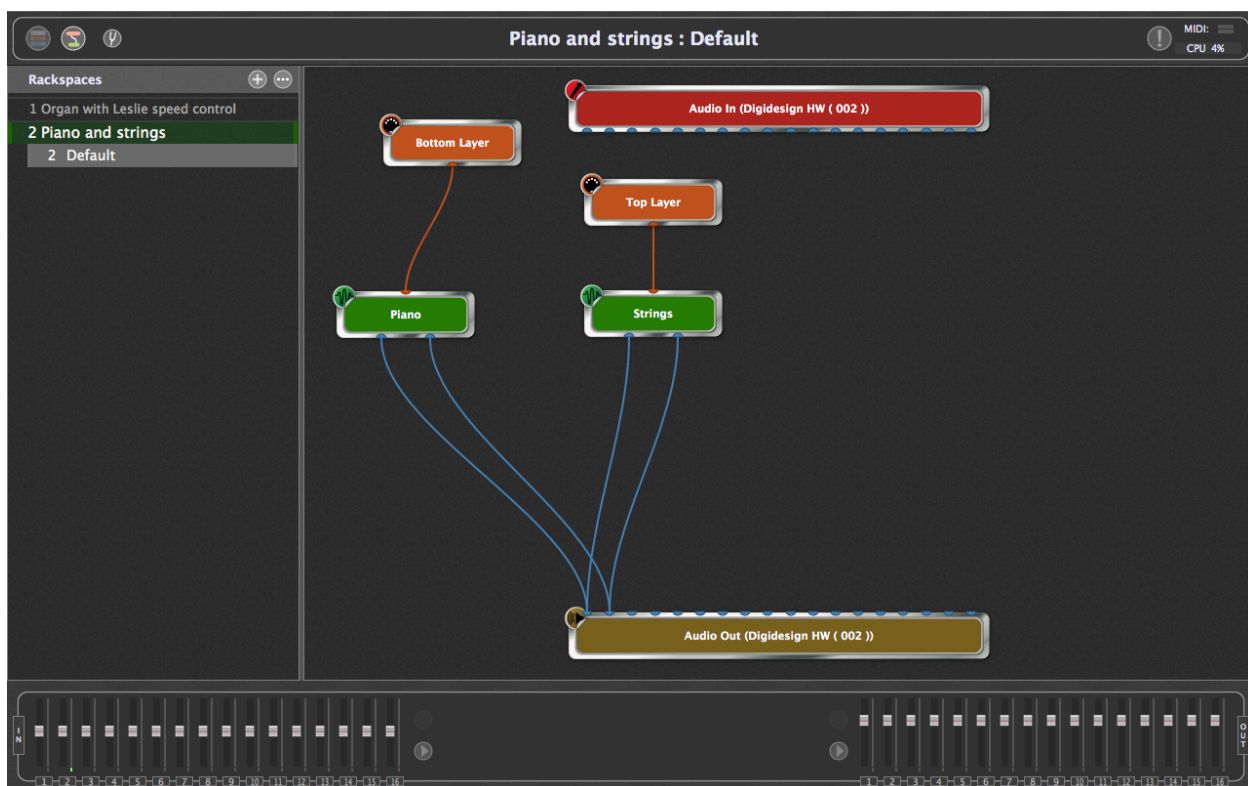
14. Create a new rackspace with multiple variations. As discussed in the [Basic Terminology](#) section of this User Guide, a rackspace can actually be pretty much anything you like: It can represent a single sound or all the elements needed for an entire song—or even just one part of a song. You can switch from one rackspace to another instantaneously, either manually, with up/down pedals, or via MIDI program change messages.

With that in mind, let's create a new rackspace by going to Edit | New rackspace, by using the SHIFT-CMD-N / CTRL-SHIFT-N keystroke commands, or by clicking on the "+" icon above the list of rackspace names. In the Rackspace Properties window that opens, rename it "Piano and strings," then, following the procedures outlined earlier in this tutorial, insert two plugins: one generating a piano sound and the other generating a string sound. Rename them "Piano" and "Strings" respectively, and make the appropriate MIDI and audio connections. Your Connections View screen should now look something like this:



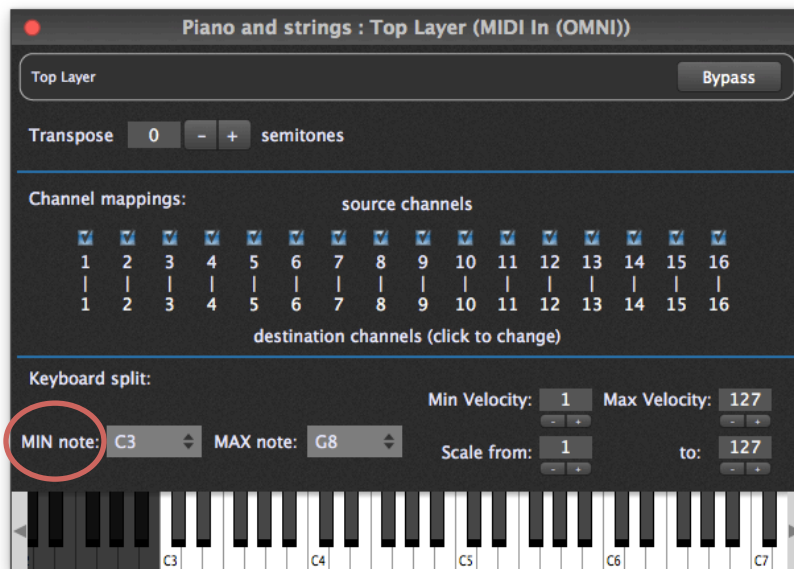
Gig Performer User Guide v 1.5.2

Gig Performer allows a single MIDI input block to play two or more sounds simultaneously, so if you now play your keyboard controller, you should hear both sounds, layered together. However, we're going to create a keyboard split so that the strings only layer with the piano when notes above Middle C are played. Doing this requires separate MIDI input blocks, so right-click in the Connections View screen and add a second "MIDI In (OMNI)" block and rename it "Top Layer," then move the orange connection from the first "MIDI In (OMNI)" block to the "Top Layer" block. Finally, rename the first "MIDI In (OMNI)" block as "Bottom Layer." Your screen should now look something like this:

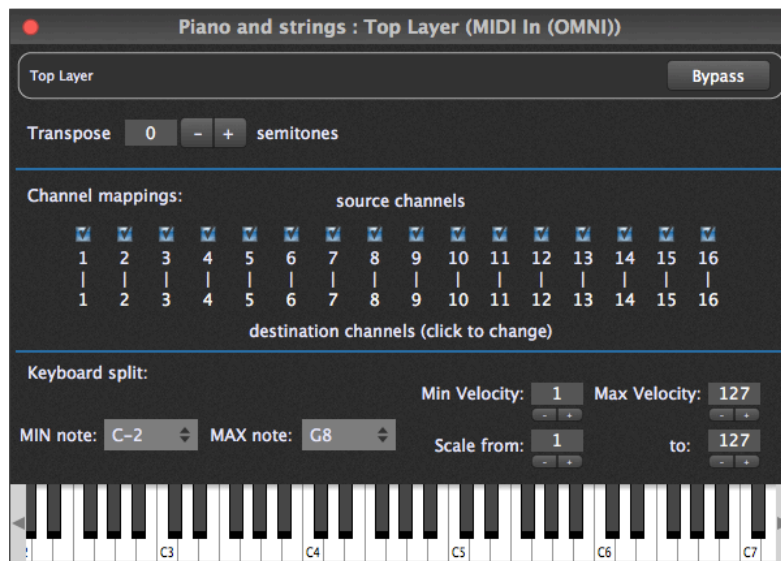


Gig Performer User Guide v 1.5.2

Next, we'll create the keyboard split. Double-click on the "Top Layer" MIDI Input block to open its plugin editor:



Click on the "MIN note" parameter (circled in red) and in the ensuing drop-down menu, scroll to "C3" and select it; the notes below C3 on the virtual keyboard become darkened, indicating that they are inactive. (Notes outside the specified MIN and MAX note range are always darkened this way.)



Your keyboard split is complete! If you now play your keyboard controller, you'll only hear strings when notes above Middle C are played. (If you'd prefer hearing the strings on their own when notes above Middle C are played instead of layering them with the piano, simply set the Max note for the "Bottom Layer" MIDI In block to B2.)

Gig Performer User Guide v 1.5.2

Now let's add widgets that allow the two sounds to be turned on and off individually. Switch to Performer View and enter Edit mode, then drag two LED Button widgets into the rack panel. Assign one to the "Piano" plugin and set it to the "Bypass plugin" parameter, then assign the other to the "Strings" plugin and set it to the "Bypass plugin" parameter as well. Click "Invert value" (in the left-hand Properties section) so that the "On" (button lit) position of the LED allows the associated plugin to be heard and the "Off" (button unlit) position causes the associated plugin to be bypassed. Caption each widget accordingly and test to make sure everything is working as expected. (Feel free to assign either or both widgets to physical devices on your MIDI controller if you like.) Your rackspace should now look like this:

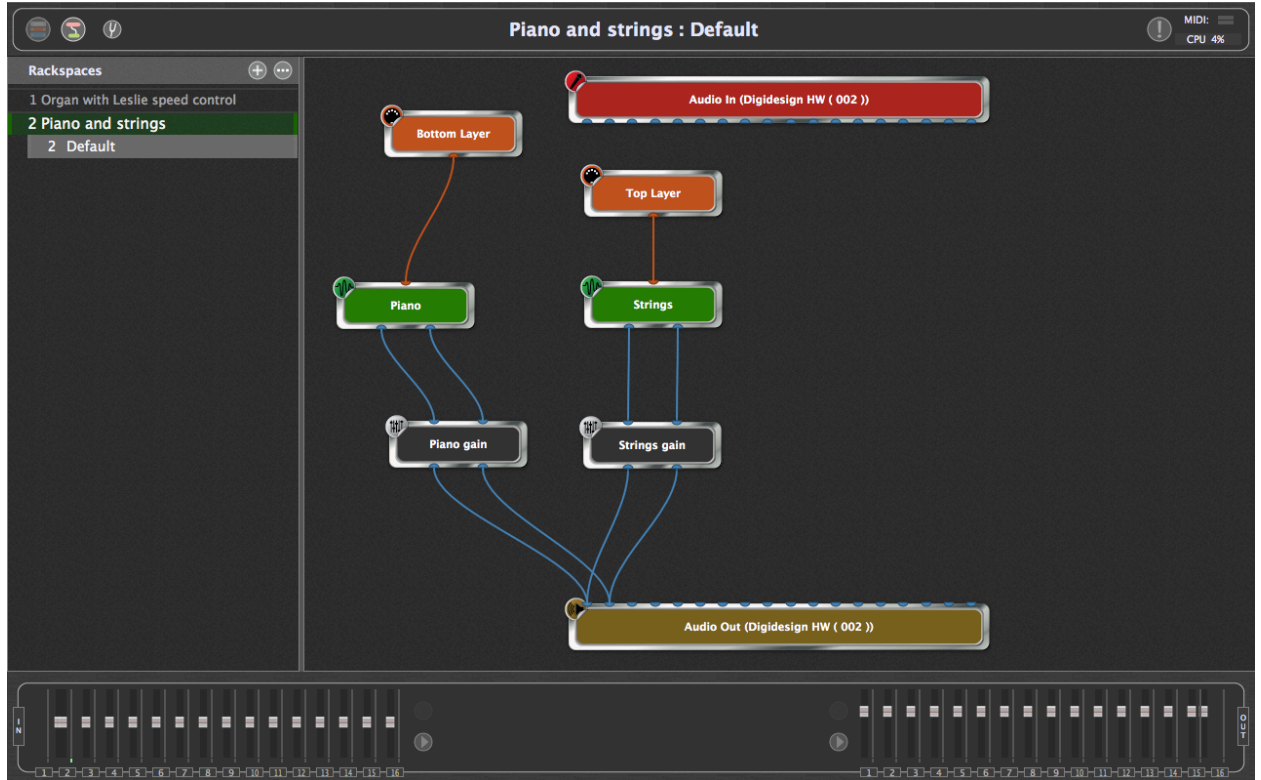


Next, we'll add a new panel and slider widgets to control the levels of the two sounds. (Rackspace can contain multiple panels, and Gig Performer provides options to resize them if your laptop provides limited screen real estate.) Although some plugins offer master volume parameters, many do not, and so the best way to accomplish this is with the use of the Gain and Balance plugin, supplied by Gig Performer. Accordingly, return to Connections View and insert two Gain and Balance plugin blocks. Connect one between the audio outputs of your "Piano" plugin and your audio interface's output block, and rename it "Piano gain." Then connect the other between the audio outputs of your "Strings" plugin and your audio interface's output block, and rename it "Strings gain."

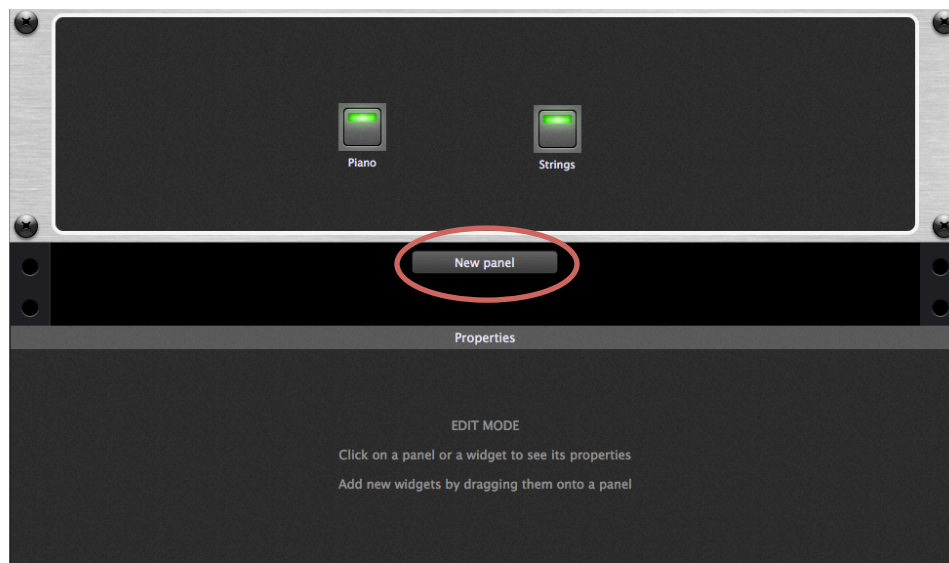
If you connect the audio outputs of multiple plugins to the input of another plugin, the signals will be mixed automatically. If you'd like to take control of that mix and create custom blends, simply insert one of Gig Performer's supplied Gain plugins ([Gain Control](#), [Gain and Balance](#), or [Gain and Balance \(m->s\)](#)) in-between, then adjust their volume controls as desired. These plugins can also be used as sends to effects such as reverb plugins. See the Audio Tutorial in this User Guide for [an example of how to do this](#).

Gig Performer User Guide v 1.5.2

Your screen should now look like this:

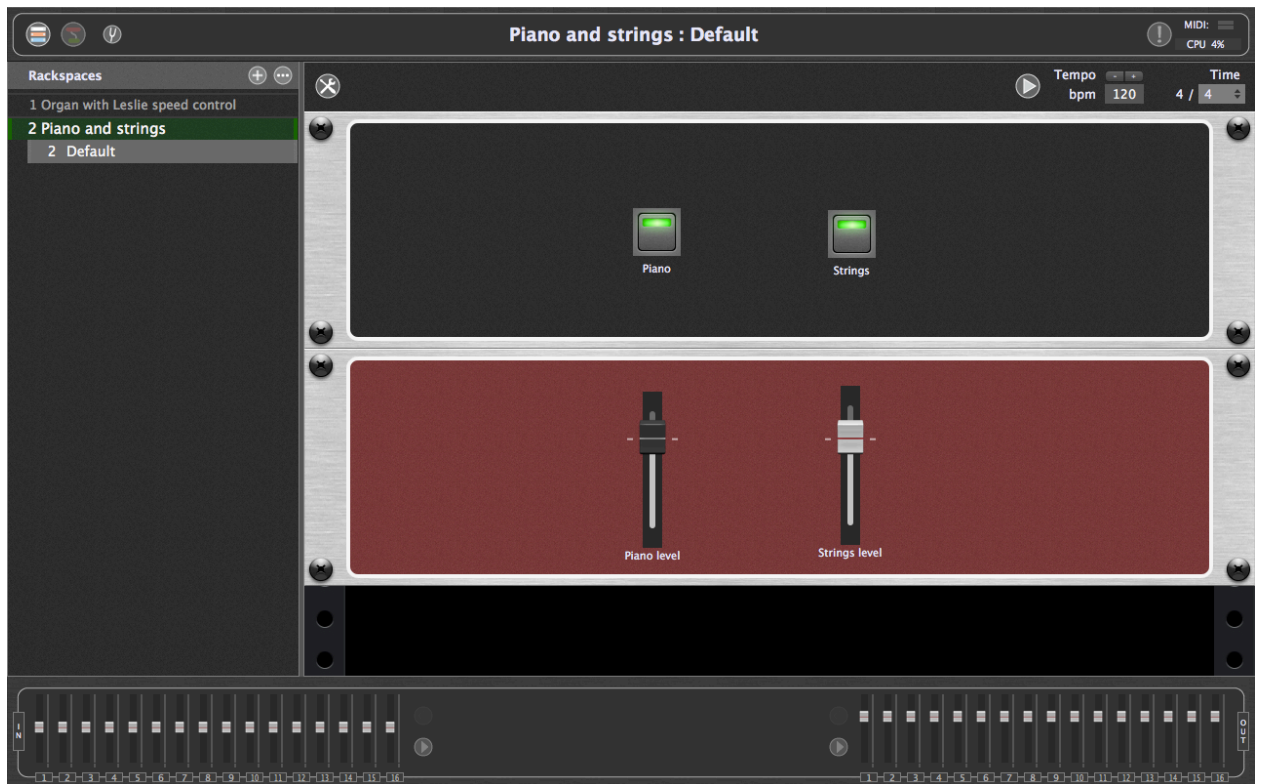


Return to Edit mode and click on the “New panel” button (circled in red) that appears between the first panel and the Properties area below. (Depending upon the screen size of your laptop, you may need to resize your screen or scroll down below the current panel to see this button.)



Gig Performer User Guide v 1.5.2

Select “New 2U panel” (“U” stands for “Rack Units”; you can always change the size later if necessary) and then double-click on the panel. From the “Style” menu that appears under “Rack Unit Properties,” select a custom color for the new panel. Next, drag one vertical black slider and one vertical white slider into the new panel. Assign the black slider to the “Gain” parameter of the “Piano gain” plugin (NOT the “Piano” plugin) and the white slider to the “Gain” parameter of the “Strings gain” plugin (NOT the “Strings” plugin). Finally, assign the black slider to control the “Piano gain” plugin’s “Gain” parameter, and the white slider to control the “Strings gain” plugin’s “Gain” parameter. Feel free to assign either or both widgets to physical devices on your MIDI controller if you like. (If you have an expression pedal connected to your controller, that would be an excellent way to fade the strings in and out—you can even substitute an Expression Pedal widget for the white slider.) Finally, caption the two sliders “Piano level” and “Strings level,” respectively. (You may need to resize them to view the entire caption.) Listen and test to make sure everything is working as expected. Your Performer View screen should now look like this:



Want to crossfade between two different sounds (such as the Piano and Strings in this tutorial), or two different effects? It's easily accomplished with [Widget Groups](#). See the Audio Tutorial in this User Guide for [an example of how to do this](#).

Gig Performer User Guide v 1.5.2

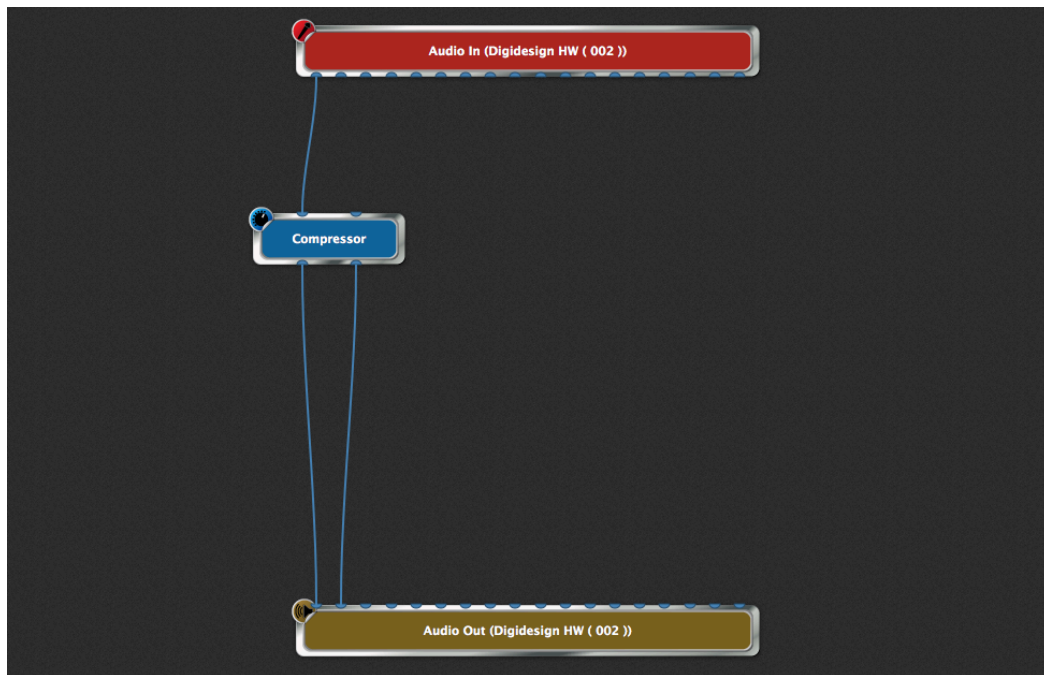
15. Finally, add a number of variations to the rackspace, following the procedures described in [step #13](#) of this tutorial. For example, you might want to have one variation that provides piano sound only (with the “Strings” LED widget unlit) and another that provides both sounds (both LED widgets lit) at full volume, with a third variation that provides strings at half volume. Go on—experiment! The sheer flexibility of Gig Performer means that the sky is the limit.
16. **Save your work.** Last but by no means least, be sure to save your work by going to File | Save As... and typing in a filename. The saved file, with the extension .gig, contains complete information about all rackspaces and variations.

You can also opt to save just the current rackspace for later importation into another Gig Performer session by selecting “Export current rackspace...” in the File menu. [Click here](#) for more information.

AUDIO TUTORIAL

This tutorial describes how to add effect plugins to vocals and/or guitar.

- 1. Launch Gig Performer.** If you have previously created a .gig file, it will automatically load (unless you have unchecked “Load last GIG file on startup” in [Options | General Preferences](#)). If so, go to [File | New Gig](#) to start with a clean slate.
- 2. Create a signal chain for voice.** We’ll start by constructing a typical signal chain for live vocals. Since we won’t be using a MIDI controller for this tutorial, start by right-clicking on the MIDI In (Omni) block, then selecting “Delete” from the popup menu that appears. Then turn off all speakers and connect a microphone to an input of your audio interface, then enter Connections View and, using the procedures described in the [previous tutorial](#), add the compressor plugin of your choice (because it is an effects processor, its block will be colored blue, as opposed to sound generating plugins, which are colored orange), rename it “Compressor,” and make audio connections to the red and brown interface blocks. Slowly turn up the volume of your sound system (to avoid feedback), then double-click on the plugin to open its editor and adjust its parameters until you are satisfied with the degree of compression. Close the editor to return to Connections View. The main area of the screen should look something like this:



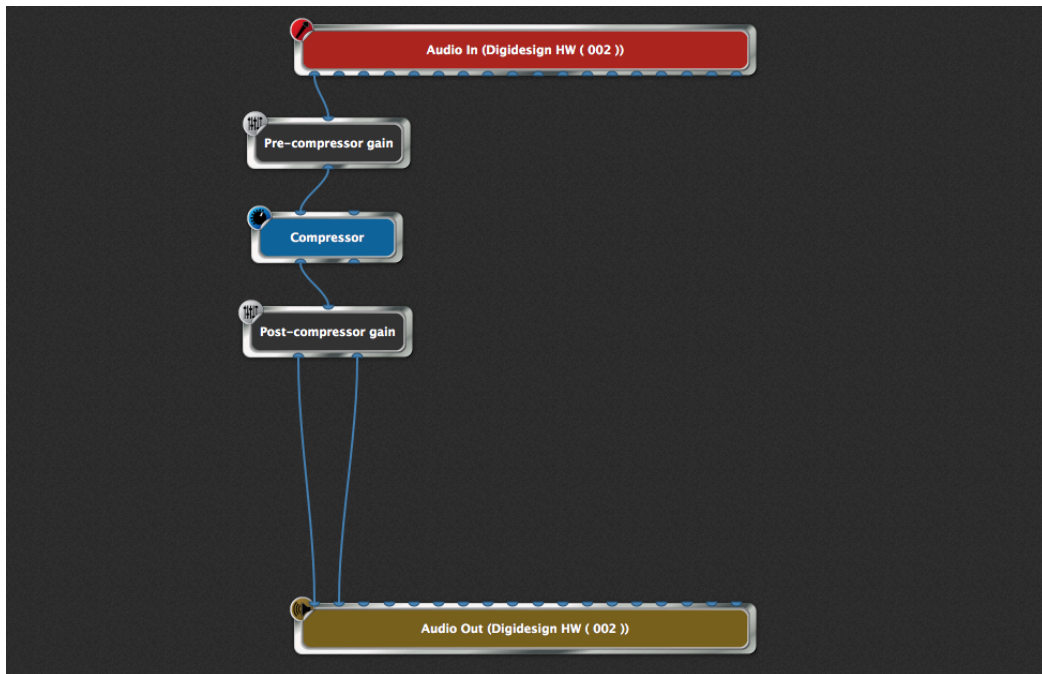
(Your screen may look slightly different, since the audio input and output blocks will reflect the name of your connected audio interface as well as the number of inputs and outputs it provides.)

Gig Performer User Guide v 1.5.2

Note: If you have multiple audio interfaces connected to your system and you want to use an interface other than the one being displayed, right-click on the Audio In and/or Audio Out blocks and select “Replace plugin.” You can even specify different interfaces for input and output.

Tip: If you’re running Gig Performer on a Mac OSX system, the Audio MIDI Setup utility allows you to create Aggregate Devices, each consisting of multiple audio interfaces. [Click here](#) for more information.

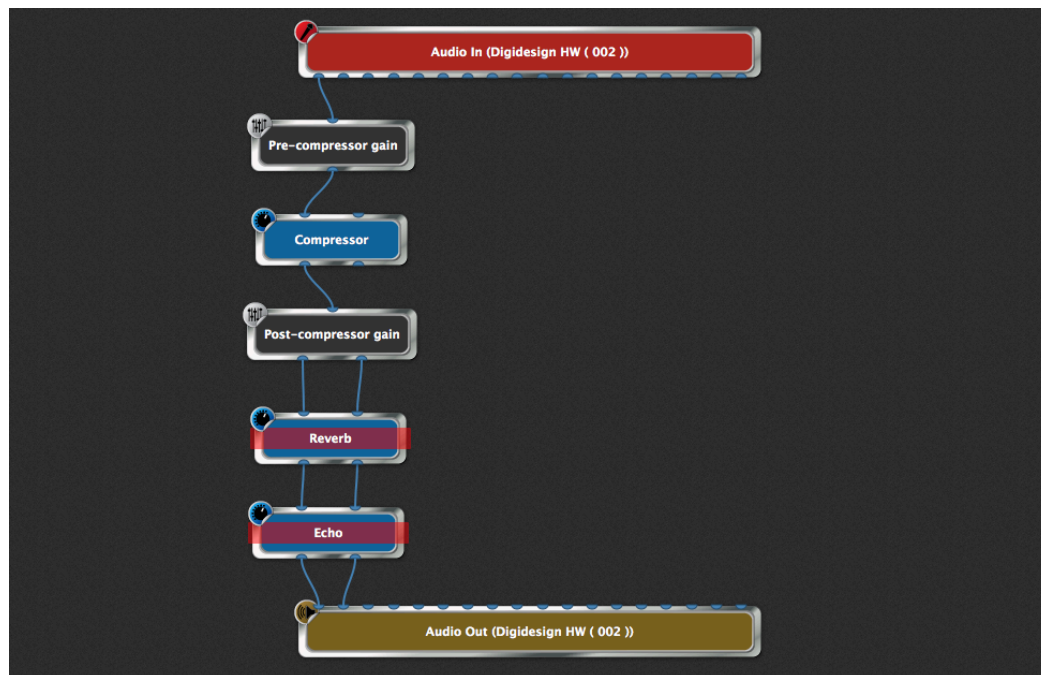
Next, for metering purposes, we’ll add two supplied Gig Performer plugins. Insert a “Gain Control (mono)” plugin before the compressor and rename it “Pre-compressor gain,” then insert a “Gain and Balance Control (mono to stereo)” plugin after the compressor and rename it “Post-compressor gain.” (The latter has the benefit of converting a mono source such as a microphone to stereo, which will allow us to add stereo reverb and echo effects.) The main area of the Connections View screen should now look like this:



Tip: There are two ways to modify connections: You can remove an existing connection by right-clicking on it and selecting “Remove this connection” (or by dragging it to any empty space) and then create a new connection. Alternatively, you can click near one end of a connection and drag it to a new pin.

Gig Performer User Guide v 1.5.2

Let's complete the signal chain with the reverb and echo / delay plugins of your choice. **Important: To avoid possible signal overloading, we recommend you take the precaution of placing reverb and echo plugins in bypass before making their audio connections.** This is easily accomplished by right-clicking on them and selecting Bypass from the pop-up menu. (Gig Performer also adds a Bypass button to the upper right-hand corner of every plugin editor.) Once that is done, rename the two plugins "Reverb" and "Echo," respectively. The main area of the Connections View screen should now look like this:

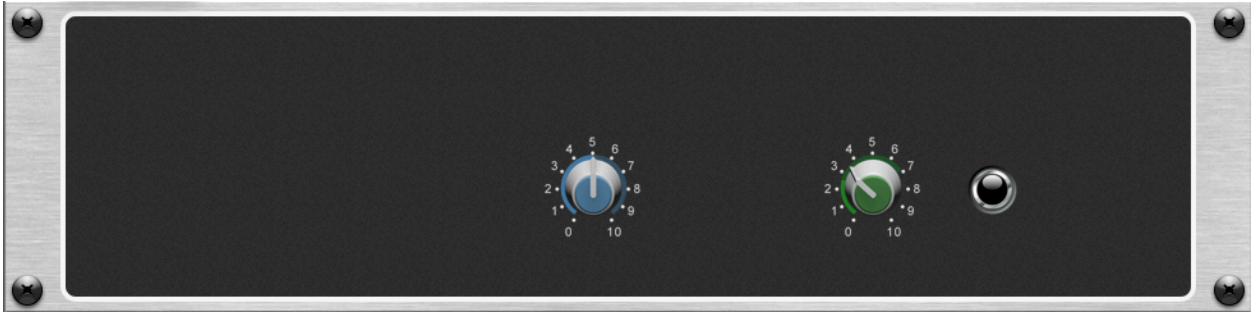


Note: Plugin blocks that are bypassed are overlaid with a red see-through rectangle, as shown in the illustration above.

Important note: There is a difference between bypassing a plugin and muting it. Bypassing actually turns off processing for that plugin, thus stopping the sound from being generated or processed. Muting, which is accomplished by inserting a [Gain Control](#), [Gain and Balance](#), or [Gain and Balance \(m->s\)](#) plugin and setting its volume to 0, causes audio signal to continue to be generated, but with the output silenced. If you're playing back a loop or click track during 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.

Gig Performer User Guide v 1.5.2

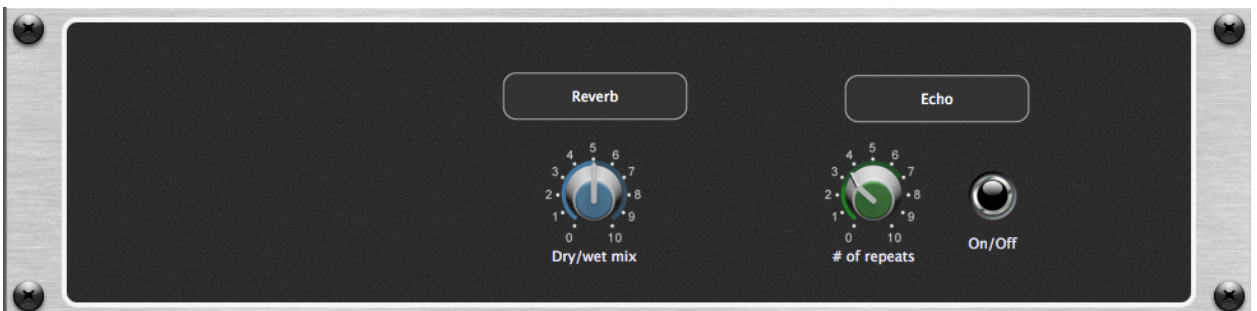
3. **Add widgets to control the vocal sound.** It can be very helpful to be able to adjust the amount of reverb added to singer's voices during live performance, and you almost certainly will want to be able to bring echo in and out as needed. Accordingly, go to Performer View and click on the Widget Layout Designer icon to enter Edit mode. Drag a blue knob, a green knob, and a red LED into the panel and arrange them roughly this way:



Add captions and assignments to these widgets as follows:

- Blue knob
 - Caption: "Dry/Wet mix"
 - Assignment: Dry/Wet parameter of your reverb plugin.
- Green knob
 - Caption: "# of repeats"
 - Assignment: Feedback parameter of your echo plugin.
(This determines the number of repeats you hear.)
- Red LED
 - Caption: "On/Off"
 - Assignment: "Bypass plugin" parameter for your echo plugin.
(Click on "Invert value" so that the plugin is bypassed when the LED is unlit and active when it is lit.)

Finally, for purely cosmetic reasons, drag two Text label widgets into the panel. Position one over the blue knob and caption it "Reverb," then position the other over the green knob and red LED and caption it "Echo." Exit Edit mode and return to Performer View. Your panel should now look like this:



Gig Performer User Guide v 1.5.2

4. Add metering. Metering can be very helpful to verify proper gain structure and ensure that a signal is not being overloaded and distorted. The meter widgets provided by Gig Performer (available in both vertical and horizontal configurations) give you the ability to visually monitor levels at any point in the signal chain. To add metering, simply insert Gain Control (or Gain and Balance Control) plugins wherever you want to be able to view signal (as we did in [step # 2 above](#)) and assign meter widget(s) to their “Magnitude” parameter. Accordingly, drag two horizontal meter widgets into the left-hand side of the panel and add captions and assignments as follows:

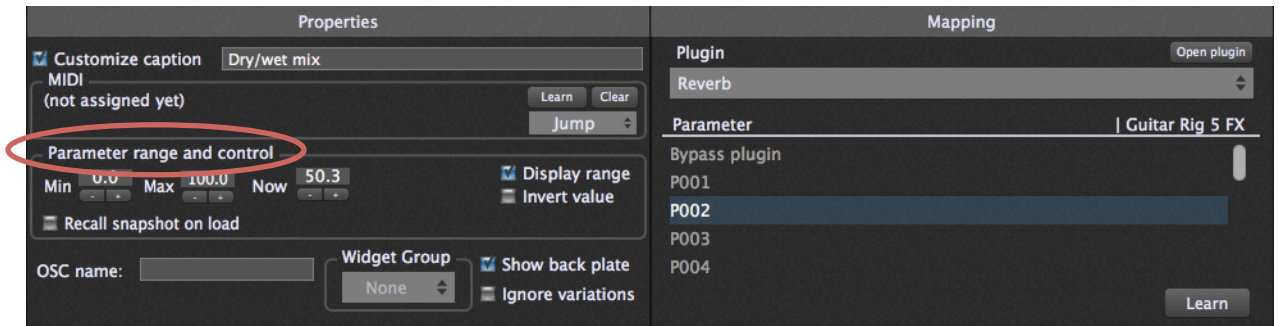
- Top horizontal level meter
 - Caption: “Pre-compressor”
 - Assignment: “Magnitude” parameter for the Pre-compressor gain plugin.
- Bottom horizontal level meter
 - Caption: “Post-compressor”
 - Assignment: “Magnitude” parameter for the Post-compressor gain plugin.

Your panel should now look like this:



Gig Performer User Guide v 1.5.2

- Add scaling.** Most singers like to have at least some reverb on their voice at all times, but rarely will they want their sound to be swimming in reverb. This is where Gig Performer’s scaling control comes in handy. It allows you to specify a minimum and maximum range for each widget, while still allowing full range of motion. To demonstrate, take your Reverb plugin out of bypass, then go back into Edit mode and click on the blue knob to select it. In the Properties section of the inspector, you’ll see an area labeled “Parameter range and control” (circled in red below):



Enter a Min (minimum) value of 30 and Max (maximum) value of 60. (The numbers can be typed in directly, or you can use the + and – buttons below to increment or decrement the value.) Now when the blue knob widget is at its fully counterclockwise “0” position, you’ll hear a dry/wet mix of 30% (which, for most plugins, will result in a small amount of reverb), while at its fully clockwise “10” position, you’ll hear a dry/wet mix of 60% (which, for most plugins, will result in a significant, but not overwhelming amount of reverb. (Feel free to adjust the Min and Max values to taste!)

Similarly, an infinite number of echo repeats can be a very unpleasant thing—the sound can quickly spiral out of control—so set the Max value for the green knob widget to 70. That way, when the green knob is at its fully counterclockwise “0” position, you’ll hear a single echo, while at its fully clockwise “10” position, you’ll hear several—but not too many—echoes. Again, feel free to adjust to taste.

Gig Performer User Guide v 1.5.2

- 6. Test, tweak, and save.** Test that all is working as expected by moving the blue and green knob widgets manually with mouse-dragging operations and turning the red LED on and off by clicking on it. (Take your echo plugin out of bypass before doing so.) If you have a MIDI controller or pedalboard connected to your system, you can associate each of these three widgets with physical devices such as knobs, sliders, buttons, and pads on your controller. Obviously the two knob widgets are best controlled from knobs and sliders, but the echo on/off widget (the red LED) can be controlled from buttons, pads, a pedal, or even a note being played! For example, you can assign it to the highest note on your keyboard—one which is probably rarely played in normal circumstances—and the echo will turn on only when that key is depressed. ([Click here](#) for an important note about using keys to control widgets.) Again, thanks to the flexibility of Gig Performer, the only limit here is your imagination. When you've got everything tweaked to your satisfaction, add any variations you want and name your rackspace—for the purposes of this tutorial, we'll call it "Vocal." Finally, be sure to save your work.
- 7. Create a new instance.** Next, we're going to be constructing a signal chain for guitar, but before we do so, we'll be placing it in a separate instance. Instances are simply multiple copies of Gig Performer operating simultaneously in the same computer. Their use allows each member of a band to have independent control over his or her own rackspaces without affecting the rackspaces of other bandmembers. For example, a guitarist can freely switch between rhythm and lead settings without affecting the reverb and echo settings being applied to vocals.

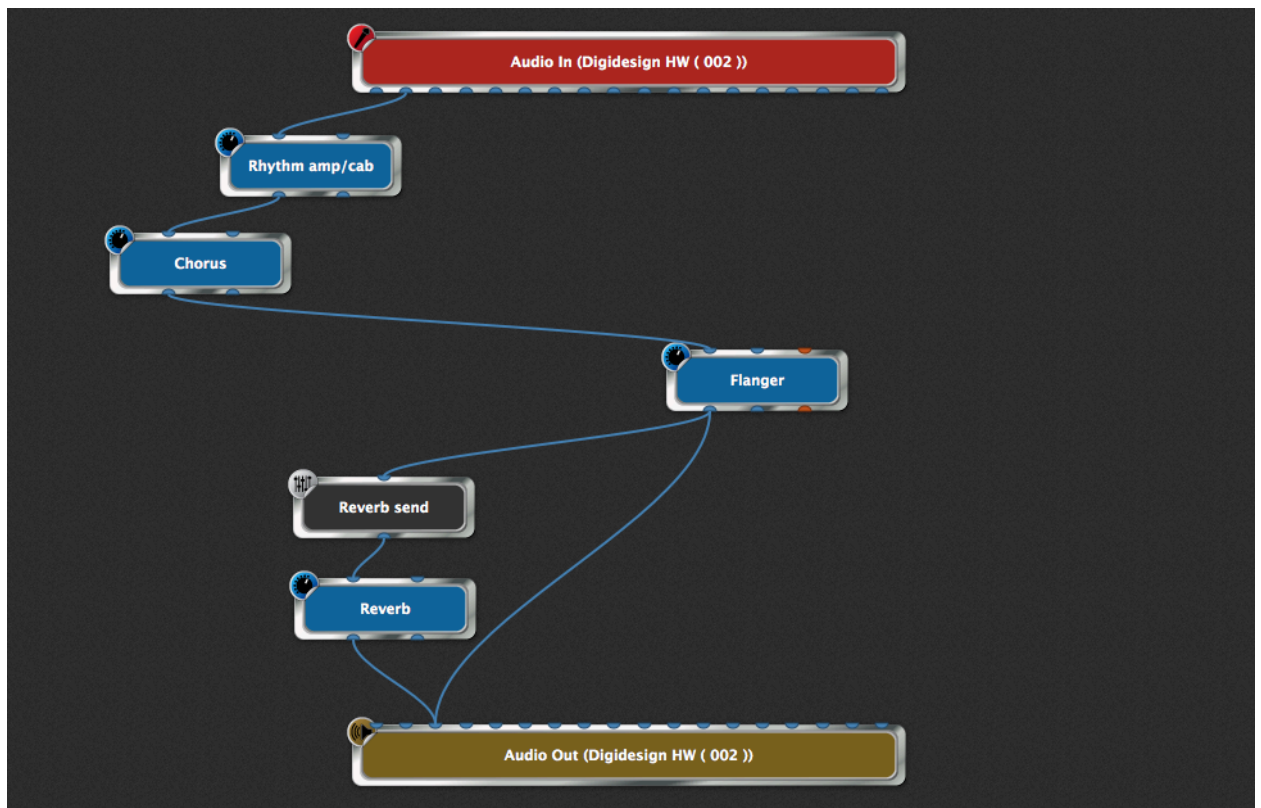
Go to Options | Advanced Settings... and make sure the "Enable multi-instance support" option is checked. (For more information, [click here](#).) Then go to File | Create New Instance and type in a name. For the purposes of this tutorial, let's call the new instance "Guitar." A dialog box will appear informing you that you have created a new instance.

Click OK to close the dialog and your new instance will appear—a complete new iteration of Gig Performer with no plugins, no connections, and no widgets.

- 8. Create signal chains for guitar.** Now it's time to build a typical signal chain for guitar—actually, two of them. Begin by once again deleting the MIDI In (OMNI) block and then connect a guitar (via a DI box if necessary) to an input of your audio interface other than the one used for the microphone earlier in this tutorial.

Gig Performer User Guide v 1.5.2

Then enter Connections View and connect that input to the amp/cabinet model plugin of your choice, followed by the chorusing plugin of your choice and the flanging plugin of your choice. Then add a [Gain Control](#) plugin and the reverb plugin of your choice. Rename the amp/cabinet plugin “Rhythm amp/cab,” the chorusing plugin “Chorus,” the flanging plugin “Flanger,” the Gain Control plugin “Reverb Send” (since it’s taking on that function; see the [text box below](#)) and the reverb plug-in “Reverb.” Connect the output of the Flanger plugin to the “Reverb send” plugin, and the output of the the “Reverb send” plugin to the Reverb plugin. Finally, connect an output from both the Flanger and Reverb plugins to one of your audio interface’s outputs (choose an output other than the one being used for the microphone signal), as shown in the illustration below:

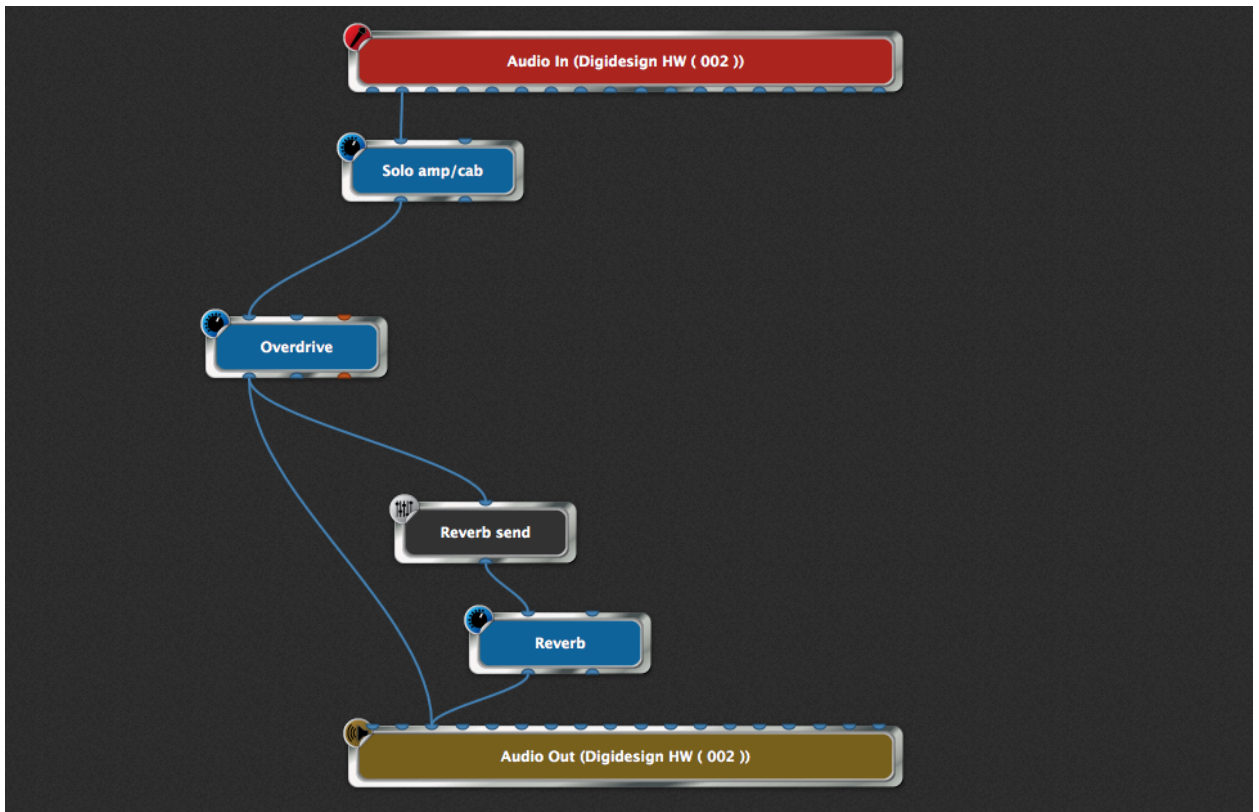


Tweak the parameters in the various plugin editors until you get a sound you like (use the Gain Control to set the desired amount of reverb), then double-click on the rackspace name. In the Rackspace Properties dialog that appears, rename the rackspace “Rhythm,” then click on Apply to close the dialog.

Gig Performer makes it easy to create **reverb sends**, where signal is routed to a reverb (or other type of signal processing plugin) from multiple sources: Simply insert a Gain plugin between the output of each source and the reverb plugin. The volume of each Gain plugin provides independent control over how much signal is sent to the reverb.

Gig Performer User Guide v 1.5.2

Next, we'll create a second rackspace for soloing. Since we're going to use the same audio connections (although we'll be substituting different plugins), go to Edit | Duplicate the current rackspace (or use the SHIFT-CMD-D / CTRL-SHIFT-D keystroke commands) to add a new rackspace, then double-click on its name. In the Rackspace Properties dialog that appears, change the name of this new rackspace from "Rhythm_1" to "Solo". In Connections View, substitute an amp/cabinet model if you prefer (you can also opt to simply use the same one you used for the Rhythm sound—Gig Performer allows you to use the same plugin multiple times, up to the limits of your RAM and CPU resources), followed by the distortion/overdrive plugin of your choice, then delete the flanging plugin. Rename the amp/cabinet plugin "Solo amp/cab" and rename the distortion/overdrive plugin "Overdrive." Your main Connections View screen should now look like this:



Guitarists especially will appreciate the handy fullscreen chromatic tuner provided by Gig Performer, available at any time with the click of a mouse or the press of a MIDI footswitch—even with the touch of a button on a smartphone or tablet running an OSC app. [Click here](#) for more information.

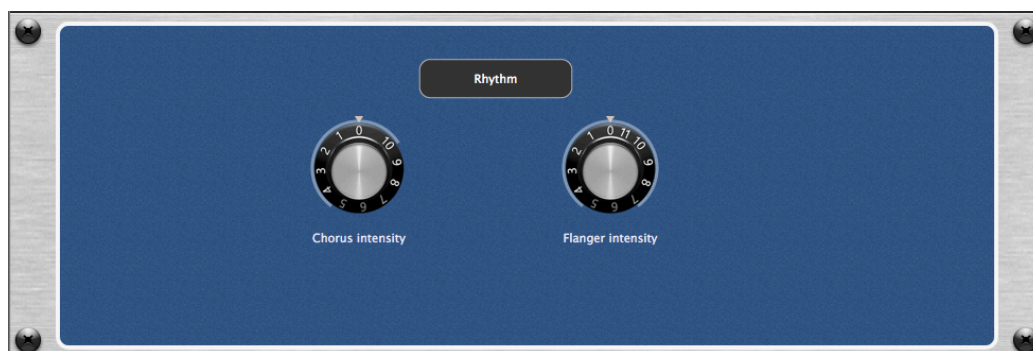
Gig Performer User Guide v 1.5.2

9. **Add widgets to control the guitar sounds.** Go to Performer View and select the Rhythm rackspace, then click on the Widget Layout Designer icon to enter Edit mode. Drag an “Old school” knob and a “The 11” knob (our Spinal Tap version of the “Old school” knob—it goes to 11!) into the panel.

Add captions and assignments to these as follows:

- “Old school” knob
 - Caption: “Chorus intensity”
 - Assignment: Effect level parameter of your Chorus plugin.
- “The 11” knob
 - Caption: “Flanger intensity”
 - Assignment: Dry/Wet parameter of your Flanger plugin.

Apply widget scaling and MIDI controller assignments to taste, then, purely for cosmetic reasons, add a text label widget and caption it “Rhythm,” then double-click on the panel and in the Rack Unit Properties menu that appears, change its “Style” (actually, its color) to blue. It should now look like this:



Now go to the Solo rackspace and drag a text label widget and an “11 knob” widget into the panel. Caption the “11” knob “Overdrive” and assign it to the Drive parameter of your Overdrive plugin. Apply widget scaling and MIDI controller assignments to taste, then caption the text label widget “Solo” and change the rack panel “Style” (color) to green. It should now look like this:

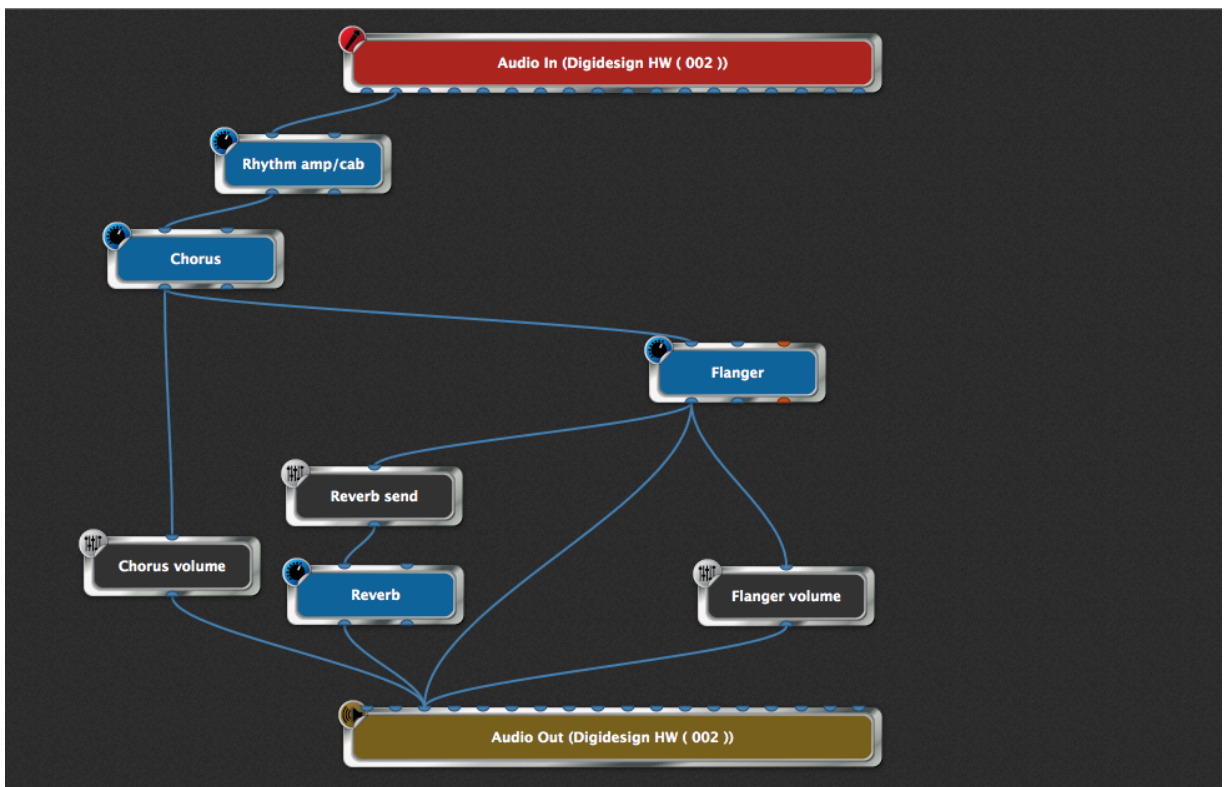


Gig Performer User Guide v 1.5.2

The advantage to putting different sounds into different rackspace (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 5 seconds—when you switch rackspace—something that’s particularly useful in situations like this, where reverb or echo is being applied to a sound. Try it!

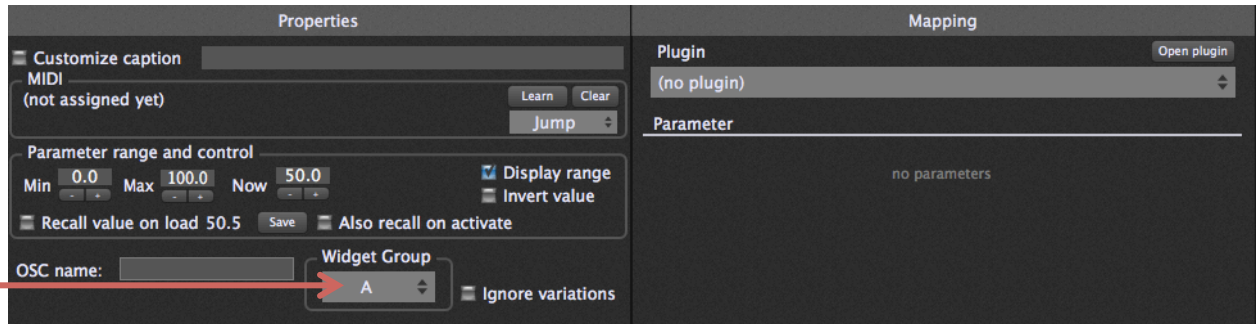
10. Create a widget group to crossfade between the chorus and flanging effects.

Now it’s time to explore Gig Performer’s powerful [widget group](#) feature. The concept behind it is simple: Any widget assigned to a group can be controlled by any or all other widgets assigned to the same group. To demonstrate, go to Connections View, select the Rhythm rackspace, and add two Gain Controls, renaming one of them “Chorus volume” and the other one “Flanger volume.” Connect the output of the “Chorus” plugin to the input of the “Chorus volume” plugin and the output of the “Flanger” plugin to the input of the “Flanger volume” plugin. Then connect the outputs of both the “Chorus volume” and “Flanger volume” plugins to the same output of your audio interface that you’ve been using for both the Rhythm and Solo rackspace. Your main Connections View screen should now look like this:



Gig Performer User Guide v 1.5.2

Next, go to Performer View and move the two knob widgets to set moderate amounts of both Chorus intensity and Flanger intensity. Then go into Edit mode and drag a horizontal slider widget into the same panel, positioning it below the two knob widgets. Next, in the Properties section of the inspector, click on the dropdown menu below Widget Group (indicated by the arrow) and select Group A. (Gig Performer provides 26 widget groups, lettered A through Z.)



Caption the widget “Chorus <-> Flanger” but don’t assign it to any plugin; that’s because we’re going to use it as the “master” to control two other widgets we’ll be placing in a separate panel. Accordingly, click on the New Panel button beneath the current panel (depending on the screen size of your laptop, you may need to scroll down or resize the window to see this button). From the dropdown menu that appears, click on “Create a new 1U panel”. This is the smallest size panel provided by Gig Performer, but it will work fine for our purposes since we won’t actually need to see its widgets during live performance. Drag a blue knob widget into the left side of the new panel and a green knob widget into the right side. Caption the blue knob “Chorus volume” and assign it to the Gain parameter for the “Chorus volume” plugin. Then caption the blue knob “Flanger volume” and assign it to the Gain parameter for the “Flanger volume” plugin. Next—**and this is important**—click on Invert Value (in the Properties section of the inspector) for the blue knob widget **only**.

Finally, assign both knob widgets to Widget Group A—the same group as the horizontal slider in the panel above. Because all three widgets are assigned to the same 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 “Invert 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. Try it! If you want visual evidence that all is working correctly, go momentarily to Connections View and double-click on the “Chorus volume” and “Flanger volume” plugin blocks to open their editors, then return to Performer View. The editors will remain open and you’ll be able to see the effect of moving the horizontal slider.

Gig Performer User Guide v 1.5.2

Your final Rhythm rackspace should look like this:



If you find the two knobs in the small bottom panel visually distracting, Edit mode provides a couple of easy ways to fix that:

a) Right-click on the top panel, then select the “Convert to 4U panel” option from the drop-down menu. This will enlarge its size, thus hiding the panel below from view.

or

b) Create one or more new panels and move them in-between your “master” blue panel and the small panel containing the blue and green knob widgets.

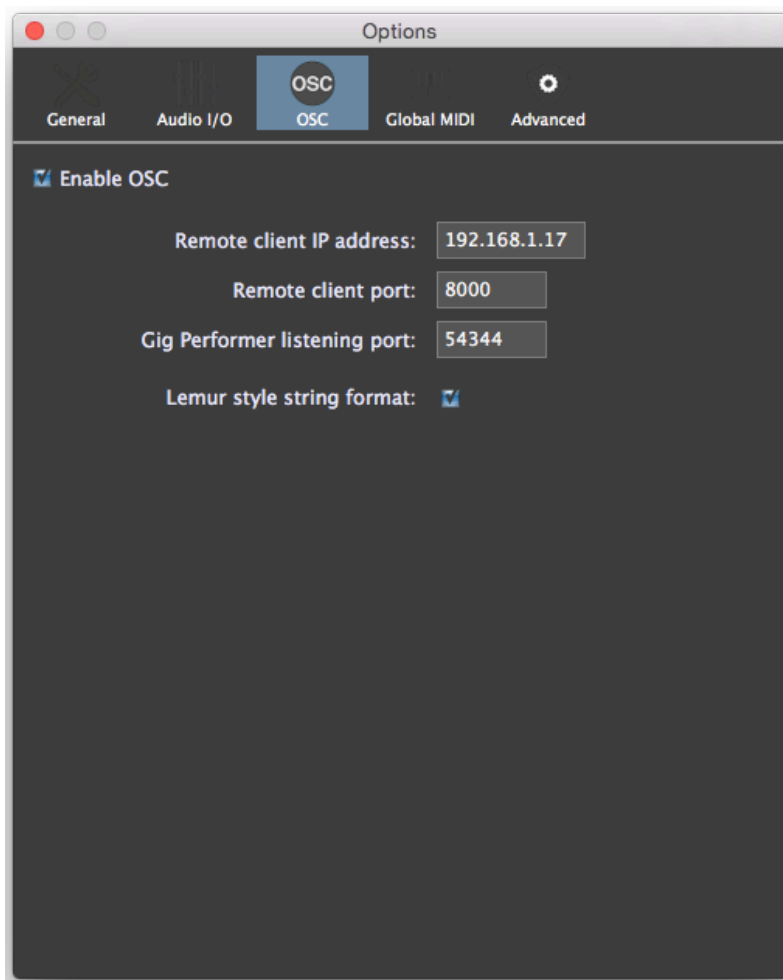
- 11. Test, tweak, and save.** Test that all is working as expected by moving widgets manually with mouse-dragging operations or from your MIDI controller. When you’re satisfied with all settings, be sure to save your work by going to File | Save As... and typing in a filename. Like all Gig Performer files, those saved when working within an instance are given the extension .gig and contain complete information about all rackspaces and variations. Bear in mind also that, because we created them in separate instances, the vocal signal chain created earlier in this tutorial continues to be completely active along with the guitar signal chain just created, and that making changes in one instance does not in any way affect other instances.

Important note: When multiple instances are open, be sure you exit from each one separately.

OSC tutorial

As described in [Appendix A](#) of this user guide, OSC (Open Sound Control) is a communications protocol that allows you to view and move widgets wirelessly from your iOS or Android tablet or smartphone. This tutorial will show you how to use Gig Performer with an OSC app.

- 1. Establish wireless communications.** For Gig Performer and an OSC app to work with one another, both your computer and the device running the app must be connected to the same wireless network, so ensure that this is the case.
- 2. Establish OSC communications.** Launch Gig Performer and then go to Options -> OSC setup... The dialog window shown below will appear:



Click on the Enable OSC button at the top so that it is checked in blue, then enter the IP address of the device running the OSC app in the “Remote client IP address” box, as indicated by the red arrow. (Consult your OSC app’s documentation app for directions as to how to locate the IP address.)

Gig Performer User Guide v 1.5.2

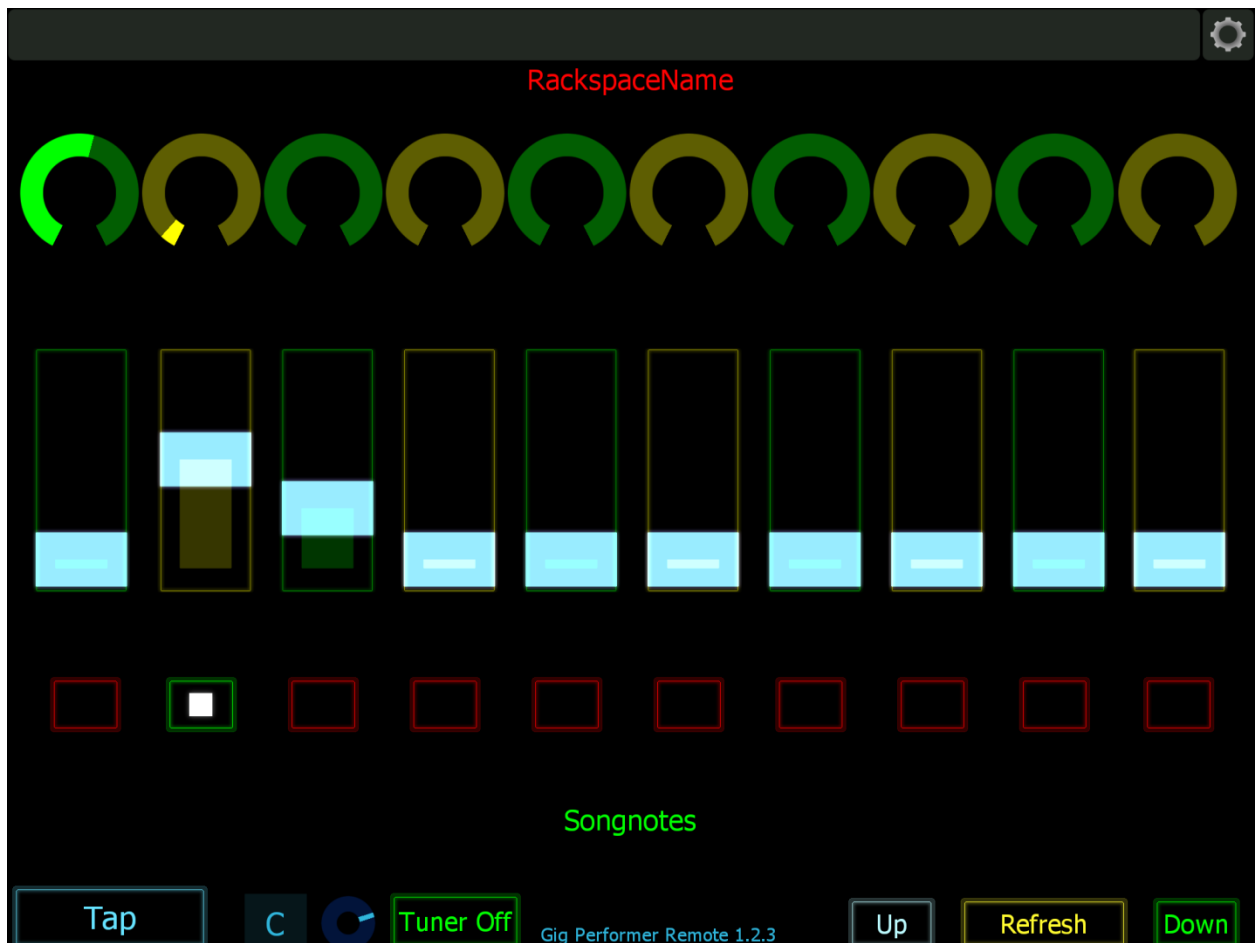
Next, enter the port number of the OSC app in the “Remote client port” box. (Consult your OSC app’s documentation app for directions as to how to locate the port number.) Note that the “Gig Performer listening port” value cannot be changed as this is permanently set. If you’re using the OSC app Lemur, check the “Lemur style string format” box. If you’re using any other OSC app, leave it unchecked. **When you’re done, close the dialog and quit Gig Performer.**

Note: If you’re running an OSC app on multiple devices—say a tablet and a smartphone—and you want to use both devices onstage (or if you want to give one to your FoH engineer to have access to your controls), there’s a simple way to accomplish that, using something called Broadcast Mode. Assuming you’re using standard class C addressing (with IP numbers beginning with 192.168.1.x), simply replace the last value of the OSC app’s IP address with 255 (i.e., 192.168.1.255) and restart Gig Performer. Now all devices whose IP addresses are on the same LAN (i.e., those devices with IP addresses all starting with 192.168.x) and are listening on the OSC app’s port number will communicate simultaneously with Gig Performer. [Click here](#) to access our Knowledge Base article with more information.

Gig Performer User Guide v 1.5.2

- If you're using Lemur or TouchOSC, download a Gig Performer template.** For your convenience, we have created templates that makes it easy to control Gig Performer widgets from the popular OSC apps Lemur and TouchOSC. If you are using either of these apps, [click here](#) to download the Lemur file, or [click here](#) to download the TouchOSC file. Note that these templates merely provide a starting point for your own OSC programming and you certainly are not required to use them—in fact, we encourage you to construct your own custom OSC environments.

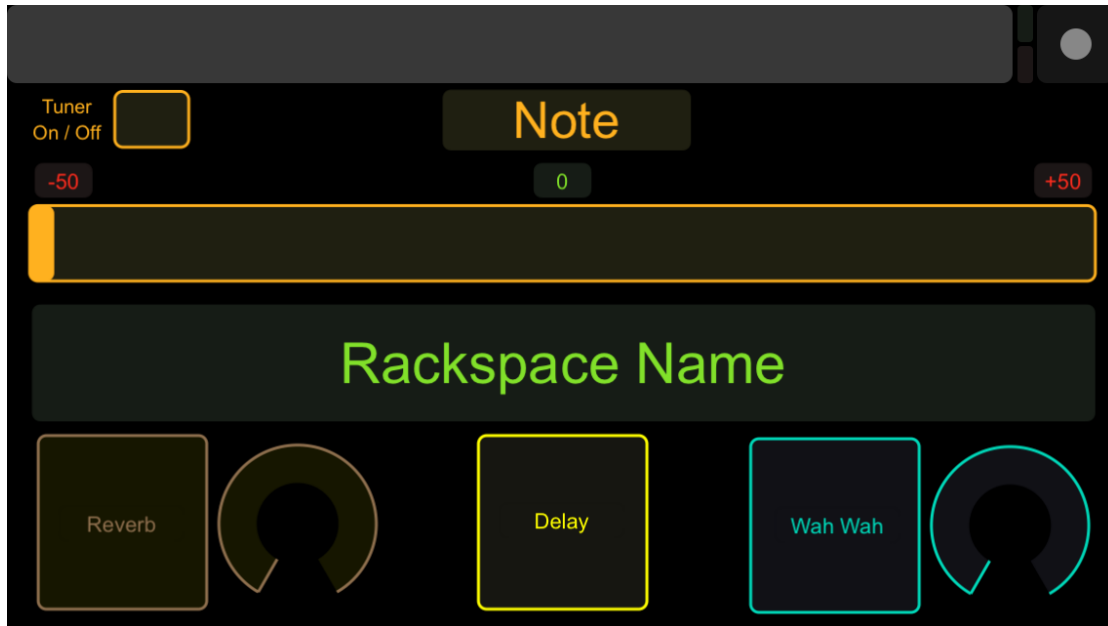
The Lemur template (as viewed in a tablet or smartphone) looks like this:



Along the top of the screen are ten knobs; underneath them are ten sliders, and underneath those are ten buttons. Below is a field named “Song Notes,” and at the bottom of the screen are dedicated buttons to set Tap Tempo, turn Tuner View on and off, and move Up and Down through rackspace and variations, plus a screen Refresh button (touching it after making any changes to a Gig Performer rackspace updates the screen).

Gig Performer User Guide v 1.5.2

The TouchOSC template, as viewed in an iPhone (the only device it can run on), looks like this:



Note that our Lemur template has been designed for general use, while the TouchOSC template has been designed specifically for guitarists. (An instructional video about using the Lemur template can be found [here](#), and an instructional video about using the TouchOSC template can be found [here](#).) Both templates display the currently selected rackspace and variation prominently; the Lemur template also displays the caption of a text label in the “Songnotes” field.

There’s absolutely no requirement you use either template (in fact, we encourage you to create your own custom OSC environments!) but if you’d like to experiment with them, relaunch Gig Performer (remember, you quit it [at the end of step #2](#) above) and then touch the Tuner On/Off button in the template. Gig Performer should immediately go to [Tuner View](#). (If for some reason the Tuner On/Off button doesn’t work, double-check that the directions in [steps #1 and #2 above](#) have been followed precisely to ensure correct bidirectional communications between the device running your OSC app and the computer running Gig Performer.)

In Lemur, if you have a microphone or instrument connected to one or more inputs of your audio interface, the virtual knob to the left of the Tuner On/Off button will move and the note number to the left of that knob will change to reflect the note coming in on the [channel\(s\) Gig Performer is currently “listening” to](#). In TouchOSC, the box at top center will display the note coming in, and the slider below it will show the tuning offset in cents, with the center position indicating that you are exactly in tune.

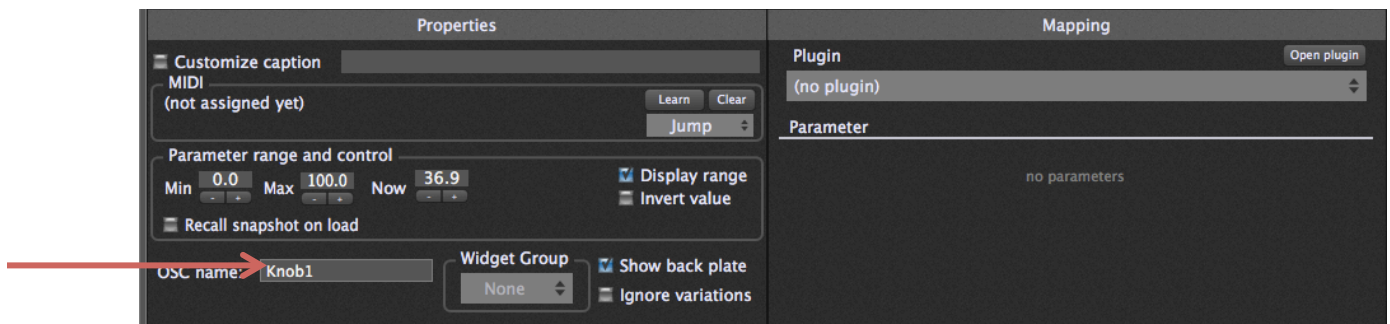
Gig Performer User Guide v 1.5.2

Touch the Tuner On/Off button again to return Gig Performer to Performer View. If you are using Lemur, touch the “Tap” button presented by the template several times. You should see the [Tempo display](#) in the upper-right hand corner of Gig Performer’s screen change bpm value as you do so; the shorter the intervals between taps, the higher (faster) the bpm value.

- 4. Give your widgets the appropriate OSC names.** 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. You can make up any names you like, as long as they are comprised of ASCII characters only, with no spaces. If you’re running Lemur or TouchOSC, here are the OSC names used in the templates we’ve created:

Object	Lemur template	TouchOSC template
Rotary knobs (Reverb and Wah Wah in Touch OSC template)	“Knob1” through “Knob 10”	“knob1” and “knob2”
Sliders / drawbars	“Slider1” through “Slider10”	(none)
Buttons (Reverb, Delay, Wah Wah in Touch OSC template)	“Button1” through “Button 10”	“button1”, “button2” and “button3” respectively
Text label caption	“Songnotes”	(none)

To use the templates, simply put Gig Performer into Edit mode, select a widget, and in the Widget Properties inspector, enter the one of these names in the “OSC name” field (as indicated by the red arrow), **then press Enter (this is important)**.



For example, if you’re using the Lemur template, add a knob widget and name it “Knob1”; if you’re using the TouchOSC template, name the knob widget “knob1”.

Gig Performer User Guide v 1.5.2

Then take Gig Performer out of edit mode and move the knob widget with your mouse. As you do so, the identically named knob in the Lemur or TouchOSC screen changes the same way. Because OSC is bi-directional (meaning that changes can be made in either the device running the OSC app or the host computer running the target software—in this case, Gig Performer), the opposite occurs too: You can move the onscreen knob in Lemur or TouchOSC and the knob widget in Gig Performer changes the same way. Try it!

Do the same with a switch or LED button widget, changing its OSC name to Button1 (for use with the Lemur template) or button1 (for use with the TouchOSC template). Exit edit mode and test to make sure that touching the corresponding button in your tablet or smartphone turns the associated Gig Performer widget on or off, and vice versa.

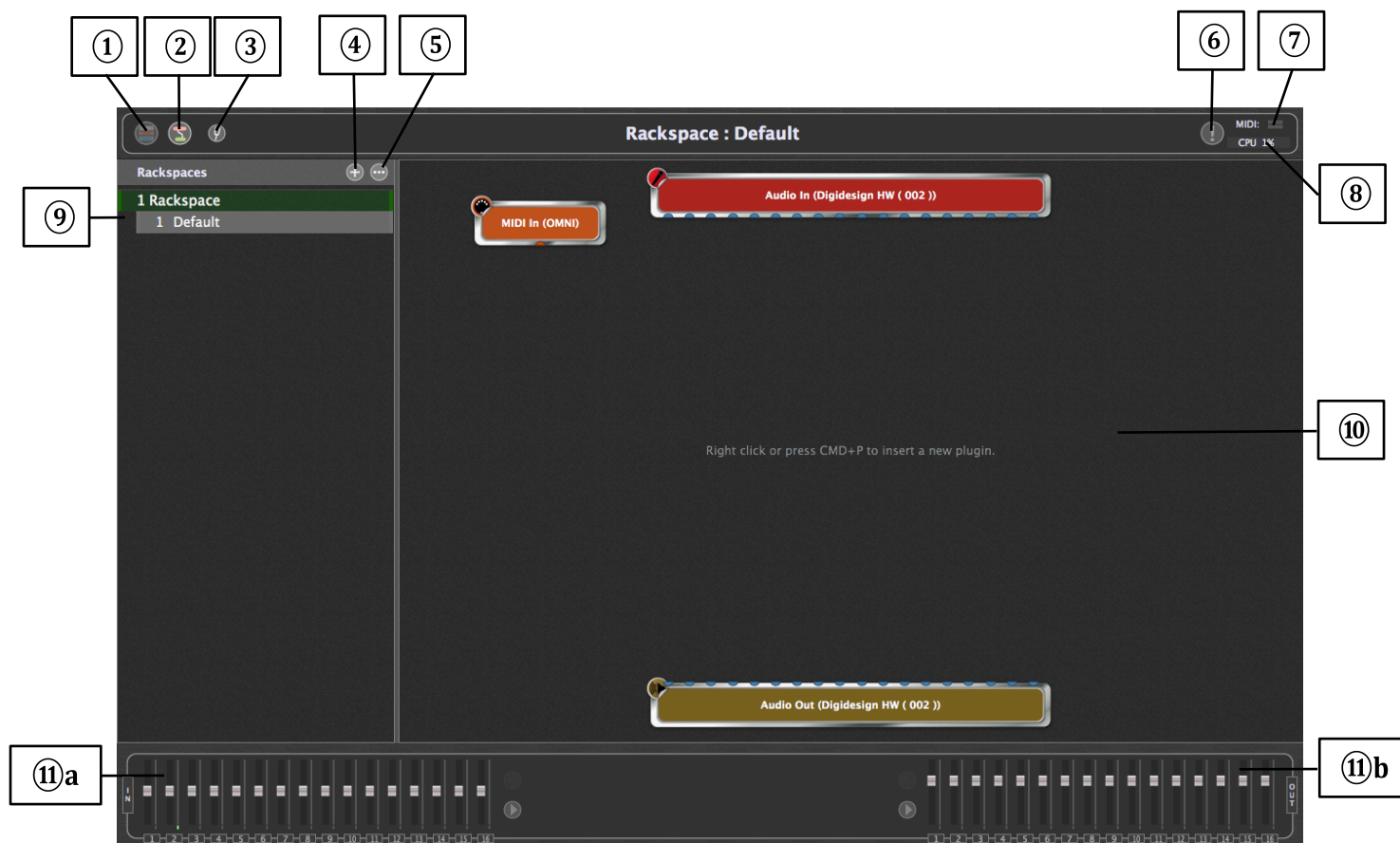
- 5. Make your existing rackspace OSC-compatible.** It's easy to make any rackspace OSC-compatible: All you have to do is assign OSC names to your widgets that match the names being used in the environments you create in your OSC app editor. For example, load any of the rackspace created in the [MIDI Tutorial](#) and [Audio Tutorial](#) presented earlier in this User Guide, go into edit mode, and select each widget in turn. The only change you have to make to each is in the "OSC name" field in the Properties area of the inspector; all MIDI device and plugin parameters remain the same. The only difference is that you can now move these widgets remotely from the tablet or smartphone running your OSC app, and vice versa (i.e., any widget adjustments you make from your laptop or from a connected MIDI controller are reflected in the tablet/smartphone screen). Try it for yourself!

Please note that this tutorial only scratches the surface of what can be accomplished when using an OSC app with Gig Performer. Again, the Lemur and TouchOSC templates provided on our website are merely a starting point and you certainly are not required to use them—in fact, we encourage you to construct your own custom OSC environments. Consult the documentation of your OSC app and [Appendix A](#) in this User Guide for more information.

REFERENCE

Screens

Connections View



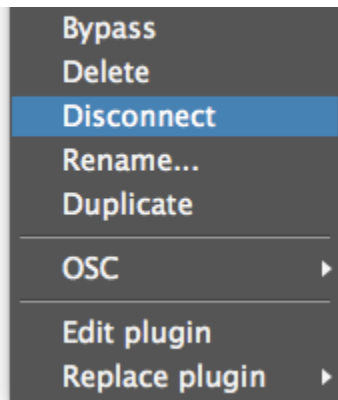
- ① **Performer View icon** – Click here to enter [Performer View](#).
- ② **Connections View icon** – Click here to enter Connections View.
- ③ **Tuner View icon** – Click here to enter [Tuner View](#) (the icon turns red). Click again to exit Tuner View.
- ④ **Add rackspace (“+”) icon** – Click here to add a new rackspace.
- ⑤ **“Three dots” icon** – Click here to open the [“More” menu](#).
- ⑥ **Panic icon** – Click on this to issue a “turn all notes off” message on all MIDI channels, along with a sustain OFF message and a Center PitchBend message, thus stopping stuck notes in their tracks.

⑦ **MIDI activity indicator** –Flickering here indicates the presence of incoming MIDI messages. Open the [MIDI Monitor](#) window if you want to see detailed information about the MIDI data being received.

⑧ **CPU usage indicator** –Shows the percentage amount of CPU overhead currently being used by Gig Performer for audio processing. (Actually, Gig Performer itself is only responsible for a very 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.

⑨ **Rackspace and variations list** –A list of all rackspaces and variations that have been created. The current rackspace is outlined in green. Rackspaces can easily be reordered by dragging them up and down using your mouse or trackpad.

⑩ **Plugin connections area** –This large area of the screen is used to display your plugins and their interconnections. ([Click here](#) for more information.) Plugins can be freely moved and repositioned by dragging them with your mouse or trackpad. Double-clicking on any plugin opens its plugin editor. Right-clicking on any plugin opens a popup menu that allows you to do the following:



Bypass - Places this plugin in bypass mode. [Click here](#) for more information.

Delete – Removes the plugin.

Disconnect - Removes all connections to and from this plugin.

Rename – Changes the displayed name (useful if you are using the same plugin more than once).

Duplicate – Creates a duplicate of the plugin.

OSC – Allows you to set an [OSC name](#).

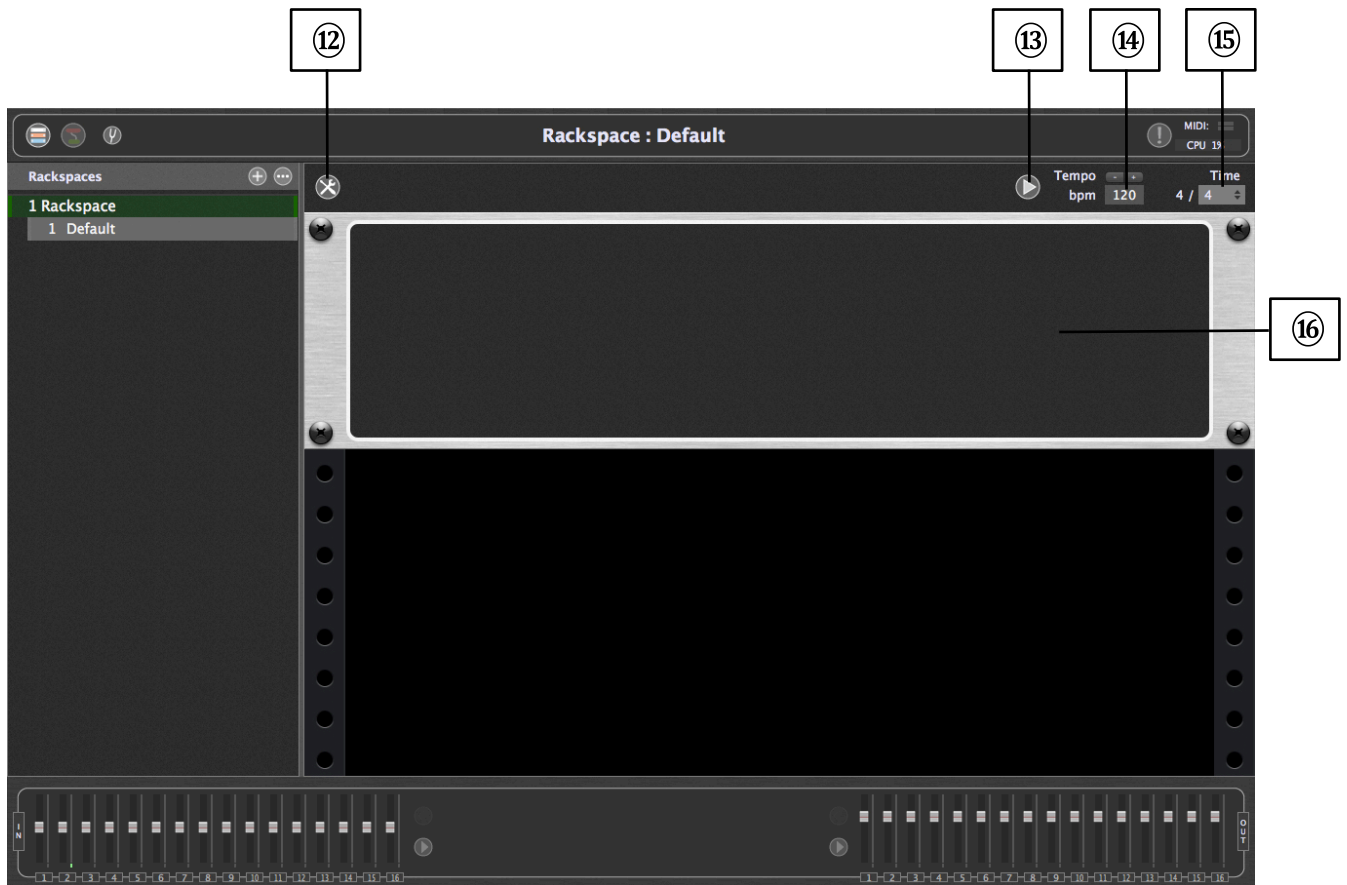
Edit plugin – Opens the plugin’s editor.

Replace plugin – Replace the plugin with a different plugin.

⑪a **Input level display** –A set of meters and sliders that allow you to view and adjust the levels of your audio interface’s various inputs. (Arrows will appear to allow scrolling if there are more than 16 inputs.) Flickering here indicates the presence of input signal. Double-click on any fader to set it to 0 dB.

⑪b **Output level display** – A set of meters and sliders that allow you to view and adjust the levels of your audio interface’s various outputs. (Arrows will appear to allow scrolling if there are more than 16 outputs.) Double-click on any fader to set it to 0 dB.

Performer View



⑫ **Widget Layout Designer icon** – Click here to enter Edit Mode.

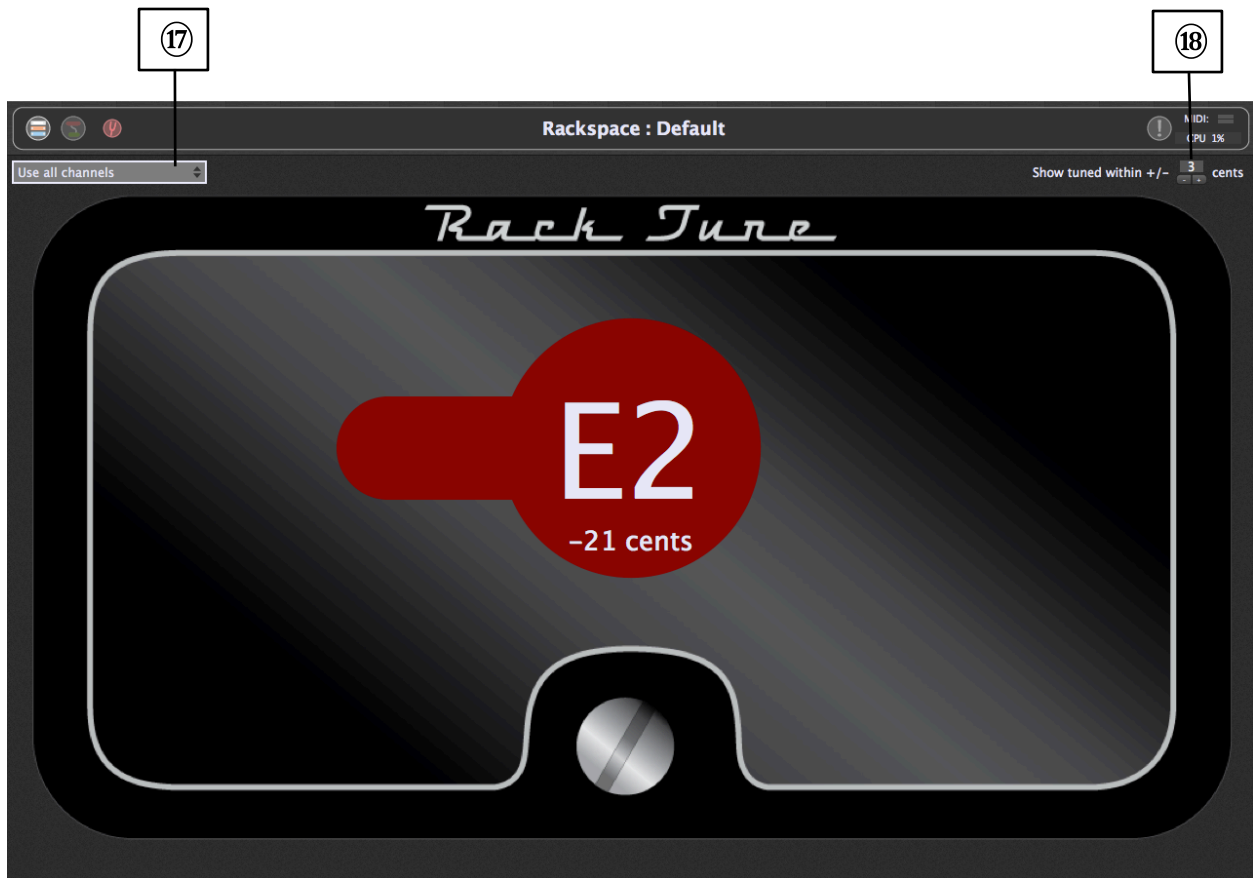
⑬ **Play/Stop button** – When engaged (“play”), all plugins that provide playback control (for example, sample, loop, or beat playback plugins) begin playing back. When disengaged (“stop”), they are all put in a stopped state.

⑭ **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 override this by entering a new value, or by using the increment/decrement buttons below the display.

⑮ **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 override this by clicking on the dropdown menu, which enables you to enter any time signature, up to 99/64.

⑯ **Rackspace** – One or more panels containing the widgets in the current rackspace and variation. (You may need to scroll down if there are many panels being used.)

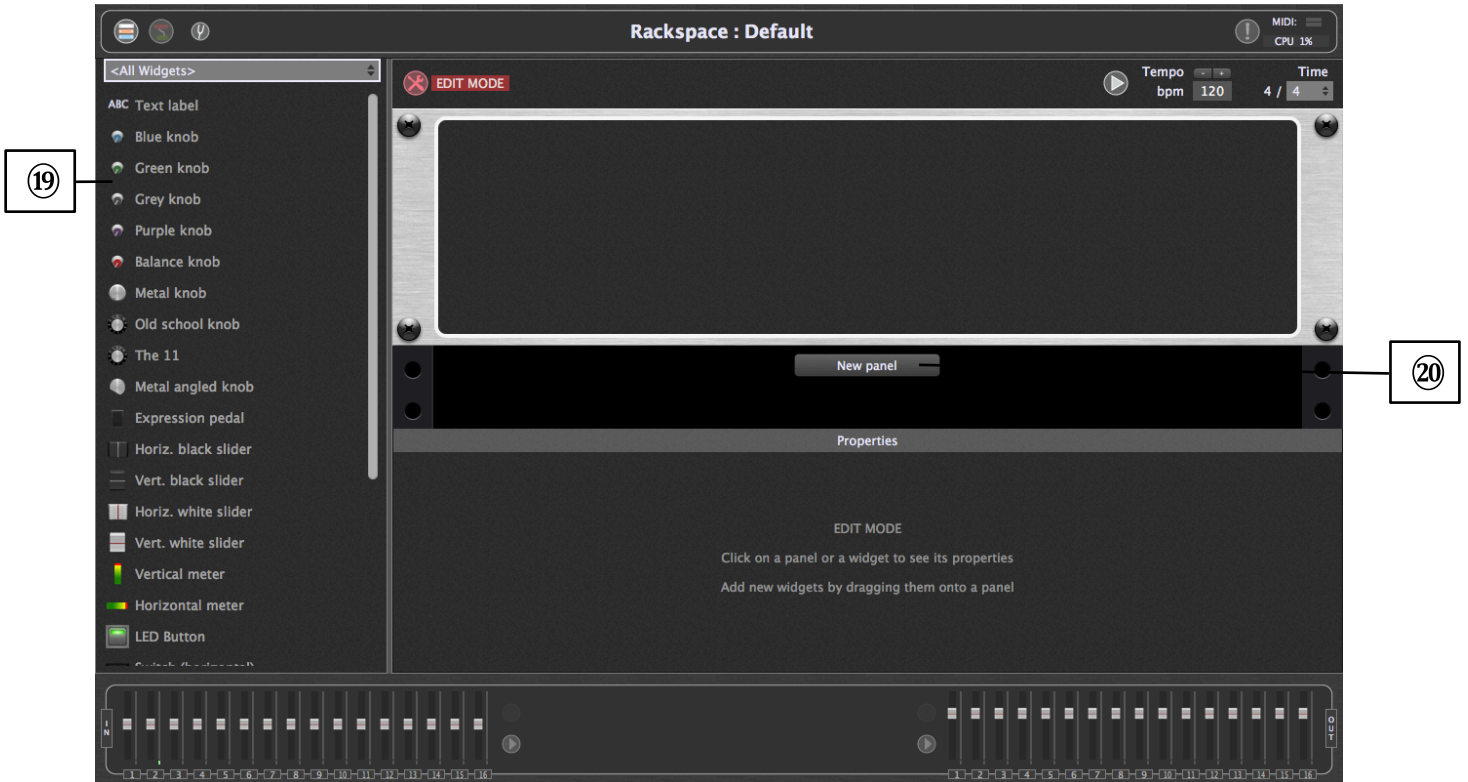
Tuner View



①⑦ **Use all channels** – Click here for a dropdown menu if you want to have the Tuner “listen” to 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.

①⑧ **Show tuned within** – Allows you to specify a range within which the main display changes from red (out of tune) to green (in tune). Available values are +/- 1 cent to +/- 5 cents; default is +/- 3 cents.

Edit Mode screen



①⑨ **Widget list** – Shows a list of [widgets](#). A dropdown menu at the top of this list allows you to view just certain types of widgets. These include:

Labels –Inactive widgets that simply allow labeling.

Knobs of different colors and designs (including our Spinal Tap “The 11”—it goes to 11!) – Active widgets that provide continuous control.

Expression pedal – Another continuous control, visually emulating the movement of an expression pedal.

Sliders and organ drawbars of different colors and designs – Active widgets that provide continuous control.

Meters (one vertical and one horizontal) – Used to provide continuous [level metering](#).

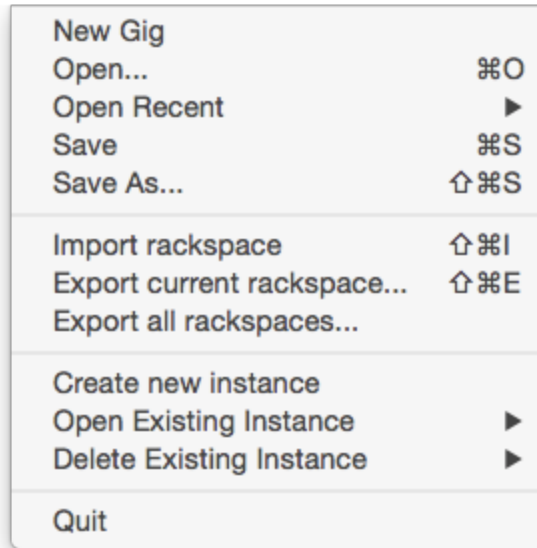
Buttons and switches of different colors and designs. – All of these active widgets provide simple on-off control.

②⑩ **New panel button** – Click here to add a new panel to the current rackspace.

Gig Performer User Guide v 1.5.2

Main Menus

File



New Gig – Clears all memory and opens a new rackspace with all default values.

Open... – Opens a saved .gig file.

Open Recent – Provides a popup menu listing the five most recent .gig files.

Save – Saves your work in a .gig file.

Save As... – Saves your work in a .gig file but allows you to enter a new name first (thus preventing previously saved .gig files from being overwritten).

Import rackspace – Allows you to import a single rackspace from a list of previously exported rackspaces. (See below.) The newly imported rackspace is simply added to your currently created rackspaces—it does not overwrite or delete any of your work. Note that you can also drag and drop one or more rackspaces from a folder directly into Gig Performer, making it fast and easy to create a complete new setlist from existing rackspaces.

Export current rackspace... – Allows you to export a single rackspace for use in another Gig Performer project.

Export all rackspaces... – Allows you to export all rackspaces for use in another Gig Performer project.

Gig Performer User Guide v 1.5.2

(Note: The following three menu items appear only when the "[Enable multi-instance support](#)" option is checked in the Advanced tab of the Preferences dialog.)

Create new instance – Allows you to create a new instance. (For more information [click here](#).)

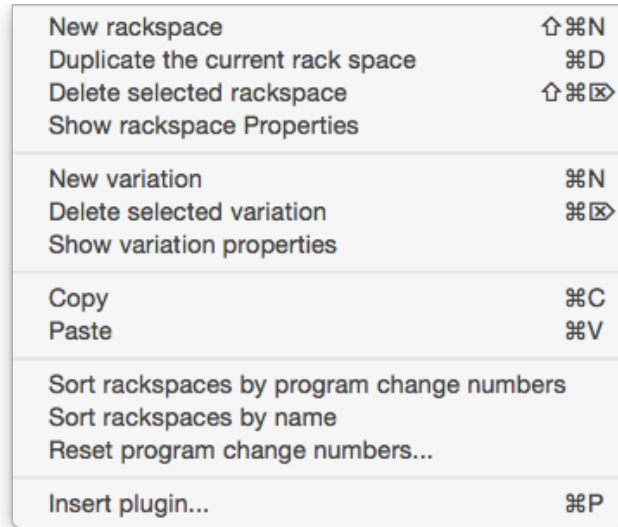
Open Existing Instance – A popup menu allows you to open a previously created and saved instance. (For more information [click here](#).)

Delete Existing Instance – A popup menu allows you to delete a previously created and saved instance. (For more information [click here](#).)

Quit – Quit Gig Performer. If you have any unsaved work, a dialog will ask you if you want to save your data first.

Gig Performer User Guide v 1.5.2

Edit



New rackspace – Creates a new rackspace.

Duplicate the current rack space – Duplicates the current rackspace.

Delete selected rackspace – Deletes the currently selected rackspace.

Show rackspace properties... – Opens the [Rackspace Properties](#) dialog.

New variation – Creates a new variation.

Delete selected variation – Deletes the currently selected variation.

Show variation properties... – Opens the [Variation Properties](#) dialog.

Copy – Copies the current selection to the Clipboard.

Paste – Pastes the data in the Clipboard.

Sort rackspaces by program change numbers – An easy way to sort your rackspaces by the program change numbers associated with each. (For more information [click here](#).)

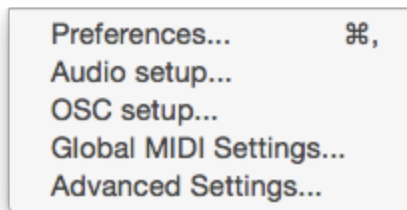
Sort rackspaces by name – Sorts your rackspaces alphabetically.

Reset program change numbers... – Resets all the program change numbers associated with all rackspaces and variations so that the first variation in the first rackspace is assigned Program Change 1 and the first variation in the second rackspace is assigned Program Change 2, etc. (For more information [click here](#).)

Insert plugin... – Opens the [Quick plugin finder](#). (Gig Performer will switch to Connections View automatically if you are currently in Performer View.)

Gig Performer User Guide v 1.5.2

Options



Preferences... – Opens the [General tab](#) of the Preferences dialog.

Audio setup... – Opens the [Audio I/O tab](#) of the Preferences dialog.

OSC setup... – Opens the [OSC tab](#) of the Preferences dialog.

Global MIDI Settings setup... – Opens the [Global MIDI tab](#) of the Preferences dialog.

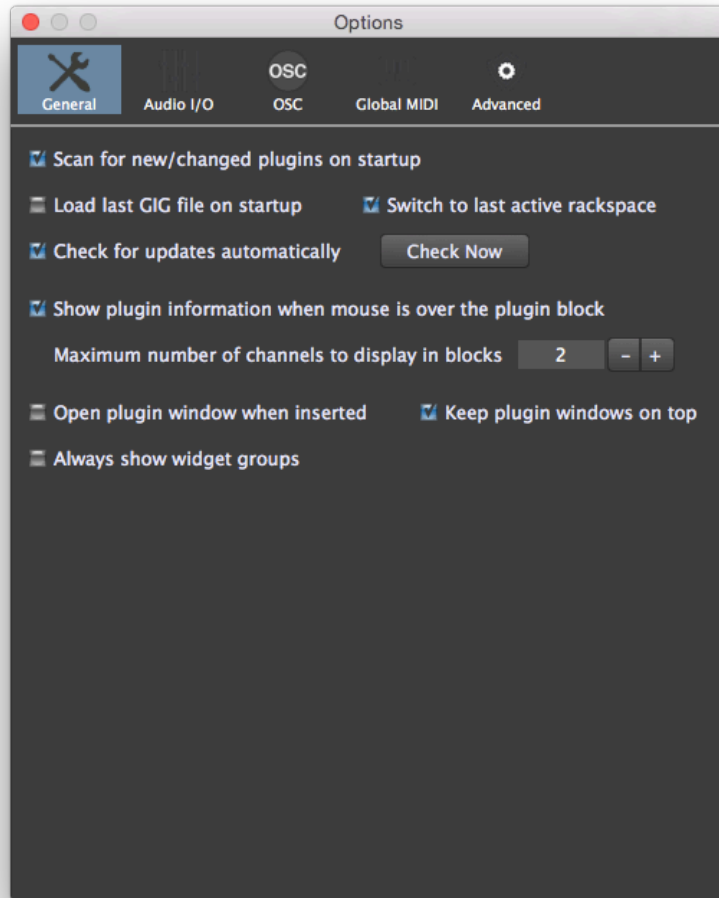
Advanced Settings setup... – Opens the [Advanced tab](#) of the Preferences dialog.

The Gig Performer options dialog is divided into five sections:

- General
- Audio I/O
- OSC
- Global MIDI
- Advanced

A full description of each is provided on the following pages.

Preferences | General tab



Scan for new/changed plugins on startup –When enabled, Gig Performer will check for new or updated plugins every time it starts. It’s generally a good idea 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 isn’t appearing, not scanning for plugins is the most probable cause. (See [Troubleshooting](#) for more information.)

Load last GIG file on startup – When enabled, Gig Performer will load the last .gig file that was used.

Switch to last active rackspace – When enabled, Gig Performer will switch to the rackspace that was active when the .gig file was last saved. Otherwise, the first rackspace will be activated.

Gig Performer User Guide v 1.5.2

Check for updates automatically – Unless you're actually on tour, it's probably useful to keep this option enabled so that you are informed of updates.

Check Now – Click on this button to manually check for updates. **Note:** This requires an internet connection.

Show plugin information when mouse is over the plugin block – When enabled, moving the mouse over blocks in the connection view will cause a popup tooltip to appear with information about the plugin.

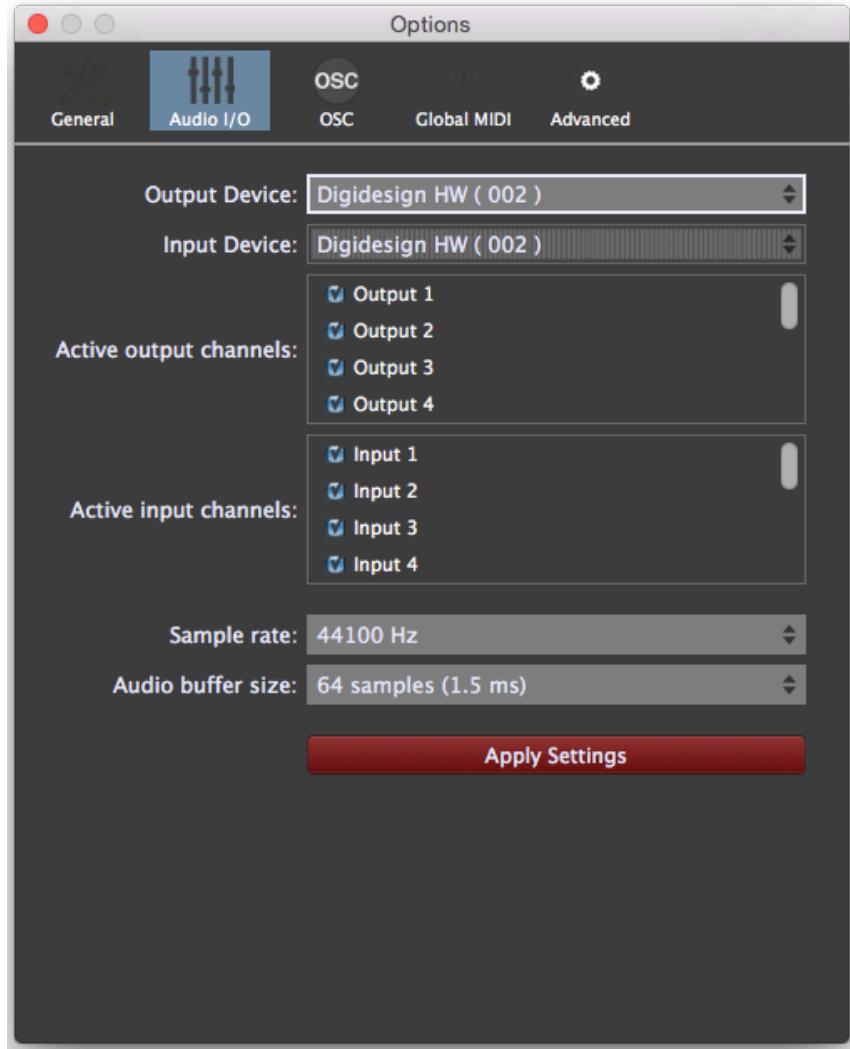
Maximum number of channels to display in blocks – This option prevents plugins from showing more than a specified number of audio channels. Some plugins have a large number of audio channels, most of which may never be used, making their blocks unwieldy to view and causing them to take up a lot of space in Connections View. Note that the value entered here does not apply to the audio interface Audio In and Audio Out blocks.

Open plugin window when inserted – When checked, the plugin editor window will open automatically whenever a plugin is inserted. Disable this feature if you prefer to create many 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

Always show widget groups – When you gang multiple widgets together, the widget group to which they belong (lettered A through Z) is displayed on the top left of the widget while editing. If you would like that widget group name to stay on while NOT editing, enable this option.

Preferences | Audio I/O tab



Output Device – Select the audio output device you want to use. If you’re running Gig Performer on a Mac OSX system and have created any Aggregate Devices (consisting of multiple interfaces) in the Audio MIDI Setup utility, it will appear here as an option as well.

Input Device – Select the audio input device you want to use. If you’re running Gig Performer on a Mac OSX system and have created any Aggregate Devices (consisting of multiple interfaces) in the Audio MIDI Setup utility, it will appear here as an option as well.

Note: Changing the audio output or input device will cause your entire .gig file to be reloaded immediately, so we recommend making such changes immediately after creating a new .gig file.

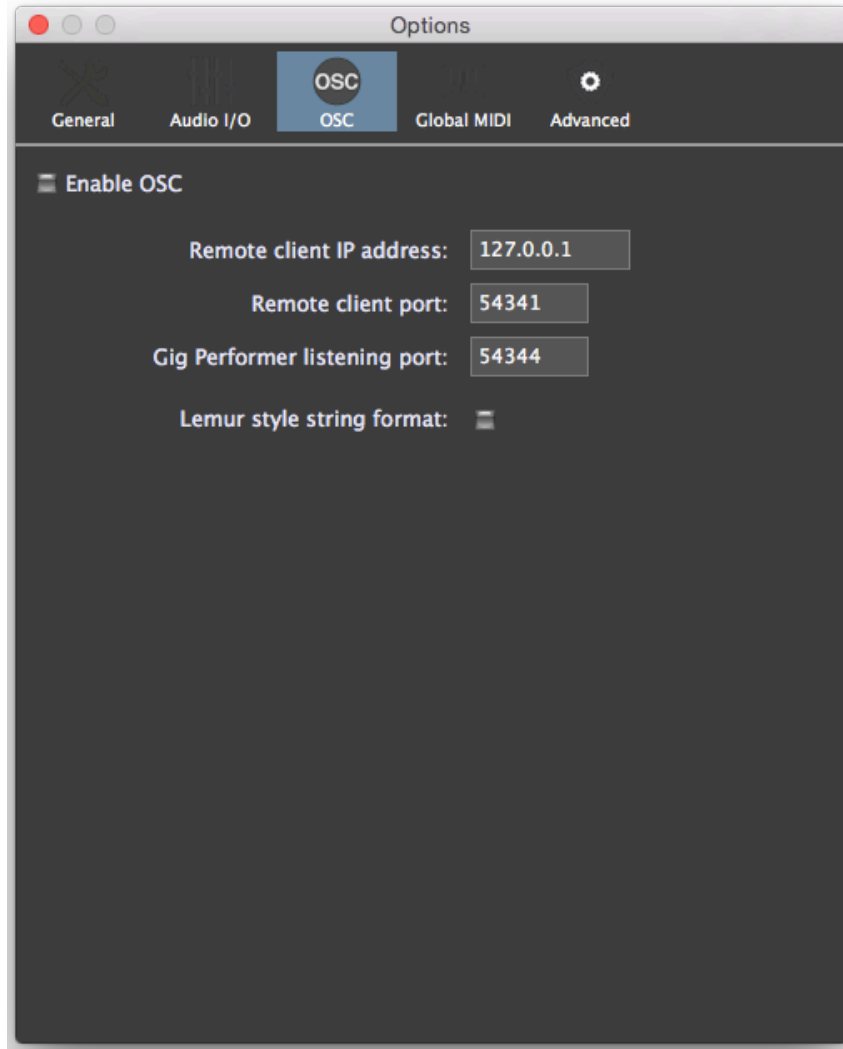
Gig Performer User Guide v 1.5.2

Active output channels/Active input channels – Determines how many input and output channels are to be made available to Gig Performer. If you are using 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) channels that 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

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 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](#).

Preferences | OSC tab



Enable OSC – Check to turn on OSC. [Click here](#) for more information about available OSC commands.

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).

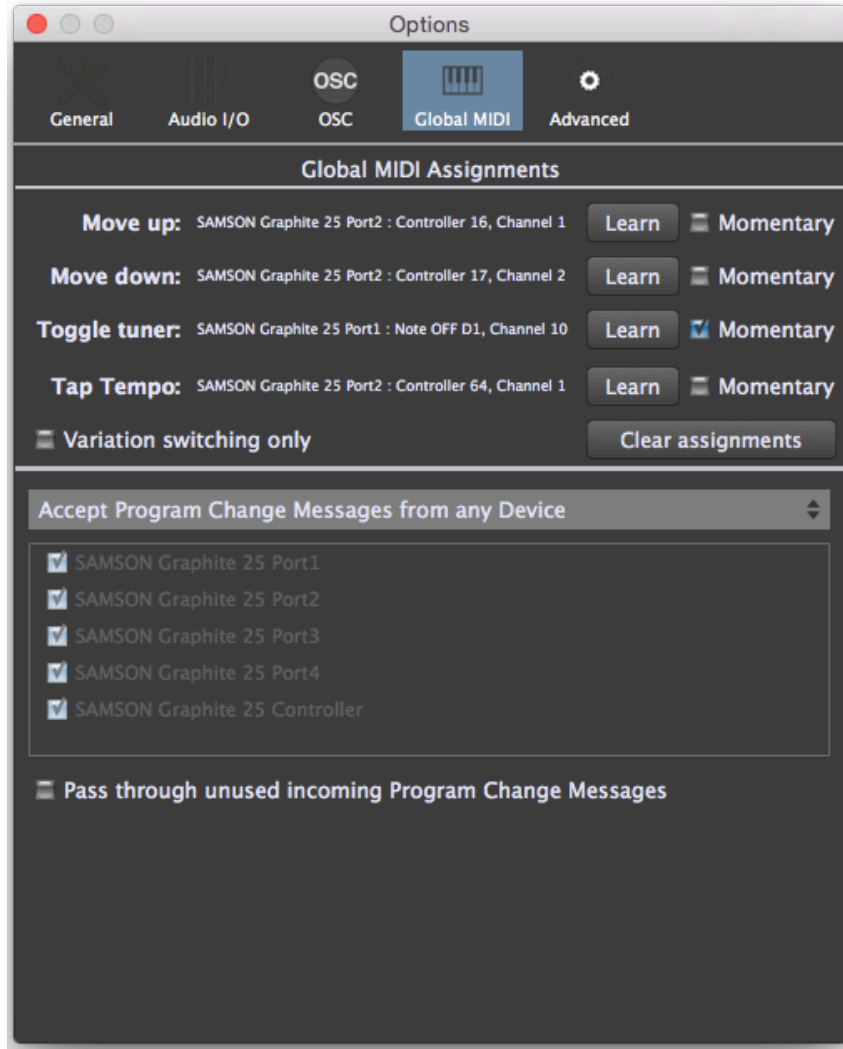
Gig Performer listening port – The port on which Gig Performer is listening for incoming OSC messages over UDP (User Datagram Protocol). **Note:** This is permanently set to 54344.

Gig Performer User Guide v 1.5.2

Lemur style string format – Gig Performer automatically sends some text and label contents to remote devices. If you are using Lemur then you will want to enable this checkbox so that Gig Performer prefixes strings with the '@content' attribute which Lemur expects. [Click here](#) for more information.

Important note: If you change any settings in the OSC tab, you will need to restart Gig Performer in order to apply them. We expect to remove this restriction in a future version.

Preferences | Global MIDI tab



In this dialog, Gig Performer can learn MIDI messages to accomplish the following actions:

Move up – Move to the previous variation, or to the previous rackspace if you are at the first variation.

Move down – Move to the next variation in the current rackspace, or to the first variation of the next rackspace if you are at the last variation of the current rackspace.

Toggle tuner – Switch to or from Tuner View.

Gig Performer User Guide v 1.5.2

Tap Tempo – Gig Performer can receive taps from a pedal or button on a connected MIDI controller or via OSC messages 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 button and tap on the button or pedal you want to use for setting tempo. (You will most likely need to enable the **Momentary** option as well.) Once you start tapping, the BPM display at the top right of the main Performer View screen will display the current tempo that has been detected. All plugins that respond to global tempo will now synchronize to this tempo. (It may be necessary to enable ‘sync’ to external tempo on individual plugins. Please check your plugin documentation for more information.) **Note:** The tempo used by connected plugins is always the global tempo set here unless it has been overridden in the [Rackspace Properties](#) dialog.

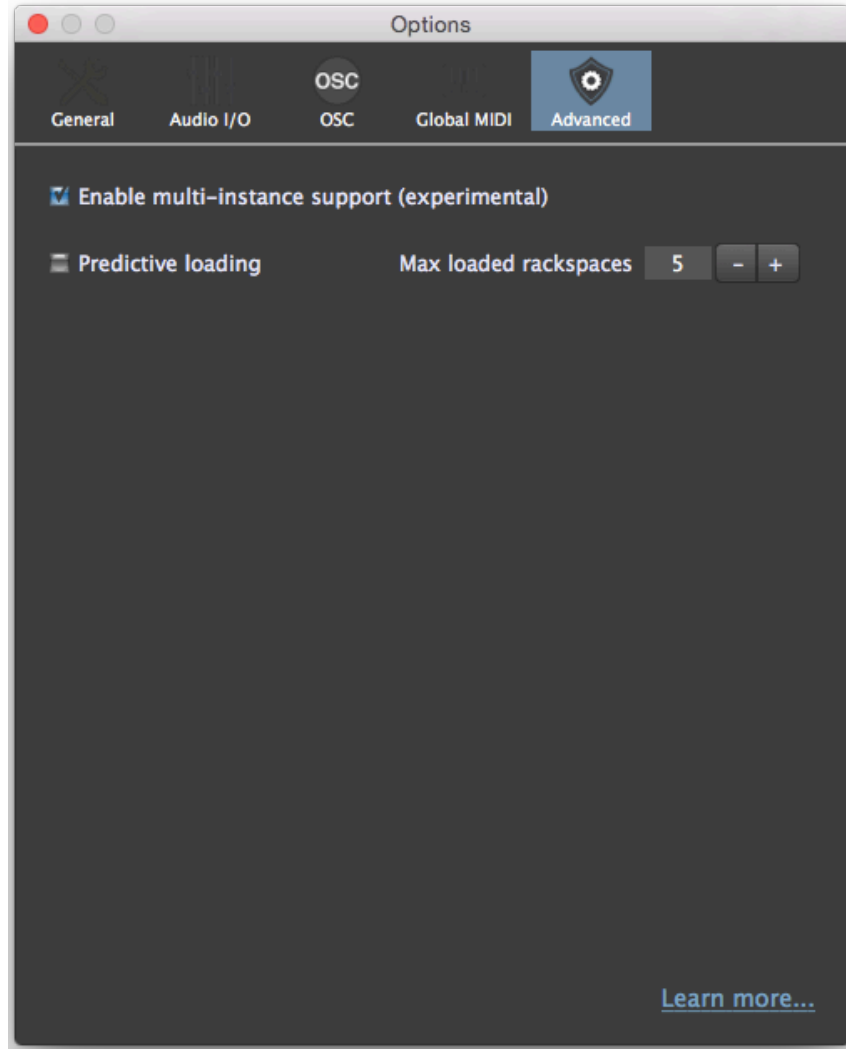
Variation switching only – If this option is enabled, you will only be able to move up or down within a single rackspace.

Clear assignments – Clear all learned assignments. (**Note:** Holding the SHIFT key down while clicking any of the Learn buttons clears just that assignment.)

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 devices.

Pass through unused incoming Program Change messages – When enabled, if a program change is received for which there is no corresponding rackspace, the message will instead be sent to the first plugin in the current rackspace. For example, if you have created 10 rackspaces and a program change message of 11 or higher is received, it will not cause the rackspace to change but will instead affect the first plugin in the current rackspace. Most plugins respond to incoming program change messages by changing presets; however, the behavior is entirely dependent upon the particular plugin. Consult your plugin’s documentation for further information.

Preferences | Advanced tab



Enable multi-instance support (experimental) – Checking this option gives you the ability to launch Gig Performer multiple times in the same computer, with each such “instance” operating separately but independently.

This has many different practical applications. For example, each bandmember can have the ability to change the rackspace affecting their particular instrument without changing the rackspace affecting different instruments; they can even use separate MIDI controllers or pedalboards to do so. Instances also make it far more efficient to process vocals, since in many circumstances only a couple of rackspace are required to manage all vocal sounds. If those rackspace live in a separate instance, you no longer have to duplicate them in every single rackspace you create.

Gig Performer User Guide v 1.5.2

Keyboardists that want to dedicate one of their controllers (such as an 88-note weighted keyboard) to one particular sound—say a grand piano—will also find instances useful. Simply create an instance with just a single rackspace containing the grand piano plugin and the appropriate MIDI In block. Now you no longer have to include that controller and grand piano plugin in every single rackspace you create, something that can help simplify your overall design and be a real timesaver. 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 will be able to successfully have open at any one time is dependent upon your system's RAM and CPU resources. Each instance can access entirely different plugins and MIDI In blocks; however, their Audio In/Out blocks must be consistent. (In other words, separate instances cannot access different audio interfaces.)

Note that when multi-instance support is enabled, three new options are added to the Main “File” menu, allowing you to create new instances, and to open or delete existing instances.

A demonstration of how to implement instances is provided in the [Audio tutorial](#) in this user guide.

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 amount of samples can use many gigabytes of RAM to hold the required data.

In normal operation, Gig Performer loads all plugins for all rackspaces 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 alternative—loading just one rackspace (i.e., one set of plugins) at a time—would certainly reduce RAM and CPU requirements considerably, but at the cost of having to wait while each new rackspace is loaded.

Enabling the “predictive loading” option provides a creative solution: Load only those rackspaces that will be needed in the immediate future (e.g. while playing the current song). Other rackspaces 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.

Gig Performer User Guide v 1.5.2

Predictive loading provides glitch-free instant switching among those rackspace 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 rackspace 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 rackspace (see [Max loaded rackspace](#) below). Note that rackspace 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](#).

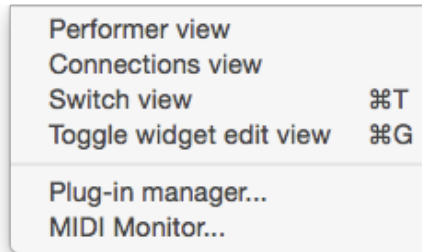
Important note: While we have tested predictive loading extensively, it is impossible to test it with all of the thousands of plugins out there. We therefore recommend that you test this feature ahead of time before using predictive loading in a live situation to ensure that the particular plugins you want to use are stable enough to withstand repeated initialization in a short period of time

Max loaded rackspace – Specifies the maximum number of rackspace that will be pre-loaded around the currently selected rackspace. The default value is 5, where the current rackspace as well as the next two and previous two rackspace are also loaded and instantly accessible. Optimal use of this feature assumes you will be moving from your current rackspace to rackspace 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 rackspace 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.

Learn more – A link to a Gig Performer [website article](#) about the Advanced options described above.

Gig Performer User Guide v 1.5.2

Window



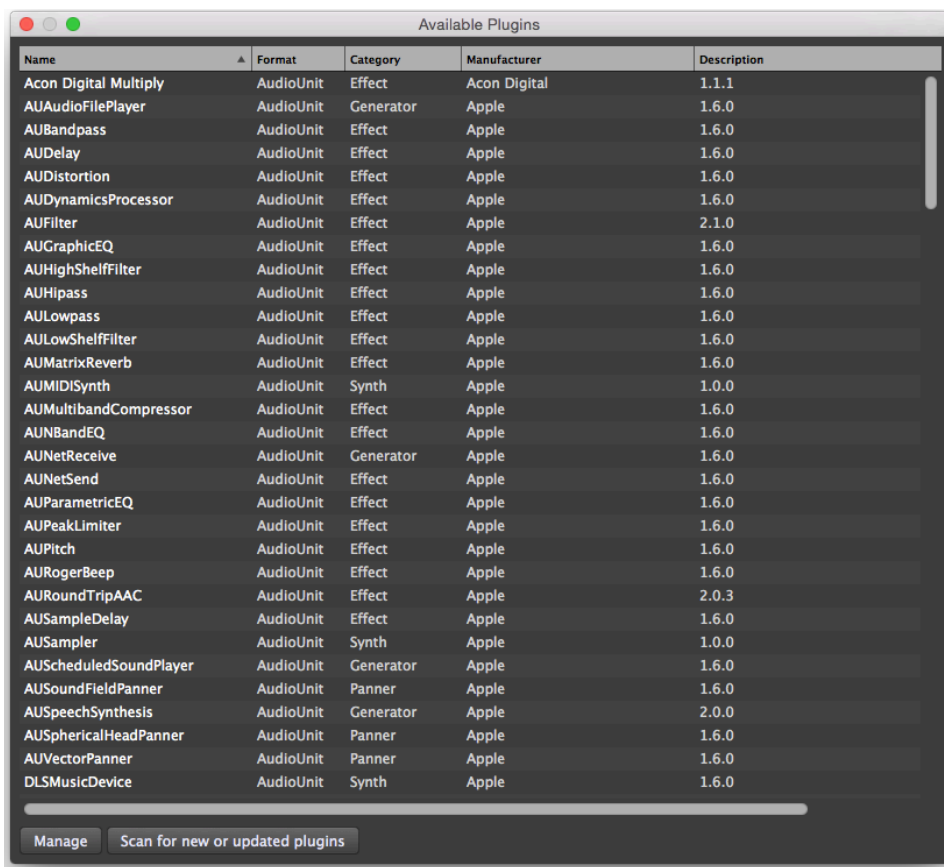
Performer view – Switch to Performer View.

Connections view – Switch to Connections View.

Switch view – Switches to Performer View if you are currently in Connections View, and vice versa.

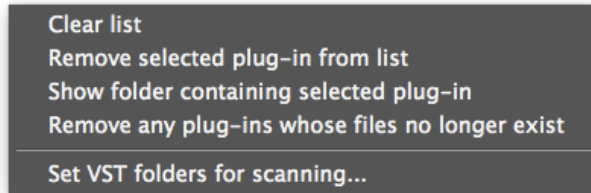
Toggle widget edit view – Takes Gig Performer in and out of Edit mode.

Plug-in manager... – Opens the Plugin Manager (shown below).



Gig Performer User Guide v 1.5.2

Clicking on the **Manage** button opens the following popup menu:



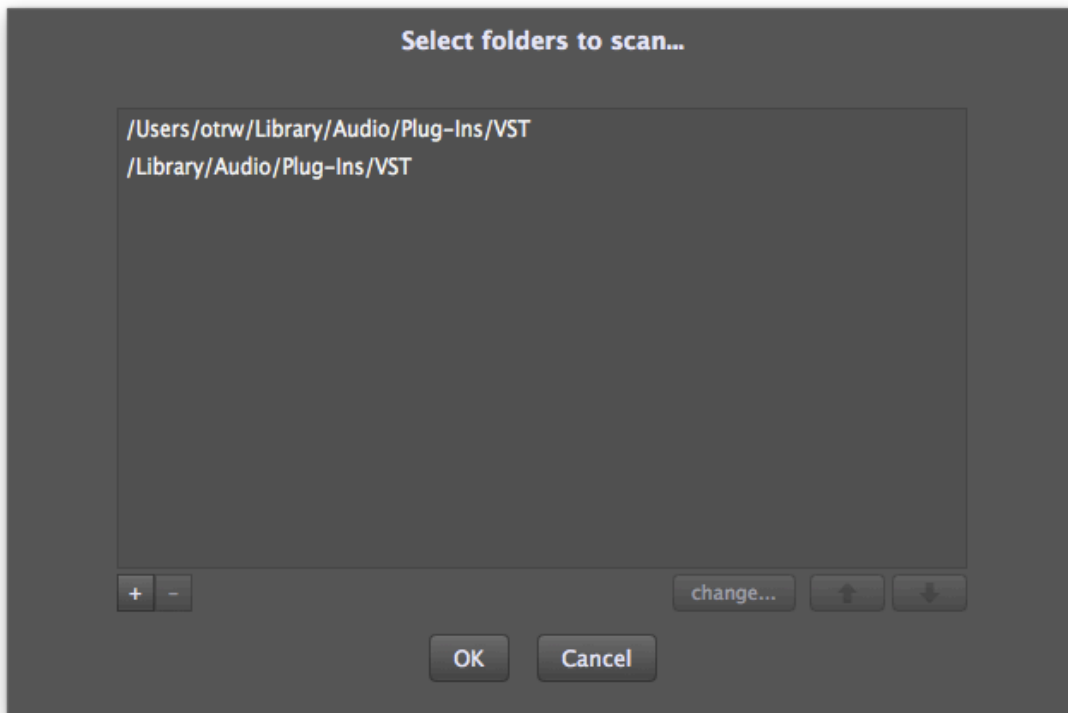
Clear list - Clears the listing of all plugins. (**Note:** While this does not actually erase any plugins from your computer, it does remove them from Gig Performer's memory, so a new scan (easily accomplished by clicking on the "Scan for new or updated plugins" button next to the Manage button) is required after this operation.)

Remove selected plugin from list - Select a plugin first to make this operational.

Show folder containing selected plugin- Select a plugin first to make this operational.

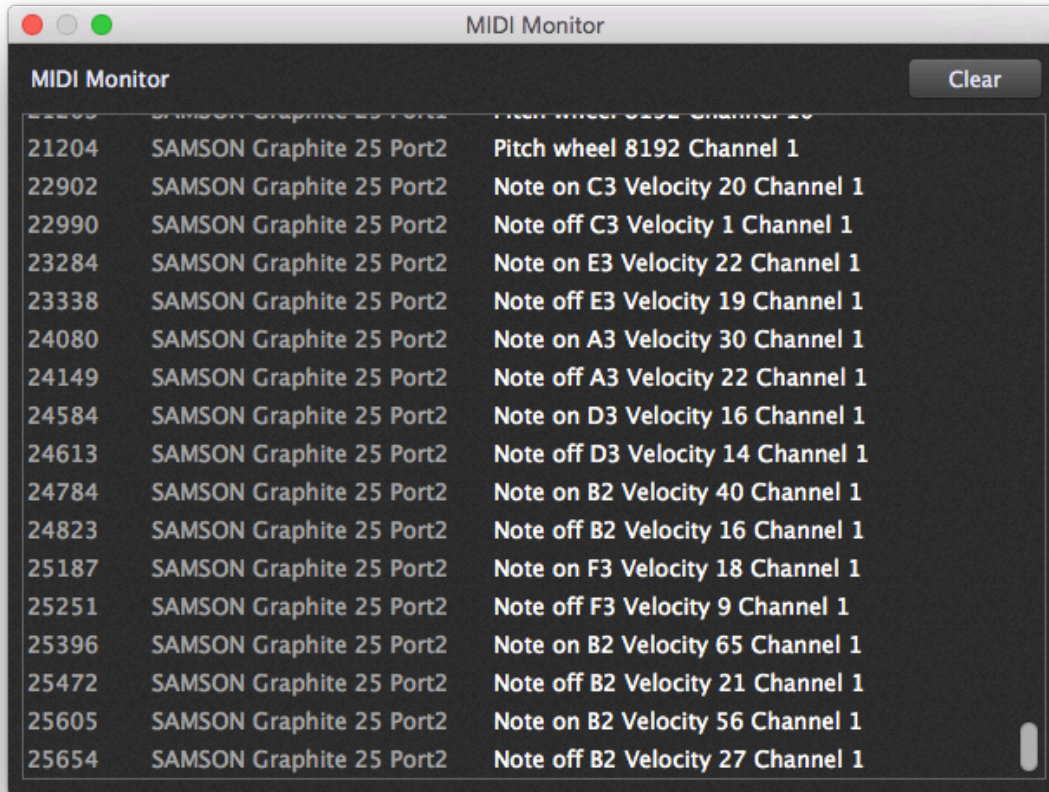
Remove any plugins whose files no longer exist - Use this to clean up your plugins list.

Set VST folders for scanning... - Opens the window shown below, which allows you to change, add, or delete the VST plugin folders to be scanned. (See [Troubleshooting](#) for more information.)



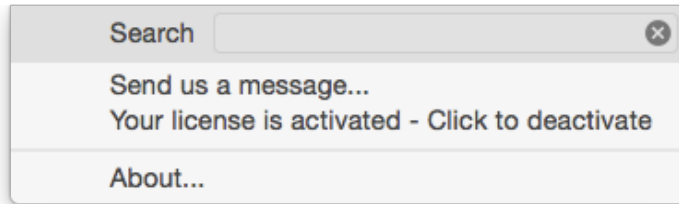
Gig Performer User Guide v 1.5.2

MIDI Monitor– Opens the MIDI Monitor window (shown below). This is functionally identical to the MIDI Monitor plugin, except that it shows all MIDI messages being received by Gig Performer.



Gig Performer User Guide v 1.5.2

Help



Search – Opens a Search dialog in which you can type your Search subject.

Send us a message... – Opens a Feedback Form dialog in which you can email us a detailed description of your problem, suggestion, bug report, or question. If you do experience a problem with Gig Performer, we strongly urge you to let us know about it so that we can fix it for the benefit of all users.

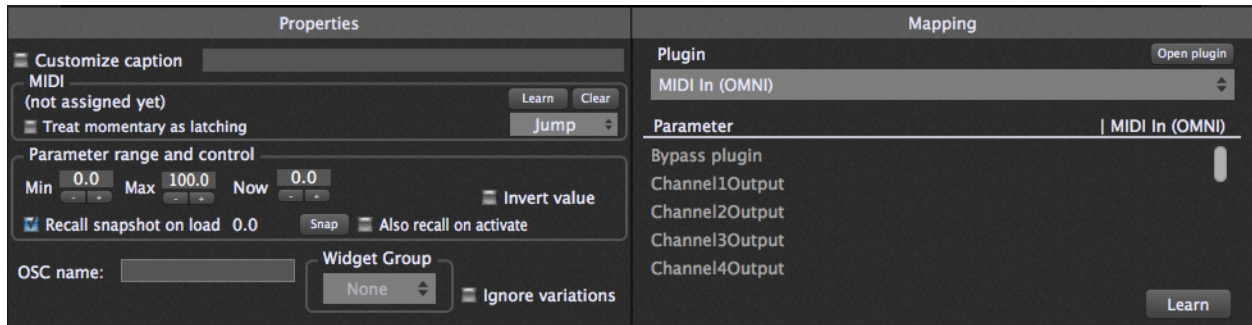
License activation/deactivation – Opens a dialog which allows you to activate your license or, if it is already activated, deactivate your license. **Warning:** Do not deactivate your license unless you are sure that you want to run Gig Performer on another system.

About... – Opens a dialog that shows the current version of Gig Performer, along with a link to our website.

Other Menus and Dialogs

Widget Properties inspector (Edit mode)

Adding or double-clicking on a widget in Edit mode opens the Widget Properties inspector:



Properties section

Customize caption – Check to have the widget display a custom caption, then type in a new name. If unchecked, the default caption will be defined by the selected [parameter](#). Normally the caption will be assigned the parameter name only. However, if you hold the SHIFT key down while selecting a parameter, the caption will include the plugin name as well as the parameter name. If you also hold the ALT key down, the caption will include the plugin name only.

Learn – Click on this button to have Gig Performer learn the physical controller you want to associate with the 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.

Clear – Click on this to clear the current controller assignment.

Jump/Catch – This dropdown box 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 immediately jump to the position of the controller. When a widget is in Catch mode, the widget will not respond to the associated physical controller until the controller's value matches the current value of the widget, at which point the widget will respond.

Treat momentary as latching – (Displayed only when editing an on-off widget [i.e., a switch or a button]). When checked, the widget remains in its ON position until you click on it (or touch the associated physical controller), and vice versa. 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.)

Gig Performer User Guide v 1.5.2

Parameter range and control – Each widget has a range of 0 to 100. *Note that this is a percentage, not an absolute value.* For example, if the plugin parameter it is controlling has an absolute range from 0.0 to 1.0, then when the widget is at its “100” setting it is actually changing that parameter to 1.0 (i.e., 100% of 1.0).

By entering **Min** and **Max** values other than 0 and 100, you can scale the range of the widget movement. For example, if you set Min to 10 and Max to 20, then the associated plugin parameter will change from 0.1 to 0.2. Finer resolution can be accomplished by entering decimal point values, in tenths of a decimal. Scaling can be inverted by setting the MIN value to a higher number than the MAX value.

You can also directly enter MIDI data values in the Min and Max fields, 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 50.4 (roughly 50% of the maximum MIDI value of 127). The **Now** field simply indicates the current setting of the widget.

Recall snapshot on load – Gig Performer normally restores all widgets to the position they were last in when a .gig file is saved. Checking this button allows you to take a snapshot of the current position of the selected widget so that it is that position that gets 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 to always start soft. The concept is that, even if you play around with widgets during the course of your live performance, when you next reload a .gig file, your originally stored values will return.

Snap – (*Displayed only when “Recall snapshot on load” [see above] is checked.*) Click on this button to take a snapshot of the current widget position

Also recall on activate – Gig Performer normally restores all widgets to the position they were last in when a rackspace is activated. Checking this button allows you to take a snapshot of the current position of the selected widget so that it is that position that gets loaded instead when the rackspace is again activated. The concept is that, even if you play around with widgets during the course of your live performance, when you next return to a rackspace, your originally stored values will return.

Display range – When checked, the widget shows the Min and Max values as a white shadow around the outside of the knob or slider.

Invert value - When checked, the widget does the opposite of the expected operation; i.e., a switch or button will cause a parameter to go to minimum when in the ON (lit) position, and vice versa.

Gig Performer User Guide v 1.5.2

Widget groups – Each widget can only be associated with a single plugin parameter. However, the 26 Widget Groups provided by Gig Performer enable you to control multiple parameters from any or all of the widgets assigned to that group. Clicking below the words “Widget Group” opens a dropdown menu with the letters A - Z. All widgets with the same letter are grouped together so that moving one widget will cause all the other widgets in that group to move as well, adjusting various plugin parameters simultaneously.

Note that you can use each widget’s Min/Max range controls to specify how much each individual widget affects its associated parameter. You can even make some parameters go in reverse, either by clicking on that widget’s “Invert value” button or by setting the MIN value to a higher number than the MAX value. This makes it possible to implement special effects like crossfades, for example.

OSC name – Allows you to set an OSC name. For more information [click here](#).

Show back plate – (*Displayed only when editing a knob or toggle switch.*) When checked, “back plate” information such as numbers or an “on/off” legend is displayed.

Ignore variations – When checked, the selected widget will be excluded from variation changes and will instead 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, but you want the widget controlling overall volume to remain constant.

Mapping section

Plugin – Click in this box for a dropdown list of plugins currently being used by the rackspace, then select the plugin you want the widget to access.

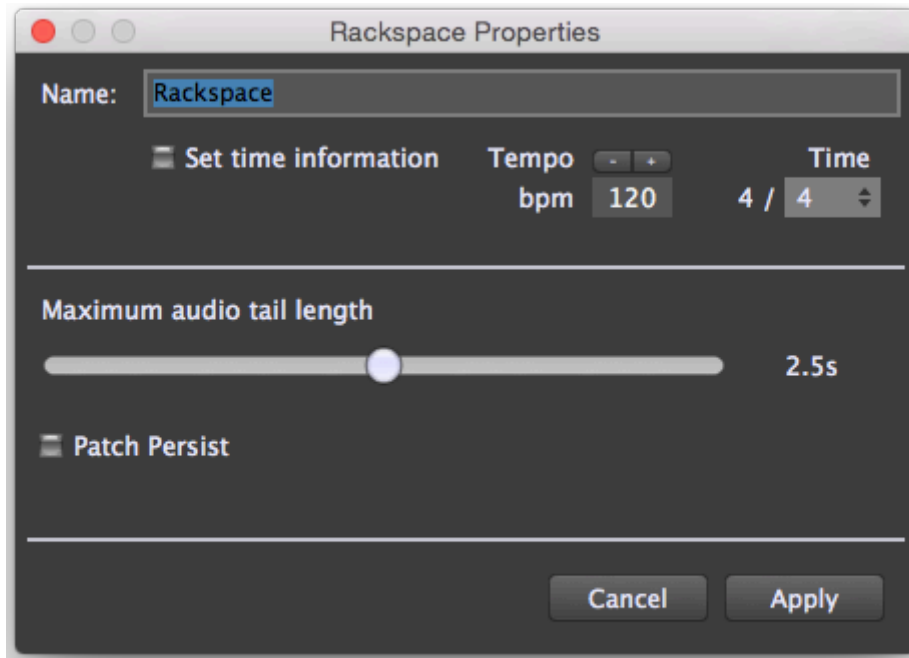
Open plugin – (*Displayed only when a plugin is selected.*) Click on this button to open the editor for the selected plugin.

Parameter – (*Displayed only when a plugin is selected.*) A list of available controllable parameters are displayed here. Select the one you want to have the widget control. Alternatively, you can press the **Learn** button. (See below.)

Learn – (*Displayed only when a plugin is selected.*) Click on this button to have Gig Performer learn the plugin parameter you want to associate with the 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.

Rackspace Properties

Double-clicking on a rackspace name opens a dialog that displays rackspace properties.



Name – To rename the rackspace, type a new name here.

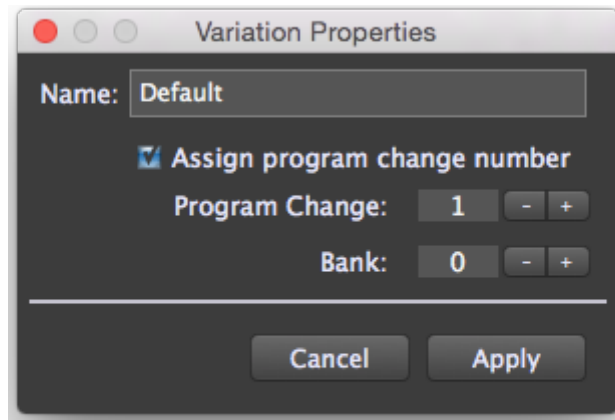
Set time information – Check this box if you want the rackspace to have its own tempo value, then enter the desired bpm (beats per minute) and time signature. If this box is unchecked, the rackspace will use the global tempo information set in Preferences. [Click here](#) for more information.

Tail length – Sets the maximum time before a sound is cut off when you switch rackspaces. Maximum length is 5 seconds. Often known as an audio “tail,” this feature is useful when using effects such as reverb or echo.

Patch Persist – If enabled (checked), whenever you switch to another rackspace, notes that you are holding down (either on your keyboard or via your sustain pedal) will continue sounding until you release the keys or pedal. (This actually 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 disabled (unchecked), whenever you switch from one rackspace to another, the sound is immediately silenced even if you are holding notes down or have pressed the sustain pedal.

Variation Properties

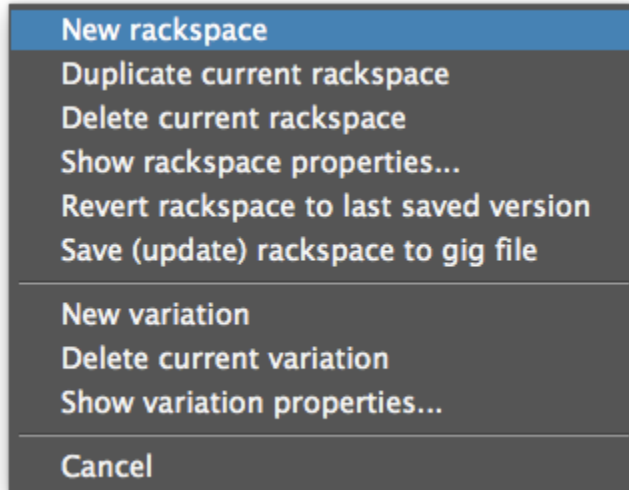
Double-clicking on a variation name opens a dialog that displays variation properties.



Name – To rename the variation, type a new name here.

Assign program change number – When checked, allows you to specify the program change number (and, optionally, the bank change number) used to switch to this variation. If a program change number has been assigned, that number will appear to the left of the variation name and an asterisk will appear to the left of the rackspace name. If unchecked, Gig Performer will assign program change numbers in ascending order; i.e., the first variation in the first rackspace will be called up when program change 1 is received; the first variation in the next rackspace will be called up when program change 2 is received, etc.

... (“More”) Menu



Clicking on the [“Three dots” icon](#) in Performer or Connections View opens the “More” menu:

New rackspace – Creates a new rackspace.

Duplicate current rackspace – Duplicates the current rackspace.

Delete current rackspace – Deletes the current rackspace.

Show rackspace properties... – Opens the [Rackspace Properties](#) dialog.

Revert rackspace to last saved version – Reverts the current rackspace to the last version saved. (**Important:** Click OK in the resultant “Are you sure?” dialog only if you are sure that you want to do this, because all changes made since the file was last saved will be permanently lost. If you want the opportunity to save your work first, click Cancel, then go to File | Save or File | Save As..)

Save (update) rackspace to gig file – Saves just the current rackspace to the currently open .gig file.

New variation – Creates a new variation.

Delete current variation – Deletes the current variation.

Show variation properties... – Opens the [Variation Properties](#) dialog.

Cancel – Exit the “More” menu.

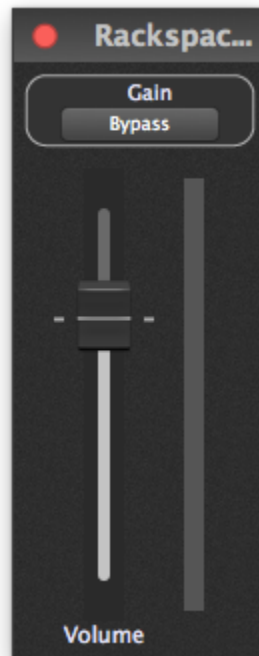
Supplied Plugins

Audio In - Represents the current physical interface used to receive audio.

Audio Out - Represents the current physical interface used to send audio.

Note: These plugins cannot be removed from your rackspace. However, you can specify different audio interfaces for Audio In and Audio Out

Gain Control



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. Setting the Volume slider to 0 causes the plugin to be muted (not bypassed). [Click here](#) for more information.

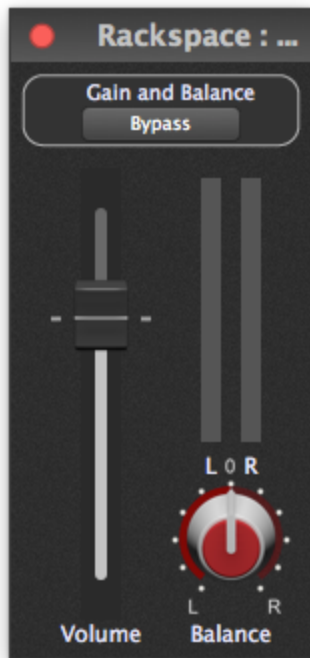
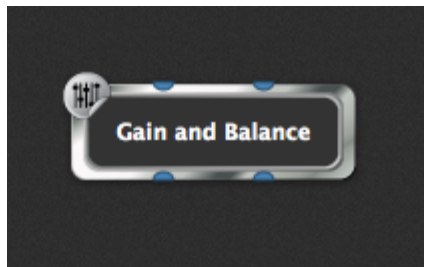
Gain Control widget control

The following parameters for the Gain Control plugin can be widget-controlled:

- Gain
- Balance
- Magnitude

The magnitude parameter is used for level metering. [Click here](#) for more information.

Gain and Balance



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. Setting the Volume slider to 0 causes the plugin to be muted (not bypassed). [Click here](#) for more information.

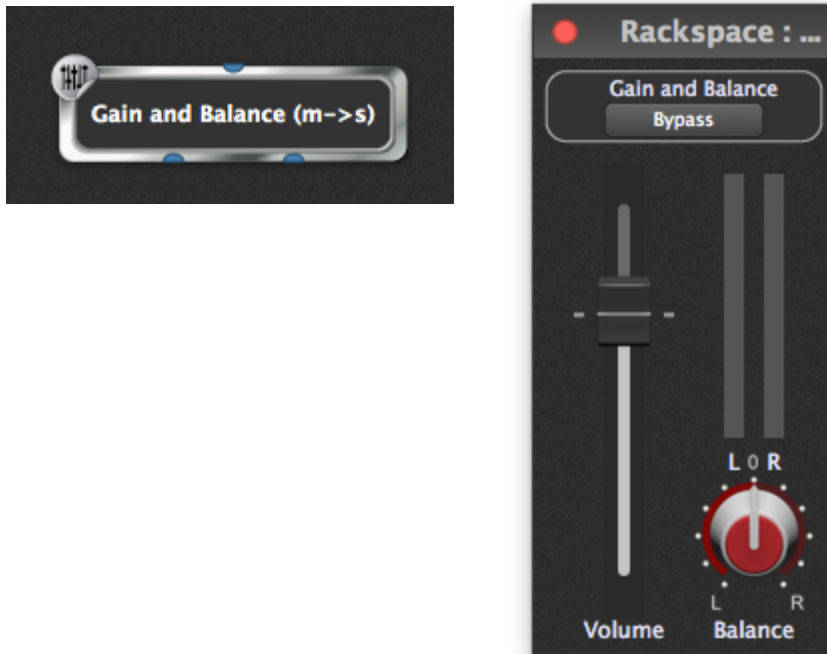
Gain and Balance widget control

The following parameters for the Gain and Balance plugin can be widget-controlled:

- Gain
- Balance
- Channel 1 Magnitude
- Channel 2 Magnitude

Magnitude parameters are used for level metering. [Click here](#) for more information.

Gain and Balance (m->s)



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. Setting the Volume slider to 0 causes the plugin to be muted (not bypassed). [Click here](#) for more information. A balance control enables the incoming mono source to be panned across Left/Right channels.

Gain and Balance (m->s) widget control

The following parameters for the Gain and Balance (m->s) plugin can be widget-controlled:

- Gain
- Balance
- Channel 1 Magnitude
- Channel 2 Magnitude

Magnitude parameters are used for level metering. [Click here](#) for more information.

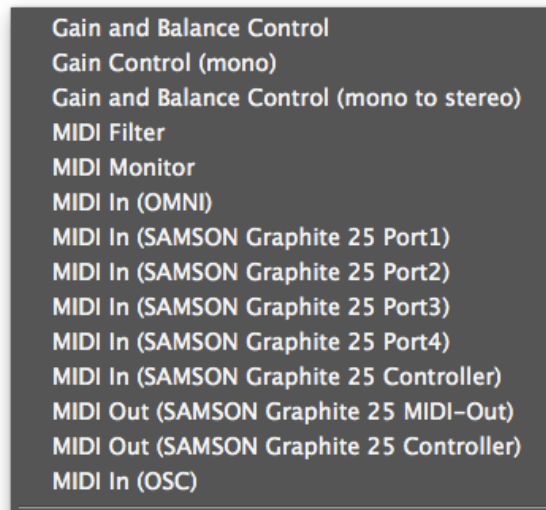
If you connect the audio outputs of multiple plugins to the input of another plugin, the signals will be mixed automatically. To create custom blends, insert one of the Gain plugins in-between, then adjust their volume controls as desired. Gain plugins can also be used to create **reverb sends**, where signal is routed to a reverb (or other type of signal processing plugin) from multiple sources. Simply insert a Gain plugin between the output of each source and the reverb plugin; the volume of each provides independent control over how much signal is sent to the reverb.

MIDI In (OMNI) and MIDI In (OSC)



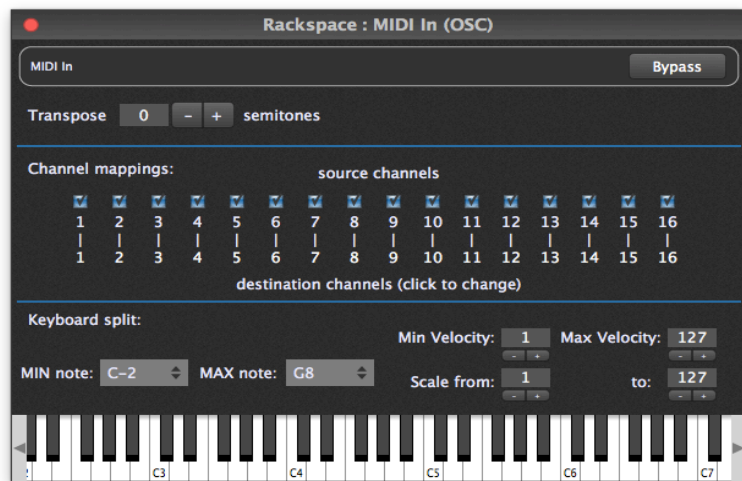
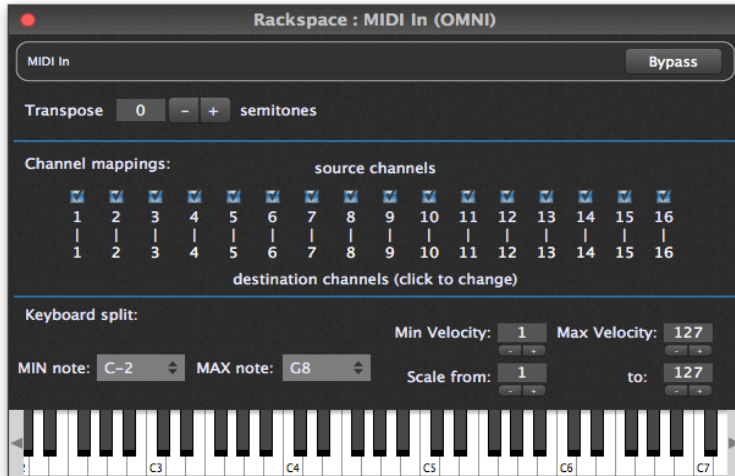
These plugins provide a single MIDI output pin and are used to receive incoming MIDI messages and pass them on to other plugins. They are identical, except in one respect: the MIDI In (OMNI) block receives messages from external devices such as connected MIDI controllers, while the MIDI In (OSC) block receives MIDI messages from OSC sources. (For more information on OSC, [click here.](#))

MIDI In (OMNI) is a generic plugin which “listens” to all incoming MIDI messages coming from all connected controllers, on all channels and on all ports. It is the default MIDI input block in any new rackspace, although it can be [deleted](#) if not needed (for example, if you are using plugins to process incoming audio signal only). If one or more MIDI controllers are connected to your system, Gig Performer automatically generates additional controller-specific MIDI In blocks, as shown below:



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.

Gig Performer User Guide v 1.5.2



The MIDI In plugin editors provide controls that enable the following advanced operations:

Key transposition

Transpose incoming note messages over a range of +/- 48 semitones.

Channel blocking and mapping

Specified MIDI channels can be blocked or rerouted to a different channel.

Keyboard splits

Create keyboard splits by specifying MIN and MAX note range values. Notes below the MIN value and/or above the MAX value are blocked; only notes between those values are allowed through.

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

Velocity scaling

By entering “Scale from” and “Scale to” values, the velocity of all incoming note messages are adjusted so as to reduce their dynamic range (i.e., softer notes will be heard louder, or louder notes will be heard more softly).

Using velocity layering with velocity scaling

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.

In addition, 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 set the Scale From/To values to 1 and 50 respectively. As a result, notes received with a velocity of 80 will produce notes with 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.

Playing plugins without an external MIDI controller

A virtual keyboard is provided to allow sound-generating plugins to be played even when an external MIDI controller is not connected.

Gig Performer User Guide v 1.5.2

MIDI In widget control

The following parameters for MIDI In plugins can be widget-controlled:

- Channel 1 – 16 Output
- MinNote (minimum note)
- MaxNote (maximum note)
- Transpose
- StopAllNotes
- StickyNoteChannel
- StickyNoteVelocity
- Note On C-2 through G8 (MIDI note numbers 0 – 127)
- MinNoteVelocity (minimum note velocity)
- MaxNoteVelocity (maximum note velocity)
- MinScaledNoteVelocity (minimum note velocity)
- MaxScaledNoteVelocity (maximum note velocity)

Channel 1 – 16 Output allows you to change MIDI channel blocking or remapping in real time.

MinNote / MaxNote allow you to alter the MinNote or MaxNote settings in real time.

Transpose allows you to alter the key transposition in real time. **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 icon](#). One usage is to turn off MIDI notes when switching variations. Although normally one wouldn't do this (because the whole point of variations is to allow you to continue holding notes down while changing plugin parameter values), it is useful if you want to use variations to actually change the underlying sound you're creating, in which case any notes from the previous sound need to stop immediately once you change variations. Also, Just associate it with any widget (typically, a button), and it will trigger automatically when you change variations.

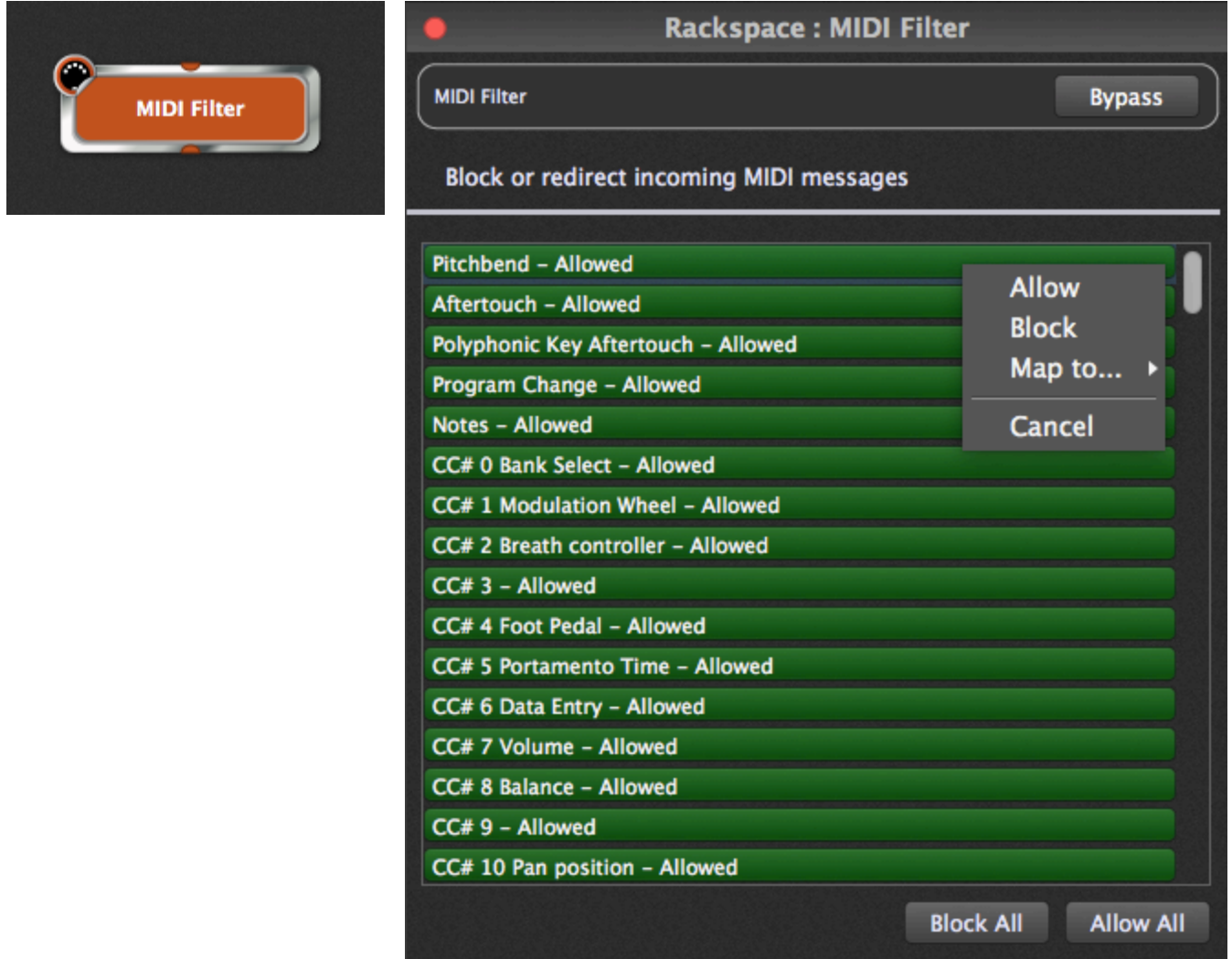
Gig Performer User Guide v 1.5.2

Sticky notes allows 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 (a switch or 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. (You can also add a widget assigned to StickyNoteChannel to select the MIDI channel.) When the button widget is turned off, an appropriate note off event will be sent to the plugin to stop the sound.

MinNoteVelocity / MaxNoteVelocity and **MinScaledNoteVelocity / MaxScaledNoteVelocity** allow you to alter these settings, respectively, in real time.

Note: If you have an audio interface or MIDI controller connected to your system that provides a MIDI output, Gig Performer will generate one or more MIDI Out blocks. [Click here](#) for more information.

MIDI Filter



This plugin provides a single MIDI input pin and a single MIDI output pin. It allows specified MIDI messages to be blocked and/or remapped. (MIDI note messages can be blocked, but not remapped.)

Clicking on any line in the MIDI Filter plugin editor causes a popup menu to appear in which you can make one of three choices:

- Allowed** Messages of this type are passed through.
- Blocked** Messages of this type are blocked.
- > Messages of this type are remapped (converted into messages of another type).

Gig Performer User Guide v 1.5.2

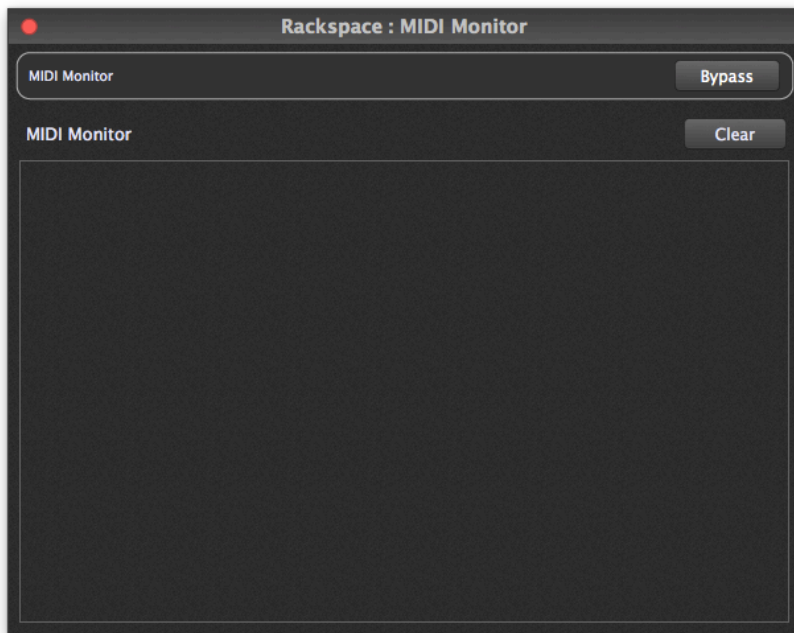
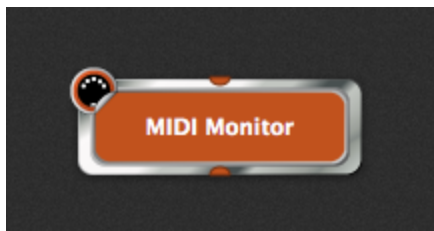
Typically inserted between a MIDI In plugin and a sound-generating plugin, common usages include blocking sustain pedal messages, converting modulation wheel messages into volume messages, and mapping aftertouch messages to mod wheel messages.

There is also an option to block 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](#) to turn all notes off.

MIDI Filter widget control

Every MIDI Filter plugin parameter can be widget-controlled. When using on-off widgets (switches or LED buttons), the selected parameter will be allowed when in the ON position, and blocked when in the OFF 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 allow the selected message, while the maximum setting will block the message.)

MIDI Monitor

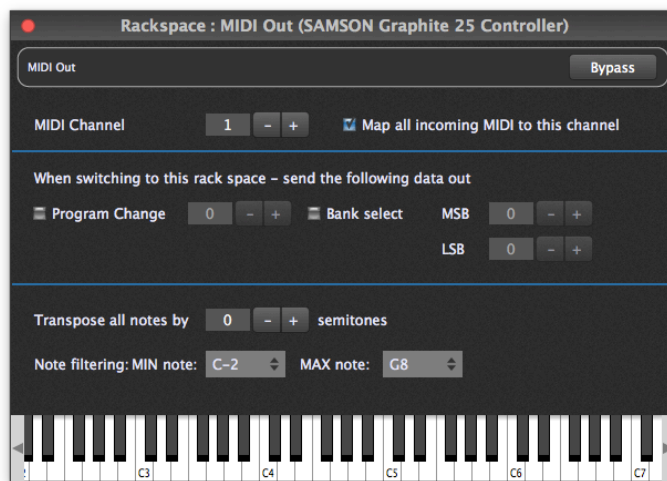


This plugin provides a single MIDI input pin and a single MIDI output pin. It displays MIDI messages from the plugin block to which it is connected and is used primarily for diagnostic purposes.

MIDI Monitor widget control

The MIDI Monitor plugin provides no parameters that can be widget-controlled.

MIDI Out



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 outgoing MIDI messages to external devices such as synthesizers or samplers.

The MIDI Out editor provides controls that enable the following advanced operations:

MIDI Channel sets the channel of the MIDI output.

Map all incoming MIDI to this channel turns off channel mapping so that MIDI data coming from a sequencer retains its channel information. Note that program change transmission (described below) is still associated with the specified MIDI channel even if this box is checked.

Program Change and **Bank Select** allows 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 rackmount synths, drum machines, etc.) load with the correct sounds and are instantly ready for use.

Transpose all notes by allows you to alter the key transposition of the MIDI output.

Note filtering allows you to set up keyboard splits for each MIDI Out block. Notes below the MIN value and/or above the MAX value are blocked; only notes between those values are allowed through.

Gig Performer User Guide v 1.5.2

MIDI Out widget control

The following parameters for the MIDI Out plugin can be widget-controlled:

- ChannelNumber
- MinNote (minimum note)
- MaxNote (maximum note)
- Transpose
- CC #0 - 127

ChannelNumber allows you to alter the MIDI output channel number in real time.

MinNote / MaxNote allow you to alter the MinNote or MaxNote settings in real time.

Transpose allows you to alter the key transposition in real time. **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.

CC allows you to transmit the specified CC (control change) messages as the widget is moved.

Gig Performer User Guide v 1.5.2

Keystroke shortcuts

File management

Mac OSX	Windows	Function
CMD-O	CTRL-O	Open a .gig file
CMD-S	CTRL-S	Save to the current .gig file
SHIFT-CMD- S	CTRL-SHIFT-S	Save the current work to a .gig file with a new name
SHIFT-CMD-I	CTRL-SHIFT-I	Import a single rackspace from a file
SHIFT-CMD-E	CTRL-SHIFT-E	Export the current rackspace to a file

Screen management

Mac OSX	Windows	Function
CMD-T	CTRL-T	Cycle between Performer View and Connections View
CMD-G	CTRL-G	Toggle widget edit view

Rackspace management

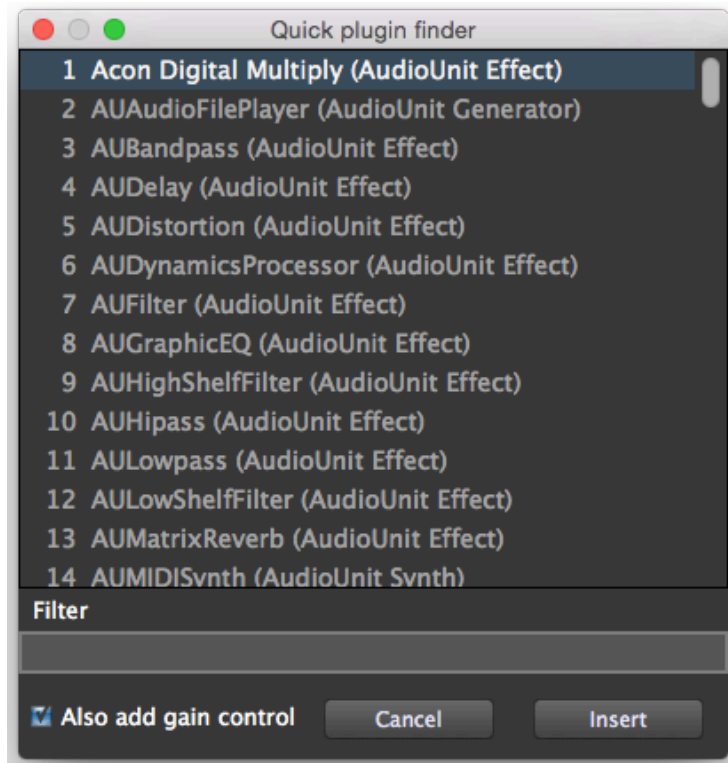
Mac OSX	Windows	Function
SHIFT-CMD- N	CTRL-SHIFT-N	Add a new rackspace
CMD- D	CTRL-D	Duplicate the current rackspace
SHIFT-CMD- BACKSPACE	CTRL-SHIFT-BACKSPACE	Delete the current rackspace
CMD-N	CTRL-N	Add a new variation to the current rackspace
CMD-BACKSPACE	CTRL-BACKSPACE	Delete the current variation
CMD-,	CTRL-,	Open the General preferences dialog

Gig Performer User Guide v 1.5.2

Connections management

Mac OSX	Windows	Function
CMD-P	CTRL-P	Insert a new plugin (Gig Performer will switch to Connections View automatically if you are currently in Performer View.)

Note: The CMD-P /CTRL-P keyboard shortcut provides a more efficient way to create a plugin than right-clicking since it opens the **Quick plugin finder** dialog (shown below), which allows you to type a partial plugin name, with the resultant list restricted to those that match. Checking the “Also add gain control” box in this dialog causes a Gain and Balance plugin to be inserted immediately after the selected plugin.



Gig Performer User Guide v 1.5.2

Additional key shortcuts

Startup

If you hold the SHIFT key down while Gig Performer is starting, it will not reload the previous Gig even if you have the preference “Load last GIG file on startup” checked.

Dragging rackspaces

Holding the SHIFT key while dragging rackspaces prevents the dragged rackspace 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, 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 (Mac OSX) or CTRL (Windows) key for fine control while adjusting a widget.

MIDI In plugin channel toggling

In the MIDI In (OMNI) and MIDI In (OSC) plugin editors, if you hold the SHIFT key down while enabling or disabling individual MIDI channels by clicking on their respective checkboxes, the other checkboxes will be set to their opposite states. This is a convenient way to quickly enable a single channel while blocking all others.

Preferences | Global MIDI LEARN buttons

Holding the SHIFT key while clicking any of the LEARN buttons in the Preferences | Global MIDI dialog clears that assignment.

Widget captions

Normally the widget caption will be assigned the parameter name only. However, if you hold the SHIFT key down while selecting a parameter, the widget caption will include the plugin name as well as the parameter name. If you also hold the ALT key down, the widget caption will include the plugin name only.

TROUBLESHOOTING

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.

Gig Performer isn't finding my plugins

Normally Gig Performer scans only certain "standard" folders for plugins, such as (in Mac OSX 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 (from the [Window menu](#)) and click on the Manage button in the bottom left-hand corner, then select "Set VST folders for scanning...". A dialog will appear allowing you to add or delete folders to the list of those scanned by Gig Performer.

Audio interface does not appear in Connections 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:

- Mac OSX: Go to Utilities -> Audio MIDI Setup and make sure that your interface appears in the list of input and output devices on the left-hand side of the 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 Connections 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:

- Mac OSX: 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: Go to Control Panel -> Hardware and Sound -> Devices and Printers -> Device Manager and 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 to open the Widget Properties inspector and make sure that it is assigned to a plugin and a plugin parameter.

If your question is not answered here, please check our website’s [Knowledge Base](#) and [troubleshooting tips](#). If your problem is still not addressed, feel free to reach out to us through our [support page](#) .

APPENDIX A: OSC

Gig Performer provides extensive support for OSC (Open Sound Control)—a networking protocol that can be used to control one application from another, with a particular focus on apps used for live performance. OSC names can be associated with plugins and widgets, allowing you to transform your tablet or smartphone into a remote display and control surface, with virtual knobs, sliders, and buttons for full wireless control. 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 MaxMSP to create very sophisticated applications that can also work interactively with Gig Performer,

MIDI over OSC

Gig Performer’s supplied [MIDI In \(OSC\) plugin](#) is designed to receive MIDI messages from remote OSC sources. As described in the [OSC Tutorial](#) in this user guide, 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.

The OSC name is used as part of the OSC address target to which messages are sent. For example, if Gig Performer’s MIDI In (OSC) plugin is assigned the OSC name “GP_MIDI_In” ([click here](#) for more information on how to assign OSC names to plugins), then the following OSC messages cause the listed result:

OSC message	Result
/GP_MIDI_In/NoteOn n v c	Sends a MIDI Note On message with the specified note number (n), velocity (v), and channel (c). (Channels are specified from 1 to 16; velocity values of 0 are treated as if they are NoteOff messages.)
/GP_MIDI_In/NoteOff n r c	Sends a MIDI Note Off message with the specified note number (n), release value (r), and channel. (Again, channels are specified from 1 to 16.)
/GP_MIDI_In/Note n v c d	Provides a convenient way to send a MIDI NoteOn message with a duration (d) in milliseconds. The associated NoteOff message is sent automatically after the specified number of milliseconds.

Gig Performer User Guide v 1.5.2

OSC message	Result
<code>/GP_MIDI_In/CC n v c</code>	Sends a MIDI cc (control change) message.
<code>/GP_MIDI_In/PB v c</code>	Sends a MIDI pitch bend message.
<code>/GP_MIDI_In/AT v c</code>	Sends a MIDI aftertouch message.
<code>/GP_MIDI_In/PT n v c</code>	Sends a MIDI polypressure message.
<code>/GP_MIDI_In/Midi <int> [<int> [<int>]]</code>	Designated for special purpose use (typically with algorithmic note generators), this message may have between one and three integer arguments, representing the explicit bytes of the various MIDI channel messages.

Plugin control via OSC

OSC names can also be associated with plugins for the purpose of controlling or querying them. Each plugin in a rackspace must have a unique OSC name. These names can, however, be reused in different rackspaces.

To get a list of all installed plugins that support OSC, send the following OSC message to Gig Performer:

/Rackspace/ListOSCPlugins

In response, a list of OSC messages is sent, consisting of a header, the actual names, and a terminator, as follows:

/PluginListStart
/PluginName <string>
/PluginName <string>
/PluginName <string>
/PluginListEnd

Assuming the plugin name “Piano”, the following OSC messages are available for plugin control:

OSC message	Result
/Piano/GetParamCount	Requests the total number of parameters in the “Piano” plugin.
/Piano/SetParam <int x> <float f>	Sets parameter x to value f (the latter being a value between 0.0 and 1.0). Note: If you actually adjust a parameter on a plugin, this OSC message is also sent out automatically.
/Piano/GetParam <int x>	Explicitly requests the value of parameter with index x. The result will be /Piano/SetParam x f (i.e, you get back the index you requested, along with its value).
/Piano/SetPreset <int x>	Causes a plugin to switch to the specified preset number. Note: Not all plugins respond to this.

Gig Performer User Guide v 1.5.2

Controlling widgets with OSC

To get a list of all OSC widgets in the current rackspace, send the following OSC message to Gig Performer:

/RackSpace/ListOSCWidgets

In response, a list of OSC messages is sent, consisting of a header, the actual widget names, their types (knob, slider, meter, etc), and a terminator, as follows:

/WidgetListStart

/Widget <string> <string>

/Widget <string> <string>

/Widget <string> <string>

/WidgetListEnd

Assuming the widget name “MyDial”, the following OSC messages are available for widget control:

OSC message	Result
/MyDial/SetValue x	Sets the value of the “MyDial” widget. x can be any value between 0.0 and 1.0, with the value of 0.5 setting the widget exactly halfway. Sending this message to a text label widget controls the brightness of the label.
/MyDial/GetValue	Requests the current value of the “MyDial” widget. Note: If you are listening to incoming messages, you will receive messages automatically when a widget is changed and you can opt to cache that information rather than explicitly asking for a value.
/MyDial/SetCaption “abc”	Sets the caption of the “MyDial” widget (including the text of a label) to “abc.”

Gig Performer User Guide v 1.5.2

Using OSC to switch rackspace and variations

OSC messages can be used for remote switching to another rackspace and/or variation within a rackspace. You can specify the rackspace by its index (the first one is always 0) or by name. A second integer parameter is optional but if present defines the variation within the rackspace (starting at 0):

/GigPerformer/SwitchToRack <int> [<int>]

/GigPerformer/SwitchToRack <name> [<int>]

The following OSC messages are available for rackspace and variation switching:

OSC message	Result
/RackSpace/SetVariation <int>	Changes the variation. The first variation is always 0.
/GigPerformer/SwitchToProgramNumber <int>	Changes the rackspace by referring to the associated MIDI program change number.
/GigPerformer/SwitchToProgramNumber <int> <int>	Sends a combined MIDI bank select and program change message.
/GigPerformer/SwitchToProgramNumber <int> <int> <int>	Sends a MIDI bank select message (with individual MSB and LSB values), followed by a MIDI program change message.
/GigPerformer/MoveUp	Moves to the previous variation or rackspace.
/GigPerformer/MoveDown	Moves to the next variation or rackspace.

Gig Performer User Guide v 1.5.2

Gig Performer also sends out OSC information whenever the rackspace or variation is changed. This information can be used to keep the remote control device in sync. In addition to sending the widget names and values (see [Controlling widgets with OSC](#) above), using the same addresses used to control those widgets over OSC, the contents of captions and text labels are also sent out.

OSC message	Result
/RackspaceName <string>	Causes the rackspace / variation name to be sent out.
/Refresh	Causes Gig Performer to resend the entire state again. This can be useful if wireless UDP (User Datagram Protocol) packets are occasionally getting lost.

Whenever Gig Performer sends out the entire state of its widgets, it will first send the message **/ClearAll**. This can be used by the remote OSC app to clear all previous values. (You will need to create a script to do that.)

If a plugin supports presets, you can send an OSC message to set it to the desired preset. For example, assuming a plugin OSC name of "Piano":

OSC message	Result
/Piano/SetPreset x	Sets the "Piano" plugin to preset x. (Presets start at 0.)

Gig Performer User Guide v 1.5.2

Using OSC to set Tempo

OSC messages can be used to set tempo, as follows:

OSC message	Result
/Tap	Causes GigPerformer to measure the rate at which incoming OSC messages arrive, using that information to set the global tempo.
/SetBPM x	Sets the current BPM. If, for example, x = 120, then the tempo is set to 120 BPM.
/AdjustBPM x	Modifies the current BPM up or down by the specified positive or negative increment.

Note: When the BPM is changed, the OSC message **/CurrentBPM <value>** is sent out by Gig Performer.

Using OSC for Tuner control

OSC messages can be used for tuner control, as follows:

OSC message	Result
/TunerView/SetValue 1.0	Sets Gig Performer to Tuner View.
/TunerView/SetValue 0.0	Causes Gig Performer to exit Tuner View.

When you are in Tuner View (and OSC is enabled), Gig Performer sends OSC messages representing the current note name and the offset, as follows:

/TunerNoteName <string>

/TunerCentsOffset/SetValue <float>

The offset value is between 0.0 and 1.0, where 0.5 means you are exactly in tune.

Gig Performer User Guide v 1.5.2

APPENDIX B - 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 <http://www.deskew.com/privacy.html>). 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

Gig Performer User Guide v 1.5.2

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 EXPRESS OR 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 NONINFRINGEMENT. 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,

Gig Performer User Guide v 1.5.2

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.