Gig Performer - Quick Start Guide

Revision 1.5
February 5th, 2017

Welcome to Gig Performer, the cross-platform audio plugin host for Mac and Windows. Gig Performer makes it easy to put your audio plugins together in various configurations, called rackspaces, to create the sound YOU want. Glitch free switching from one rackspace to another, along with low CPU usage and flexible control from remote devices makes it easy to kick your live performance into high gear.

Welcome to Gig Performer

Thank you for your interest in Gig Performer. We are certain that you will find Gig Performer to be a powerful solution to manage your software synths and effects for use in a live performance situation.

Before you start using Gig Performer, please make sure you agree to the Gig Performer license agreement, a copy of which is included in the Appendix

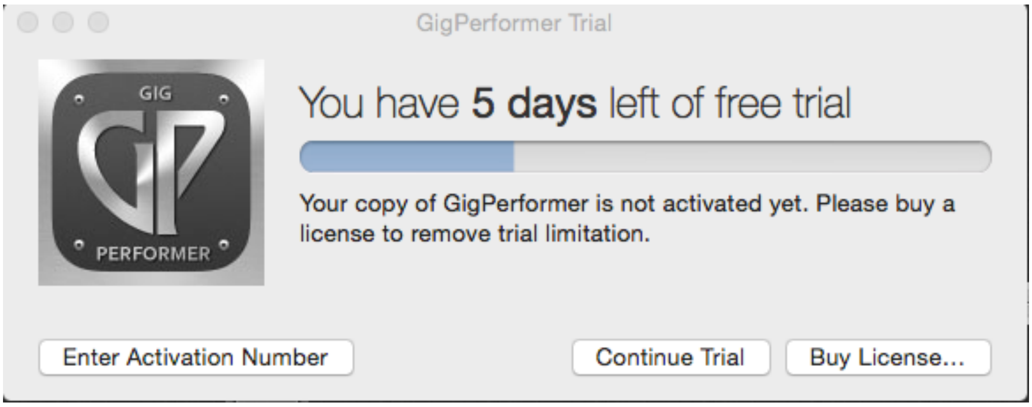
1 Benefits

Whether you are a keyboard player, a guitarist, vocalist, drummer or live sound engineer, Gig Performer has something to offer you.

As a keyboard player, you can use Gig Performer to manage all your sounds using software synths and effects with full real-time control
As a singer, easily add compression, EQ and other desired effects to your vocals.
As a guitarist, you can use Gig Performer to process your guitar sounds any way you want, complete with the ability to switch from one sound to another in real-time without any audio glitching
As a live sound engineer, you can use Gig Performer to intercept audio coming from your mixing system, add desired effects processing and send the results back to your mixing board.

In fact, you can even run multiple Gig Performer instances simultaneously so that multiple members of a band can use Gig Performer independently on the same computer, assuming of course that you have a reasonably modern computer with sufficient speed and RAM.

2 Entering your license code after purchase



You have **5 days** left of free trial

Your copy of GigPerformer is not activated yet. Please buy a license to remove trial limitation.

If you have downloaded the trial, then on or before the time your trial expires you can buy a permanent license for Gig Performer by clicking on the Buy License button which will take you to the online store.

Once you have purchased a license, you will receive a license ID, also known as an activation number. After you have received your license ID (make sure your anti-spam blocker doesn't prevent the confirmation email with your license from being received), simply click on Enter Activation Number and type or paste your license code.

3 Gig Performer Features

Gig Performer has many features to help you create your live show. Among our many features are:

Visual-based plugin connections
Glitchless switching
Patch Persist
Predictive Loading
Tap Tempo
Widget Groups
Multi-instance support
OSC support
External MIDI device support
Customizable front panels
Low CPU usage
Predictive loading

Please see http://www.gigperformer.com/features-and-benefits.html for a full list of current features.

We update Gig Performer often with new features so check back often.

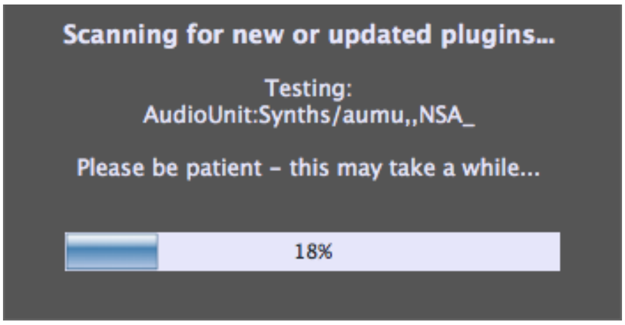
We welcome your suggestions so please feel free to reach out to us through our support page, http://www.gigperformer.com/support.html

Getting started with Gig Performer

Thank you for downloading or purchasing Gig Performer. This quick start manual will help you understand the basic concepts provided by Gig Performer so that you can be up and running as quickly as possible.

Note: not all features are present in all versions of Gig Performer.

1 Plugin scanner



Scanning for new or updated plugins...

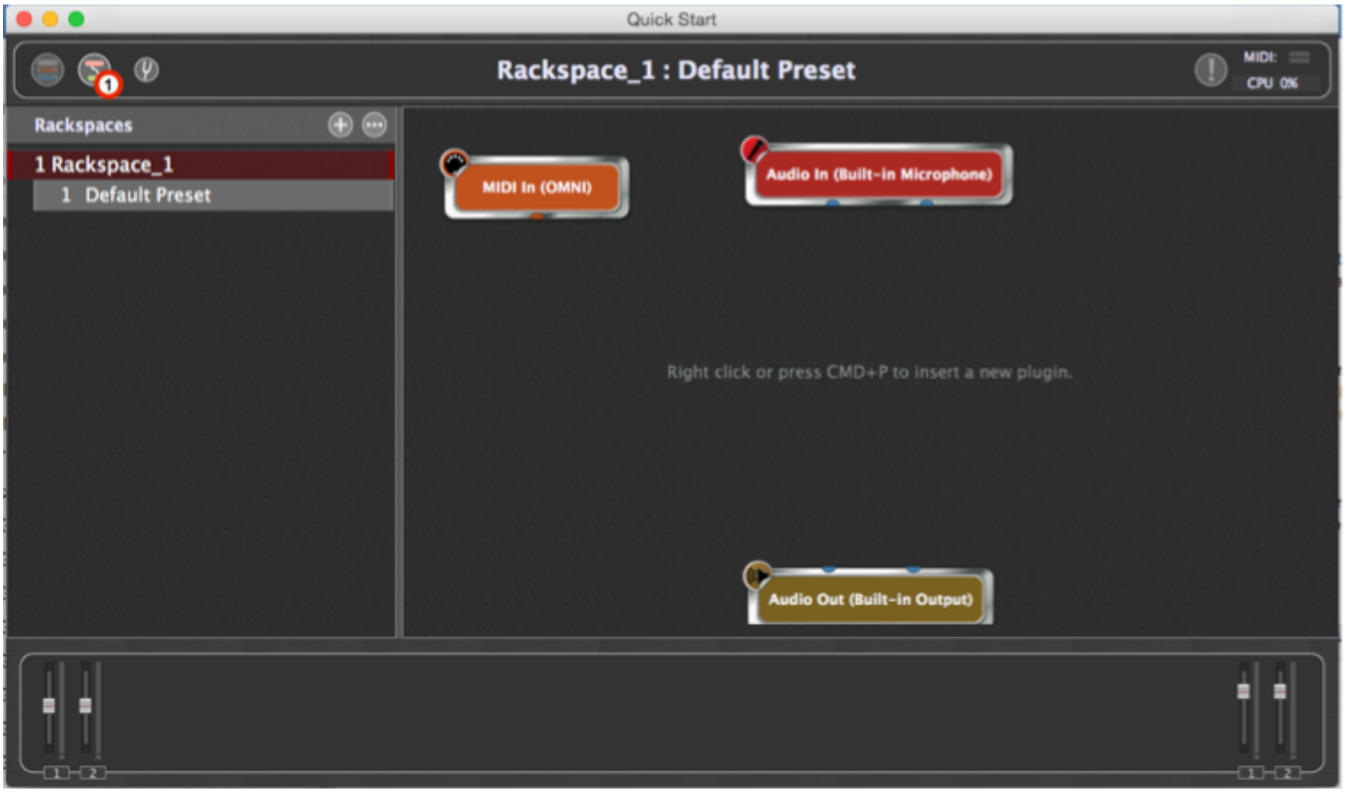
Testing:
AudioUnit:Synths/aumu,,NSA_

Please be patient - this may take a while...

18%

The first time you run Gig Performer, it will scan your system looking for plugins. On MS Windows it will examine all VSTs and on OS X it will examine both VSTs and Audio Units (AUs). This process can take some time but it is only necessary to do this step whenever new plugins are installed and so on subsequent uses, this step is almost instant (and you can disable that check until the next time you install a new or updated plugin)

2 The connections view

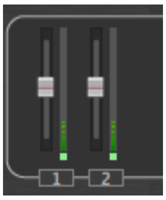

Quick Start
1
Rackspace_1 : Default Preset
MIDI:
CPU 0%
Rackspaces
1 Rackspace_1
1 Default Preset
MIDI In (OMNI)
Audio In (Built-in Microphone)
Audio Out (Built-in Output)
1
2
1
2

Right click or press CMD+P to insert a new plugin.

Once the plugin scanner has completed you will have immediate access to the connection view. If you don't see this view, press the Connections icon (1) This view is where you can insert your plugins, connect them to other plugins and edit plugin parameters directly. The red block at the top represents the input portion of your audio interface. The brown block at the bottom represents the output portion of your audio interface.If you have more than one audio interface, you can change it from the preferences menu.

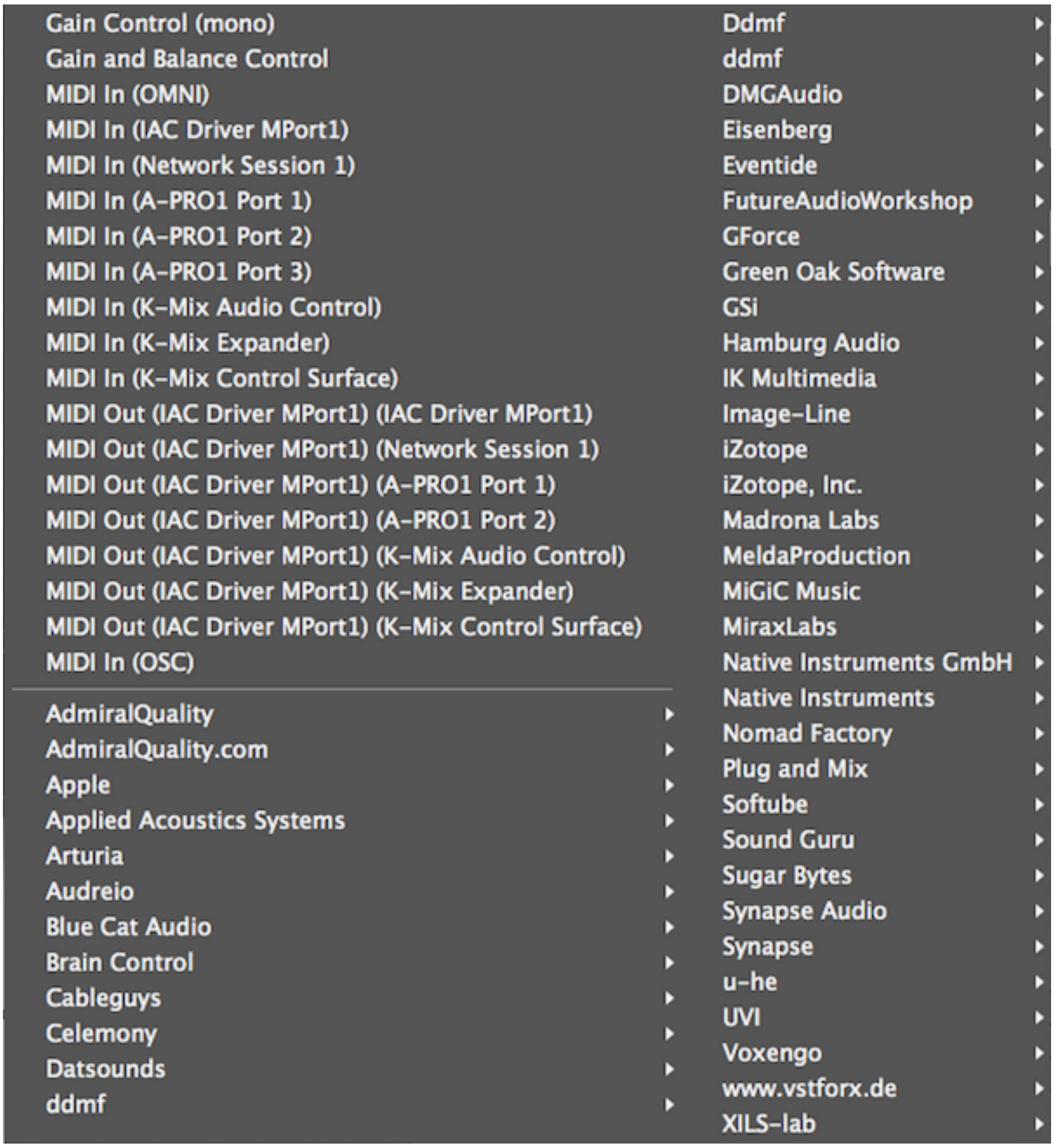
Note that you can use different audio interfaces for input and output if you wish.

3 Audio input monitoring



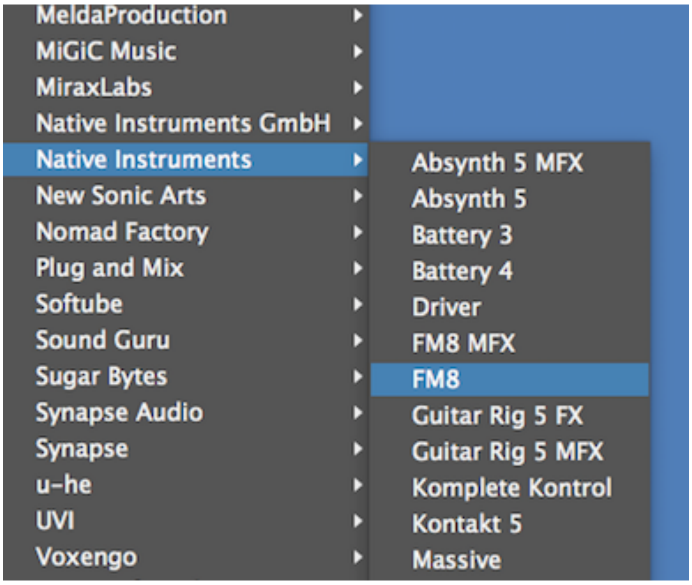
Incidentally, if you see some LEDs flashing on the bottom left of the Gig Performer window, it most likely means that your default audio input is the microphone and Gig Performer is picking up sounds. The vertical faders display the volume of the input while the LEDs at the bottom flash if there is ANY input

4 See all available plugins



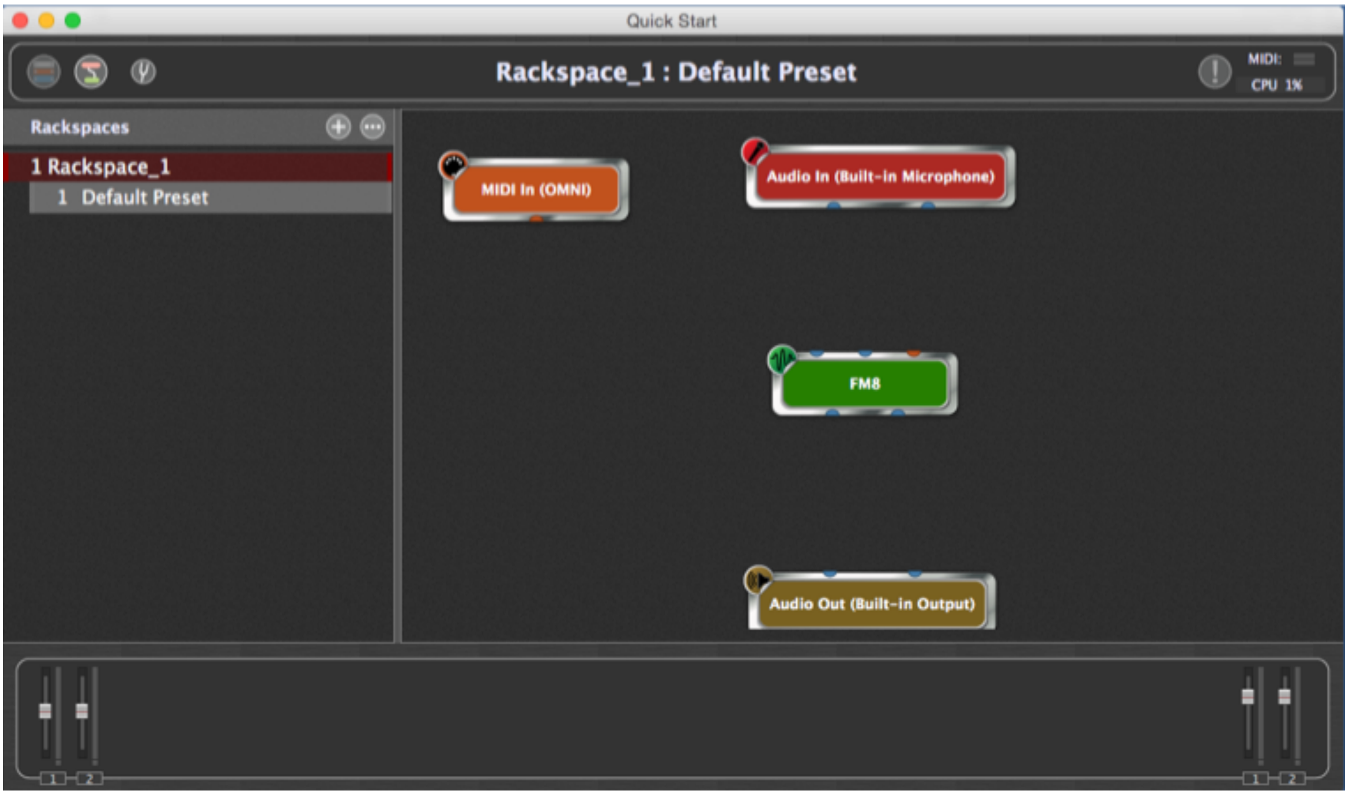
Right-click (or two-finger tap if you're using an Apple trackpad) to bring up the plugin selector menu from where you can choose a plugin to insert.
Important: Gig Performer itself has only a few plugins for MIDI support and an audio gain control (mono and stereo versions). Your own plugins will appear under the line separator (1) categorized by the plugin manufacturer. The example here is from a system

5 Selecting a plugin



Here I have clicked on the FM8 plugin from Native Instruments.

6 Creating a plugin

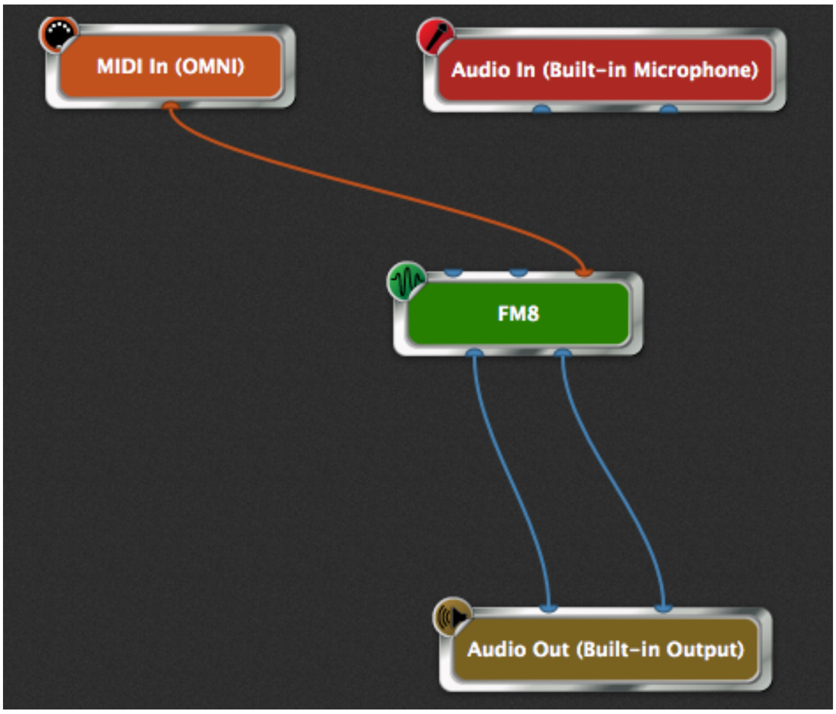

Quick Start
Rackspace_1 : Default Preset
MIDI:
CPU 1%
Rackspaces
1 Rackspace_1
1 Default Preset
MIDI In (OMNI)
Audio In (Built-in Microphone)
FM8
Audio Out (Built-in Output)
1
2
1
2

After a short interval, the new plugin will appear as a green block in the connection view. By default, the name of the plugin is displayed although you can change the name if you want to see a name that represents the actual sound you are using. As we will see later when we learn about associating knobs and sliders with plugin parameters, changing the name of the plugin is very useful to distinguish multiple copies of the same plugin that are being used for different purposes.

The plugin block has several pins both above and below it. These are used to connect this plugin to other plugins. The blue pins are for audio signals and the orange pin is for MIDI signals. In the case of the FM8 plugin, there are two audio outputs at the bottom to send out the audio produced by the plugin. There are two audio inputs at the top as well. This means that the plugin is able to accept audio from somewhere else, typically because it is able to do effects processing. The orange pin on the top is used to receive MIDI from a keyboard or other control surface. Keyboard players will want this immediately while guitarists or singers might not have any need for it at all.

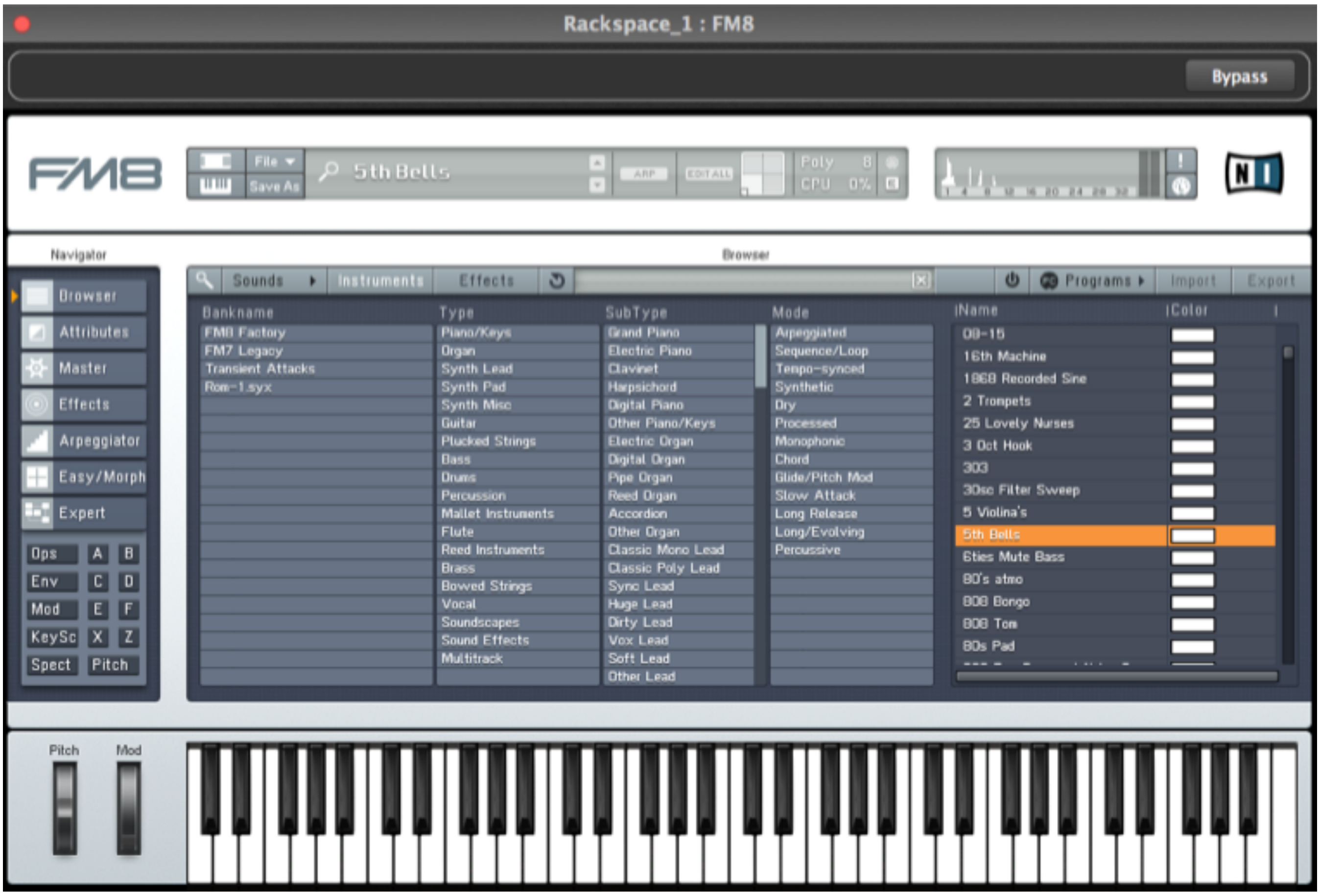
By the way, you can move a plugin block by clicking on it and dragging it where you wish

7 Connect everything



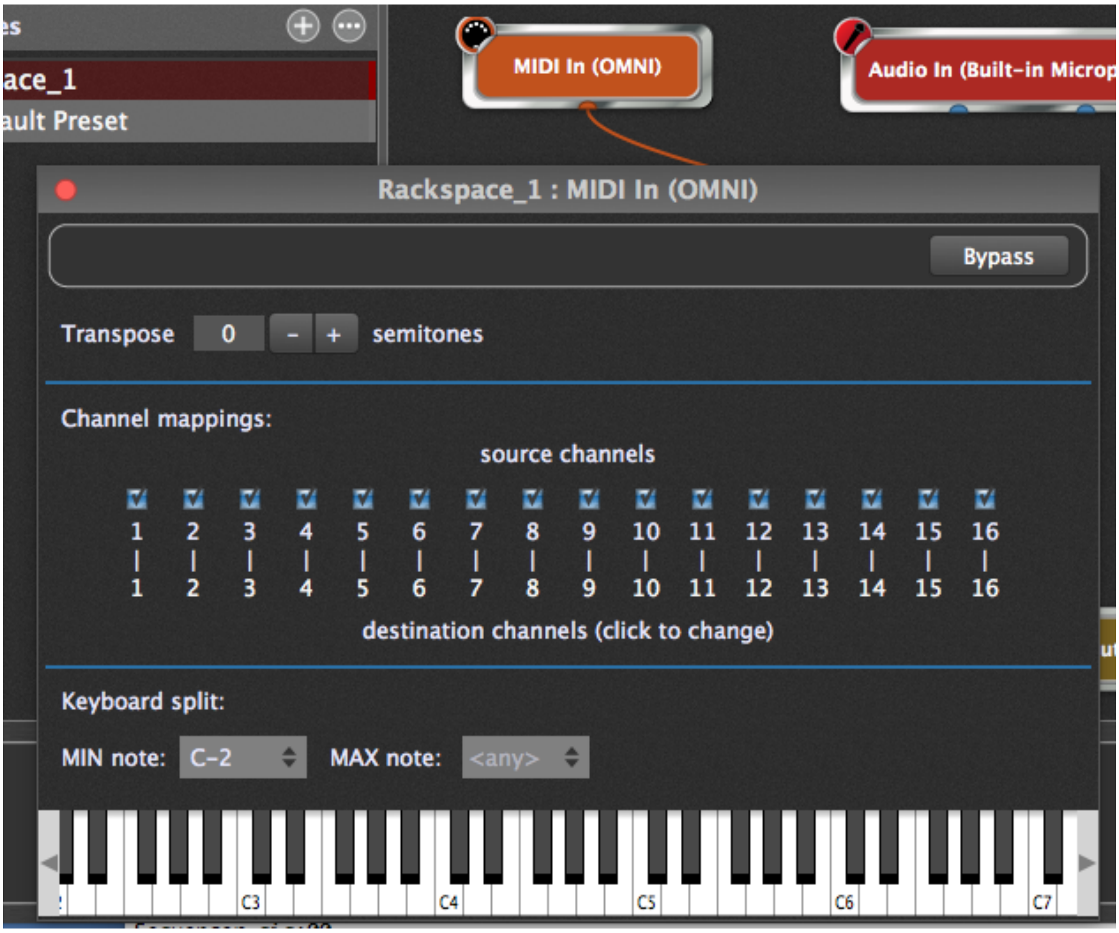
(From now on, we're not going to include images of the entire Gig Performer window, just the section of interest). In this picture we have connected the two outputs of the FM8 synth to the two inputs of the audio output device. We have also connected the output of the MIDI In (OMNI) to the MIDI input pin of the FM8 synth plugin.
To make these connections, simply click on a pin and, while holding down the mouse button (or keeping your finger on the trackpad), drag the cursor over the desired pin in another block and then release (unclick!) the mouse.

8 Select a sound on your plugin



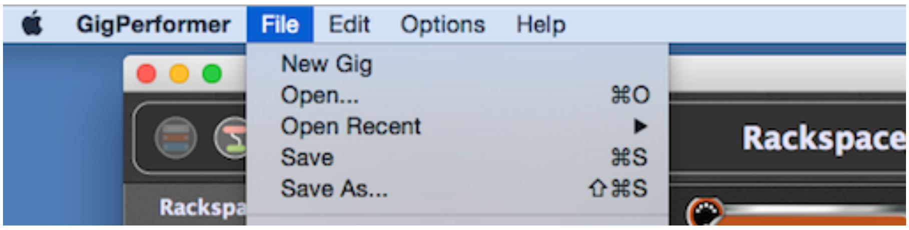
Double-click on the FM8 (or whatever synth plugin you created) to open its editor. Select your desired sound. Of course the steps to do this will depend on the actual plugin you are using. If your plugin editor has a virtual keyboard, clicking on keys should now cause sound to be generated. If you have a MIDI keyboard connected to your computer, you should also be able to play sounds from it directly. You can close the synth plugin editor window after you have finished selecting or editing your desired sound.

9 But I don't have a MIDI keyboard



If you do not have a MIDI keyboard connected to your computer, you can double-click on the "MIDI In (OMNI) block to open its editor where you will see a virtual keyboard. You can click on this keyboard to create MIDI notes that will be sent to all plugins that are connected to the MIDI In (OMNI) block.

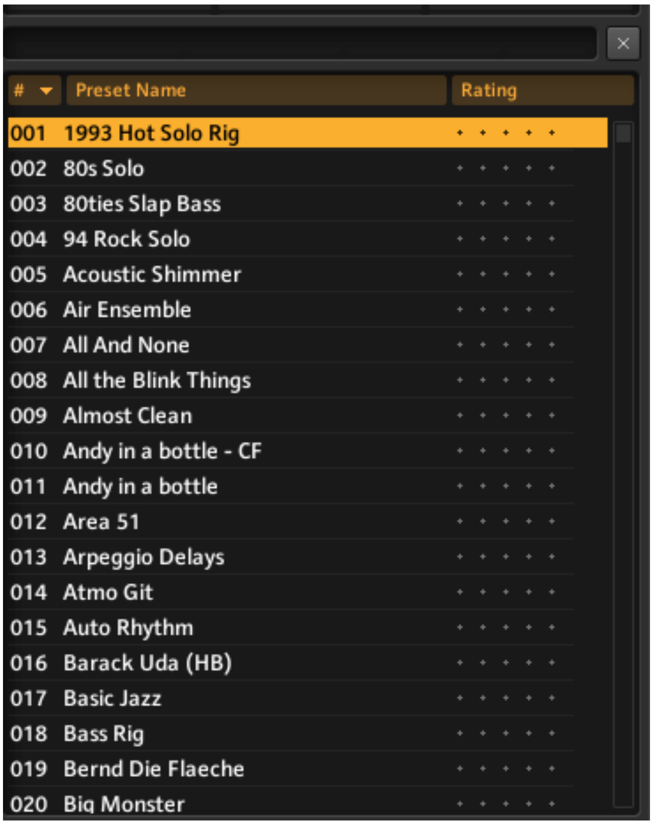
10 Saving your gig



You can save your new gig by clicking File | Save As... and choosing a desired filename. The saved file, with the extension .gig, contains the complete state of your plugins and connections. In other words, if you tweak the parameters of a plugins and save the gig, the next time you load that gig, all those settings will be reloaded.

Adding more plugins

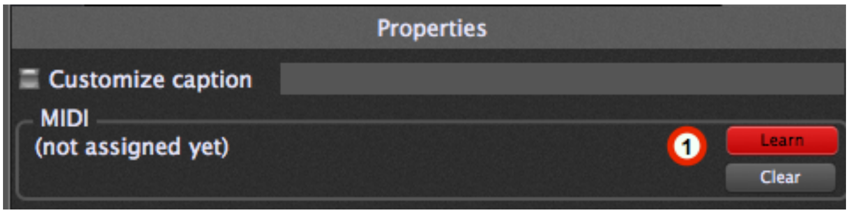
1



#	Preset Name	Rating
001	1993 Hot Solo Rig	· · · · ·
002	80s Solo	· · · · ·
003	80ties Slap Bass	· · · · ·
004	94 Rock Solo	· · · · ·
005	Acoustic Shimmer	· · · · ·
006	Air Ensemble	· · · · ·
007	All And None	· · · · ·
008	All the Blink Things	· · · · ·
009	Almost Clean	· · · · ·
010	Andy in a bottle - CF	· · · · ·
011	Andy in a bottle	· · · · ·
012	Area 51	· · · · ·
013	Arpeggio Delays	· · · · ·
014	Atmo Git	· · · · ·
015	Auto Rhythm	· · · · ·
016	Barack Uda (HB)	· · · · ·
017	Basic Jazz	· · · · ·
018	Bass Rig	· · · · ·
019	Bernd Die Flaeche	· · · · ·
020	Big Monster	· · · · ·

Right click on some empty space in the connections view to popup the plugin selector menu again. Click on the Gain and Balance Control menu item.

2



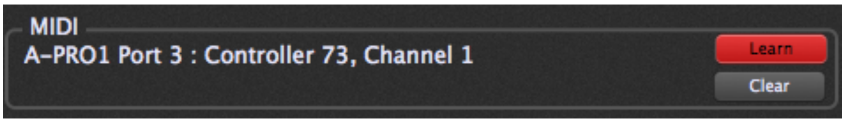
Properties

□ Customize caption

MIDI
(not assigned yet)

A new plugin block will appear, called Gain and Balance (we will change that name later). At this point that plugin block has no connections and hence no impact on the results.

3



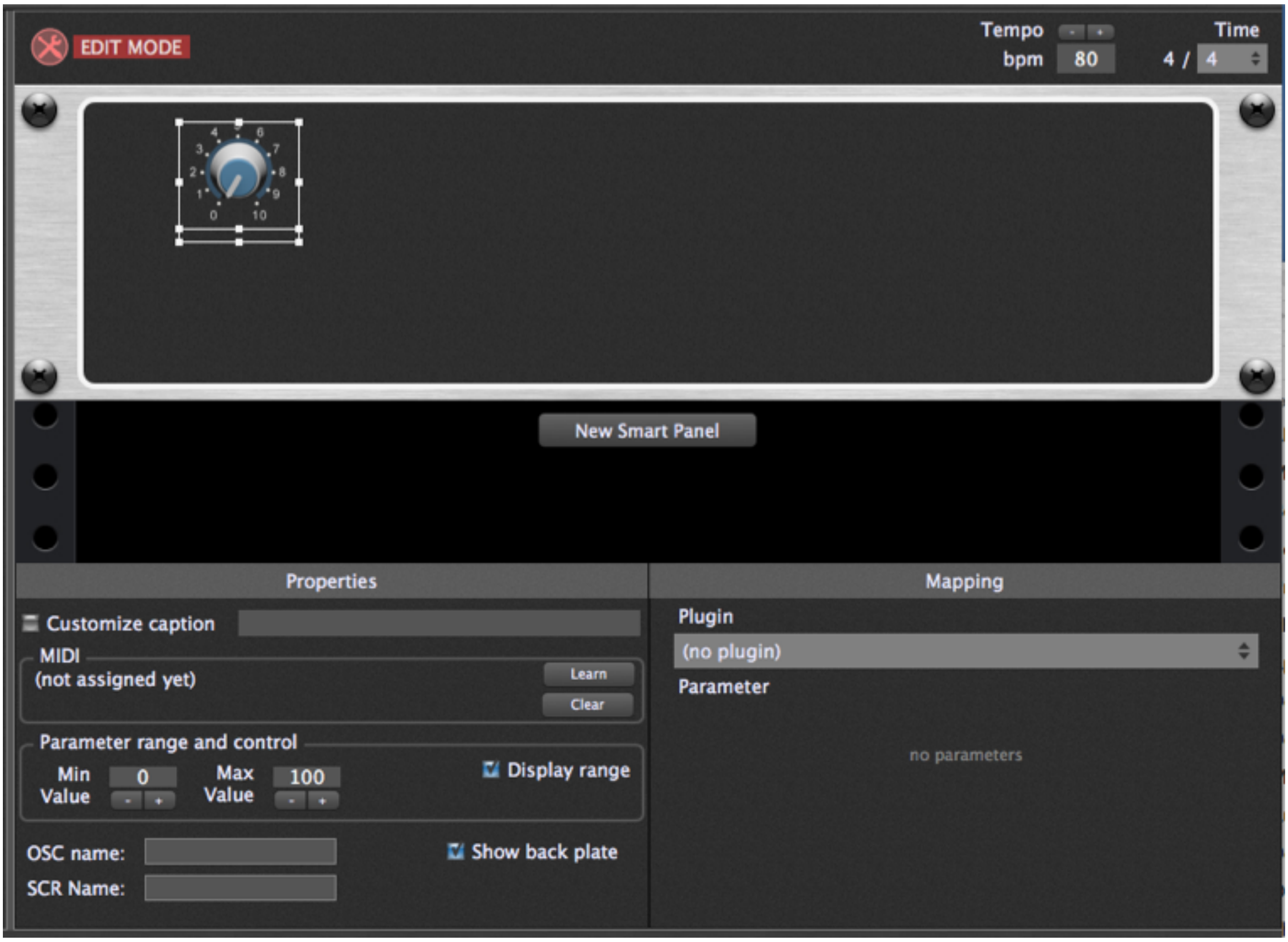
MIDI
A-PRO1 Port 3 : Controller 73, Channel 1

As with any plugin block, if you double-click on the Gain and Volume plugin block, its editor window will open up. The Gain and Volume editor contains a Volume level with level meters to its right and a stereo Balance knob at the bottom right.

The Bypass button, present in all plugin editor windows is a Gig Performer button that allows plugins to be, well, bypassed. For the Gain and Volume plugin, that means that no volume adjustments will be applied to any incoming sound.

Now lets connect the Gain and Volume plugin.

4



Tempo bpm 80

Time 4 / 4

□ Customize caption

MIDI (not assigned yet)

Parameter range and control

Min Value 0

Max Value 100

☑ Display range

OSC name:

SCR Name:

☑ Show back plate

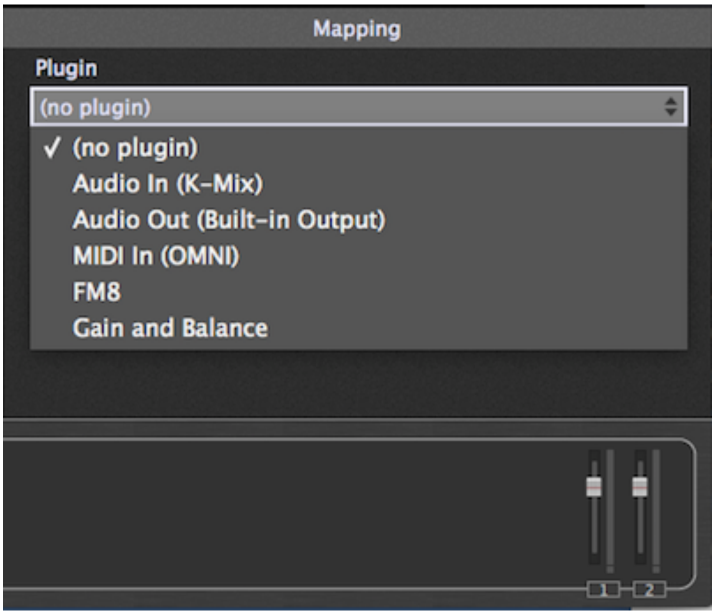
Plugin (no plugin)

Parameter

We have several ways to modify connections. We can
a) Remove an existing connection and then create a new one
b) We can simply click near one end of a connection and drag it to a new pin (or to empty space to remove it)

Right click anywhere on the connection between the FM8 plugin block and the Audio Out plugin block to bring up a menu and click on Remove this connection (1). The connection will disappear. Do the same thing for the second connection.

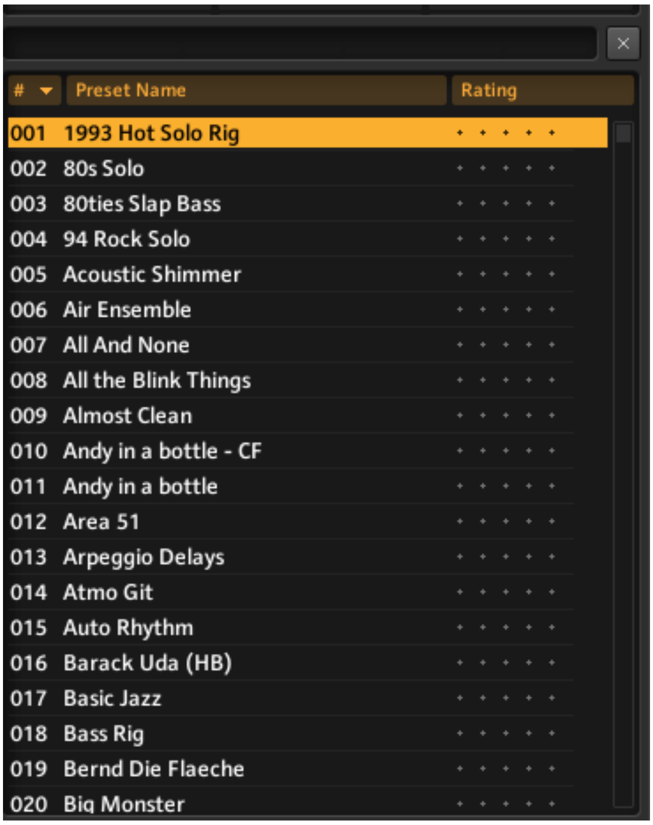
5



Now using the same steps as you did in "Getting started", step 7, connect the synth plugin to the Gain and Balance plugin and then the Gain and Balance plugin to your Audio Out plugin. Feel free to drag the Gain and Balance plugin to make it line up below the FM8 synth if you wish. Moving plugin blocks or their relative positions has no impact on audio.
Try playing the synth plugin again from your keyboard and adjust the gain slider on the Gain and Balance plugin (double-click to open that plugin again if you had closed it)

Adding a widget to control the volume

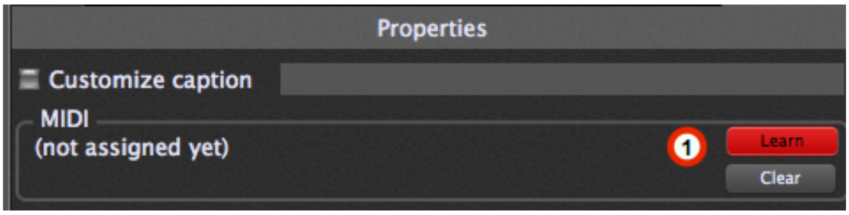
1



#	Preset Name	Rating
001	1993 Hot Solo Rig	• • • • •
002	80s Solo	• • • • •
003	80ties Slap Bass	• • • • •
004	94 Rock Solo	• • • • •
005	Acoustic Shimmer	• • • • •
006	Air Ensemble	• • • • •
007	All And None	• • • • •
008	All the Blink Things	• • • • •
009	Almost Clean	• • • • •
010	Andy in a bottle - CF	• • • • •
011	Andy in a bottle	• • • • •
012	Area 51	• • • • •
013	Arpeggio Delays	• • • • •
014	Atmo Git	• • • • •
015	Auto Rhythm	• • • • •
016	Barack Uda (HB)	• • • • •
017	Basic Jazz	• • • • •
018	Bass Rig	• • • • •
019	Bernd Die Flaeche	• • • • •
020	Big Monster	• • • • •

Click on the top left icon (1) to switch Gig Performer to the performer view.

2



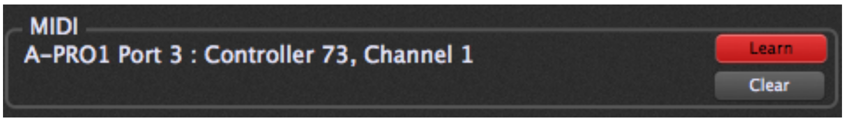
Properties

☐ Customize caption

MIDI
(not assigned yet)

The performer view is where you create widgets (knobs, sliders, buttons and so forth) that are used to control your plugins. As we will see later, you can cause those widgets to be associated with physical controls on your keyboard or other control surface so that you can control your plugins directly from your keyboard or other controller.
When you switch to the performer view for the first time for a given rackspace, you will see a single empty panel (1) into which you can insert your widgets. You can also create extra panels as needed to group your widgets as desired.
To edit the performer view, click on the Widget Layout Designer icon (2)

3

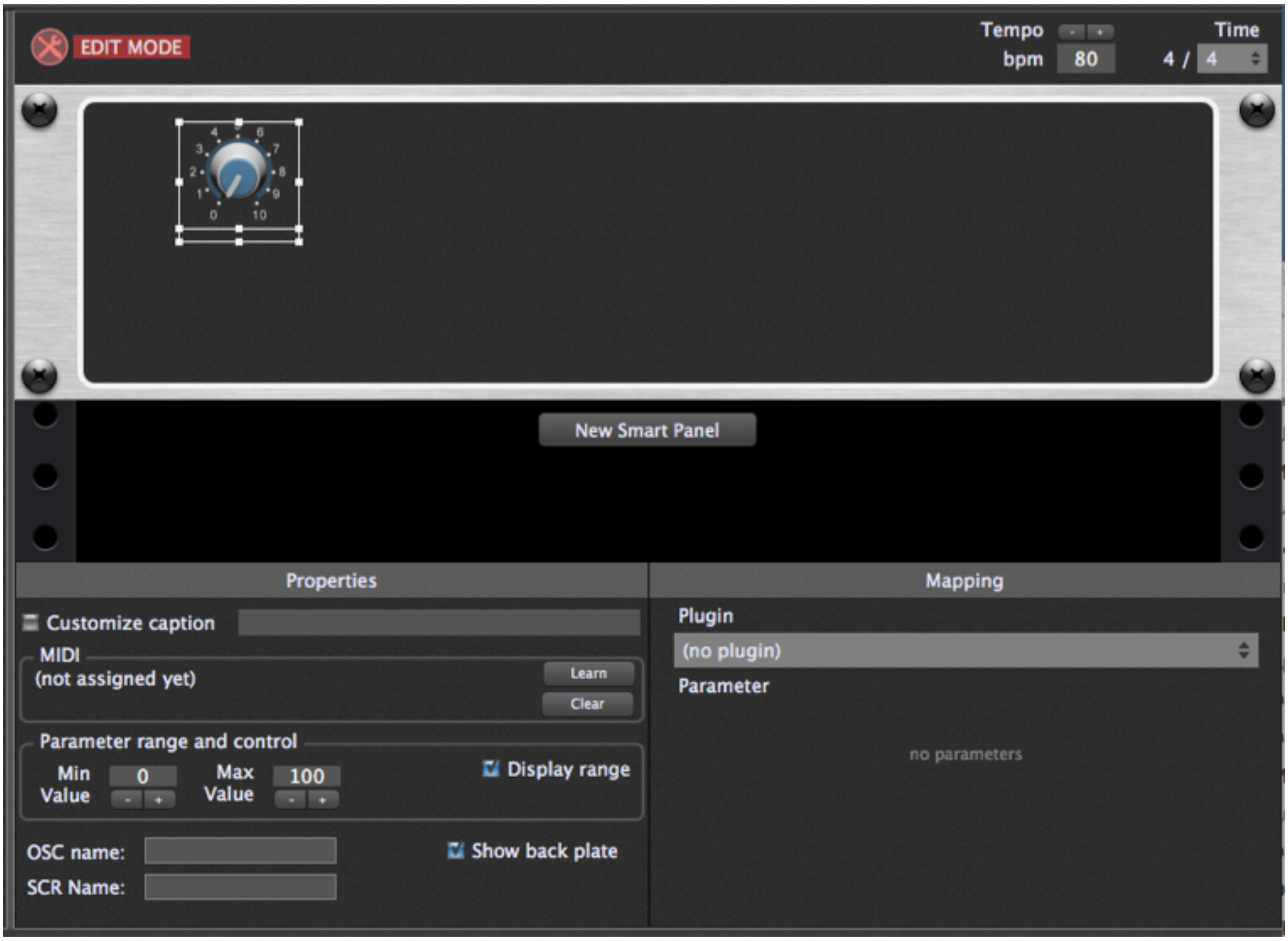


MIDI
A-PRO1 Port 3 : Controller 73, Channel 1

When you switch into the widget layout designer, the list of rackspaces that normally appear on the left

are replaced by a list of available widgets including knobs of different colors, some sliders, meters and toggle widgets. You can place these widgets on a panel by dragging them to the desired location.
Important note - there are no technical differences among the various knobs and sliders, they all do the same thing, that is they generate values when you move them. The differences are solely cosmetic so that you can easily group related items visually or have certain kinds of parameters (for example all volume parameters) stand out clearly.

4



Tempo bpm 80

Time 4 / 4

□ Customize caption

MIDI
(not assigned yet)

Parameter range and control
Min Value 0
Max Value 100
☑ Display range

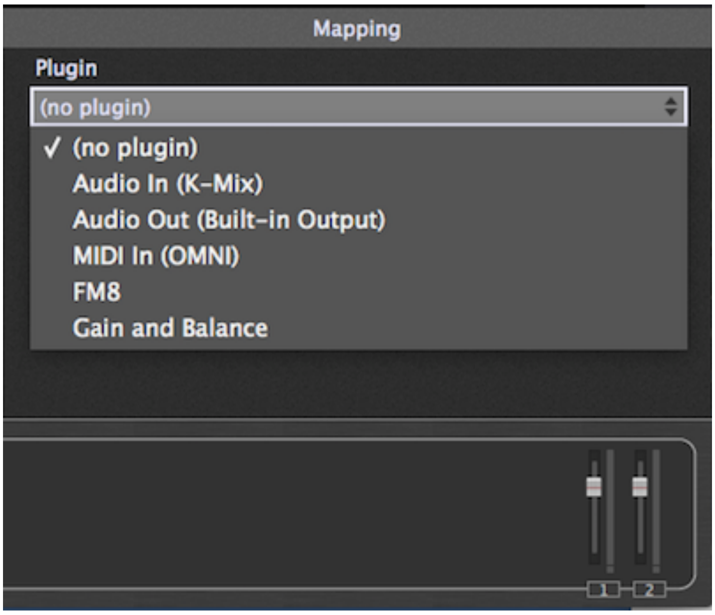
OSC name:
SCR Name:
☑ Show back plate

Plugin
(no plugin)

Parameter
no parameters

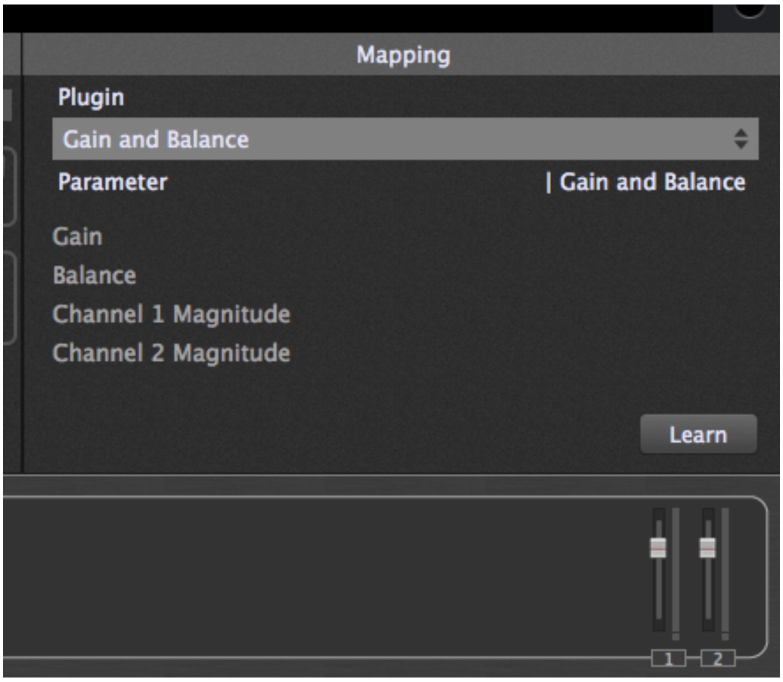
To start, drag a blue knob from the left column somewhere into a panel. It will appear highlighted within a white rectangle and you can resize the knob by dragging any of the corner points. The small section at the bottom will contain the label for this widget.
When a widget is highlighted, its properties will be displayed in an inspector area lower down. The inspector area is broken into two sections, Properties and Mapping. The Properties section allows you to set a label manually as well as associate the widget with a physical knob or slider on a control surface. The Mapping section allows you to associate a widget with a parameter in a plugin. Together, this allows you to use the sliders or knobs on your keyboard or control surface to change parameters in real time.

5



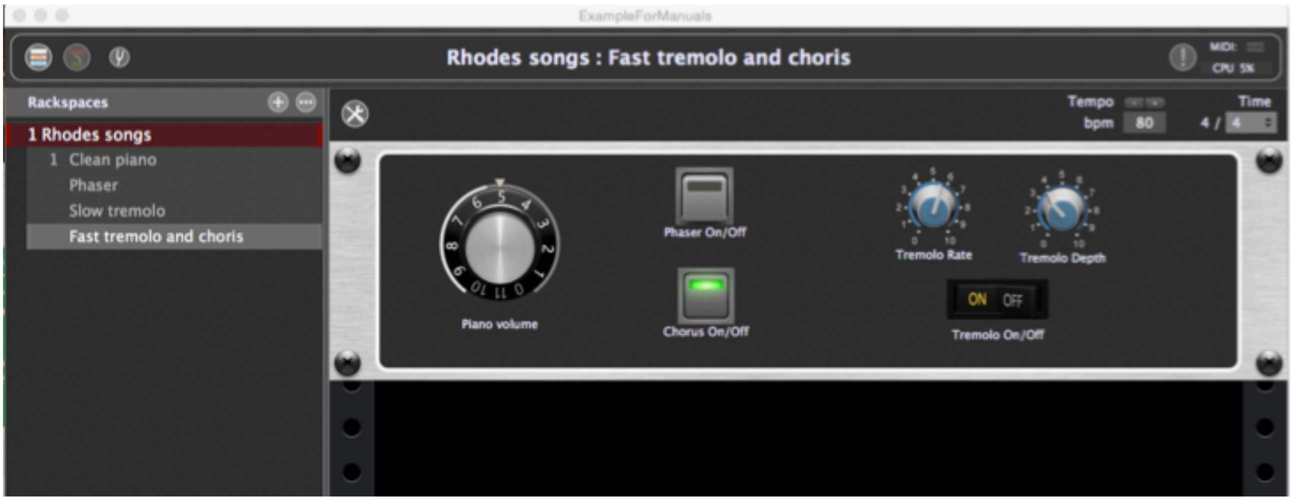
With the knob still selected, click on the (no plugin) item to display a list of plugins currently in use. You will see all the plugins that were created in the connection view including both the automatically created plugins (Audio In, Audio Out and MIDI In) along with the two plugins we created explicitly ourselves.

6



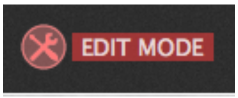
Click on the Gain and Balance plugin and below you will immediately see the available parameters. Click the Gain parameter.

7



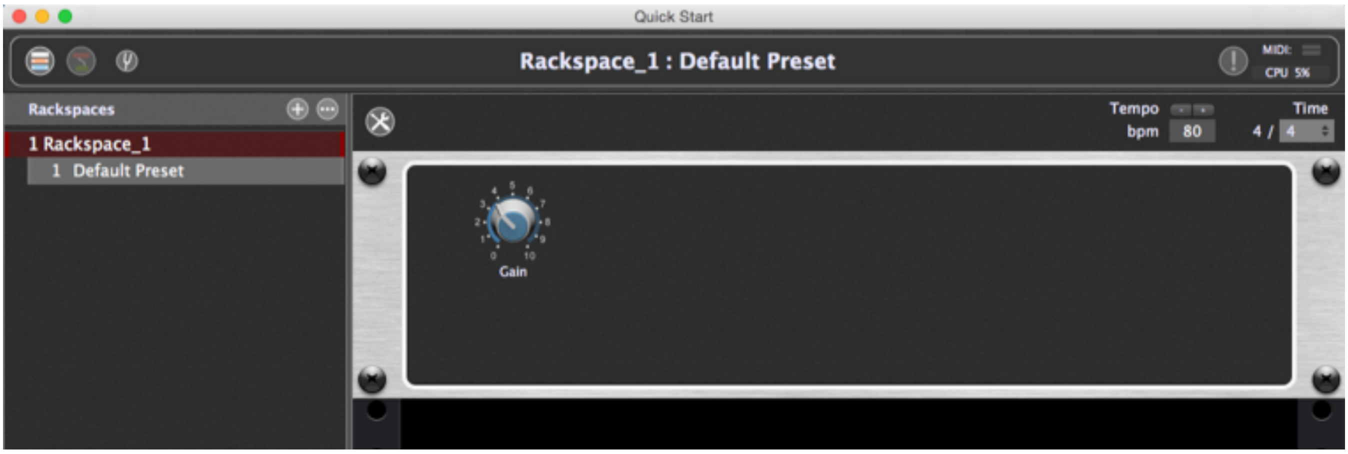
Clicking on Gain (1) causes the default caption (2) to be set to "Gain" and so you see the label "Gain" underneath the knob (3)
The knob value will immediately reflect the current gain of the Gain and Balance plugin.

8



Now click on the Widget Layout Designer icon again to exit edit mode.

9



You now have a single widget and if you turn it while you play your keyboard you will hear the volume change as expected

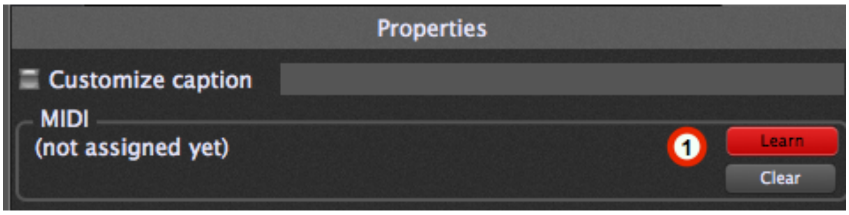
Controlling multiple parmameters with widget groups

1

#	Preset Name	Rating
001	1993 Hot Solo Rig	• • • • • •
002	80s Solo	• • • • • •
003	80ties Slap Bass	• • • • • •
004	94 Rock Solo	• • • • • •
005	Acoustic Shimmer	• • • • • •
006	Air Ensemble	• • • • • •
007	All And None	• • • • • •
008	All the Blink Things	• • • • • •
009	Almost Clean	• • • • • •
010	Andy in a bottle - CF	• • • • • •
011	Andy in a bottle	• • • • • •
012	Area 51	• • • • • •
013	Arpeggio Delays	• • • • • •
014	Atmo Git	• • • • • •
015	Auto Rhythm	• • • • • •
016	Barack Uda (HB)	• • • • • •
017	Basic Jazz	• • • • • •
018	Bass Rig	• • • • • •
019	Bernd Die Flaeche	• • • • • •
020	Big Monster	• • • • • •

Suppose you want to control multiple plugin parameters simultaneously. With Gig Performer you can accomplish this easily using a concept we call widget groups. The idea is simple. You associate individual widgets with parameters as normal. But once you have done this, if you associate those widgets with the same widget group, then moving any widget in the group will cause all the others to move in exactly the same way.

2



Properties

□ Customize caption

MIDI

(not assigned yet)

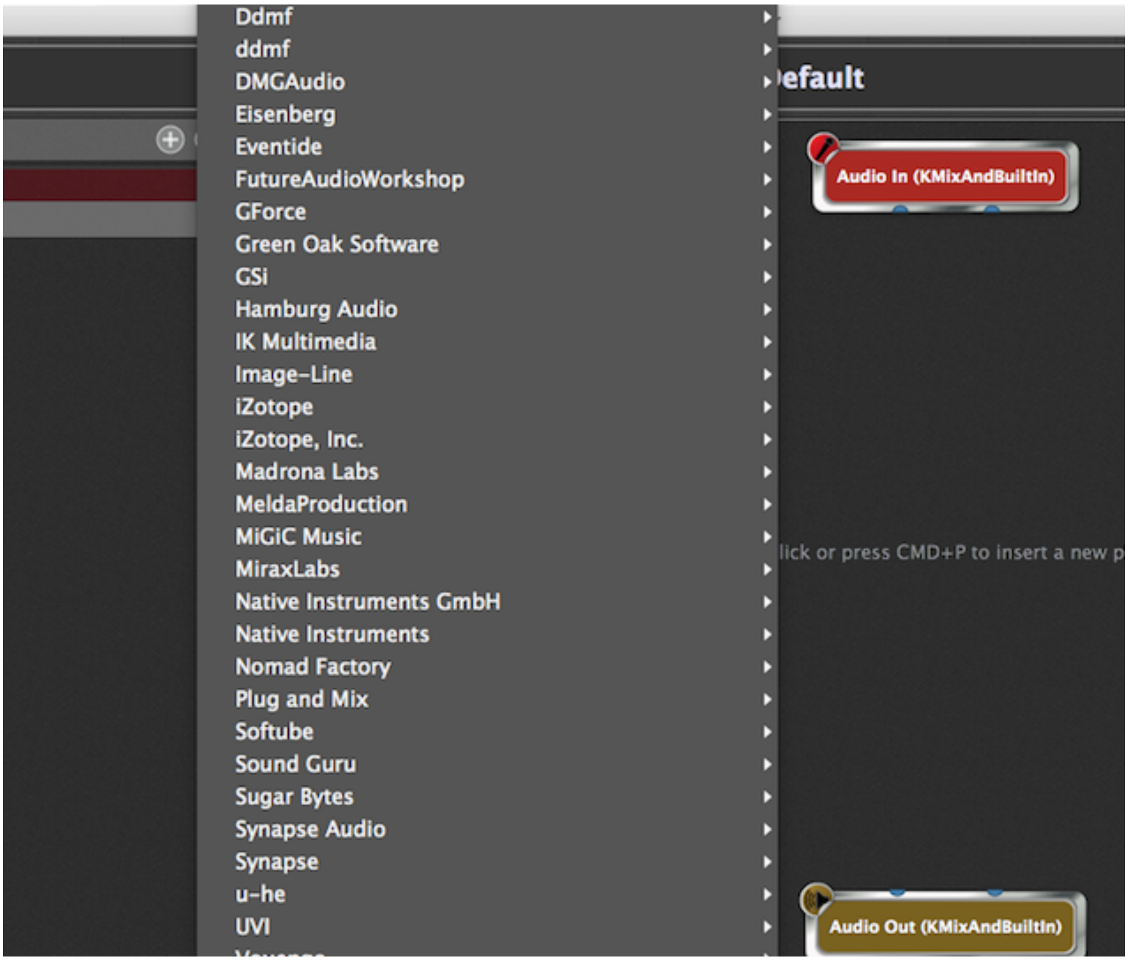
To associate a widget with a widget group, simple select the widget and then choose the desired Widget Group from the popup menu.

Note - even though all widgets in the same widget group will move exactly the same way, widget values are separately scalable so that you can control how much each widget influences the parameter it controls

Working in the connections view

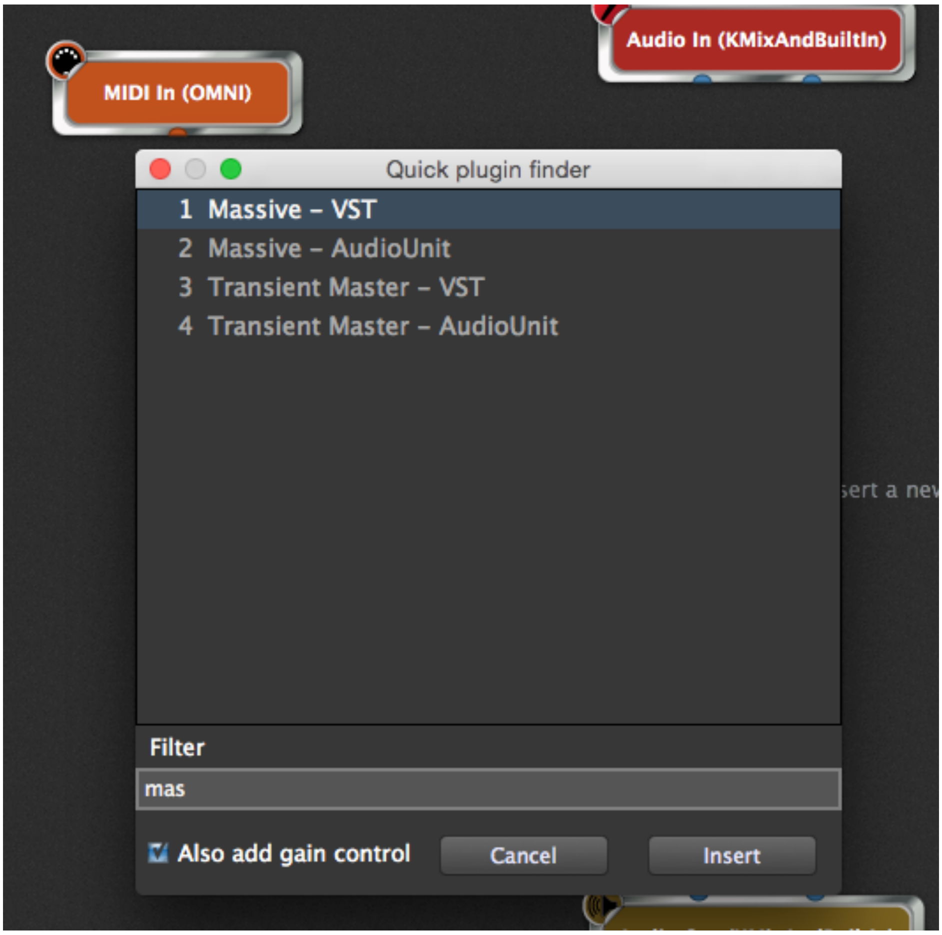
As we saw earlier, the connections view is where you can create plugins and connect them up any way you wish. Some people think of such a structure as an "Algorithm", probably due to that nomenclature being used in the original Yamaha DX7 to define a particular set of connections. However, we prefer terms like "Architecture" or "Structure" as the word "Algorithm" really means a sequence of instructions rather than a static description.
Each rackspace has its own structure consisting of an arbitrary collection of plugins connected any way you wish. Since you can instantly switch from one rackspace to another, this gives you incredible flexibility to create exactly what you need for a particular sound or set of sounds.

1 Inserting a new plugin



Inserting a new plugin is as simple as right-clicking somewhere in the view and selecting the desired plugin from the the menu that pops up.

2 Using the CMD-P shortcut to insert a plugin

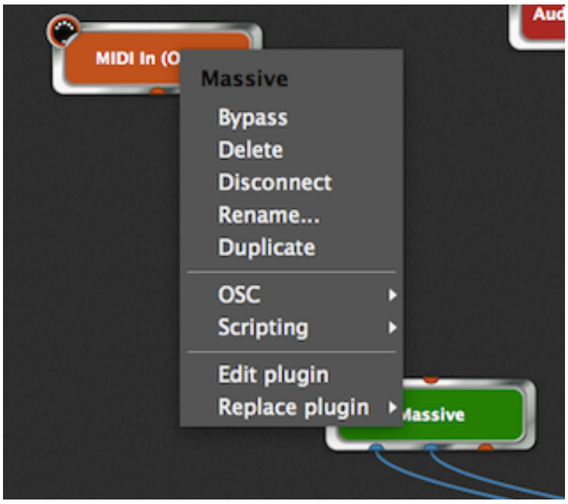


Filter
mas

☑ Also add gain control

The CMD-P keyboard shortcut gives you another way to create a plugin more efficiently using the Quick plugin finder as you can type a partial plugin name and the list of possible plugins will be restricted to those that match what you type.

3 Other plugin operations



If you right-click on an existing plugin block, you will see a menu popup through which you can perform other plugin operations

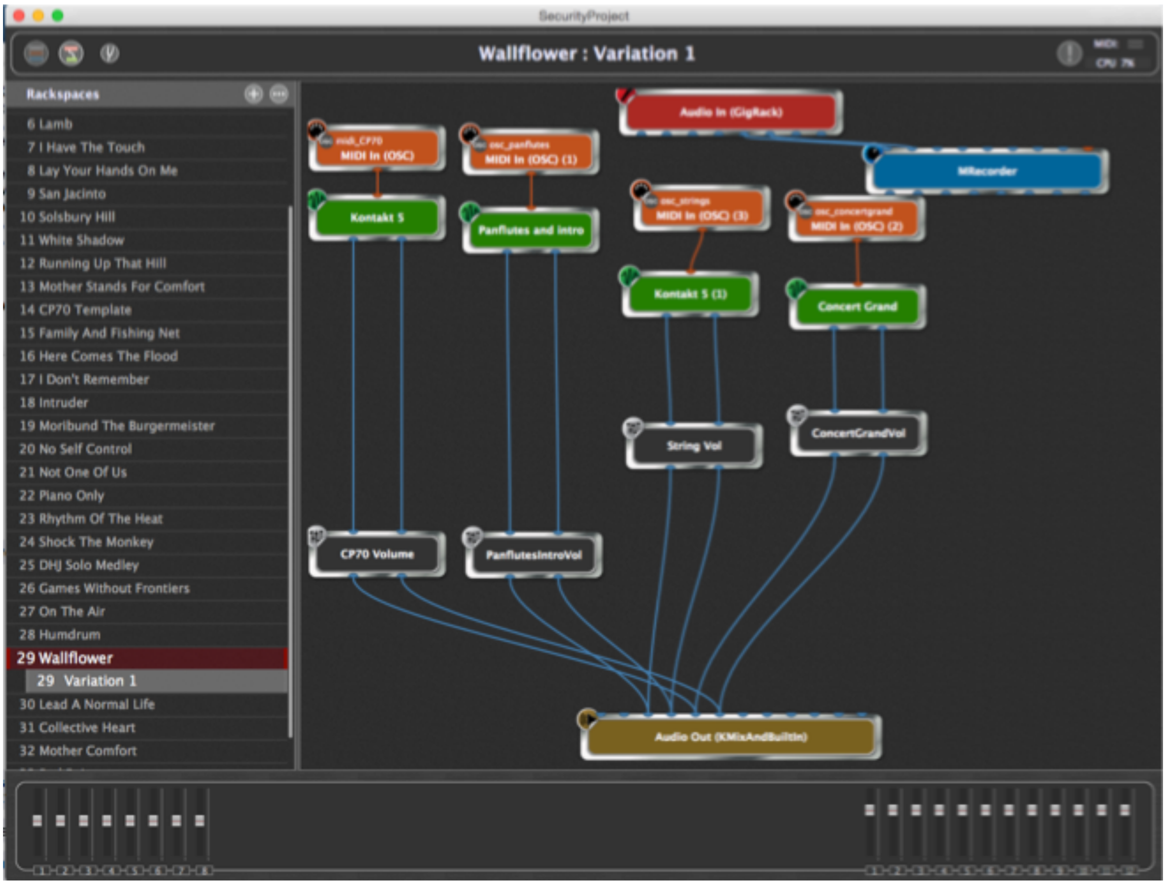
Bypass - disables this plugin
Delete - remove the plugin
Disconnect - remove all connections to and from this plugin
Rename - change the displayed name - this is useful if you are using the same plugin many times and you'd prefer to display the name of the sound (say) that the plugin is producing
Duplicate - create a new plugin instance (quicker than using the menus)
OSC (See section on OSC management)

Scripting (Not available in current version)
Edit plugin - open the plugin editor (you can also just double-click on the plugin block)
Replace plugin - shows the plugin menu so you can use a different plugin

Rackspaces and Variations

Gig Performer allows you to have multiple Rackspaces --- each Rackspace represents a collection of plugins connected together however you wish. You can switch from one rackspace to another any time you want, either by using UP/DOWN commands (if you're a guitarist using a MIDI pedal board controller for example), by sending MIDI program change messages with optional bank switching messages as well as through OSC (see the chapter on OSC for more on this).

1 Rackspace list



SecurityProject

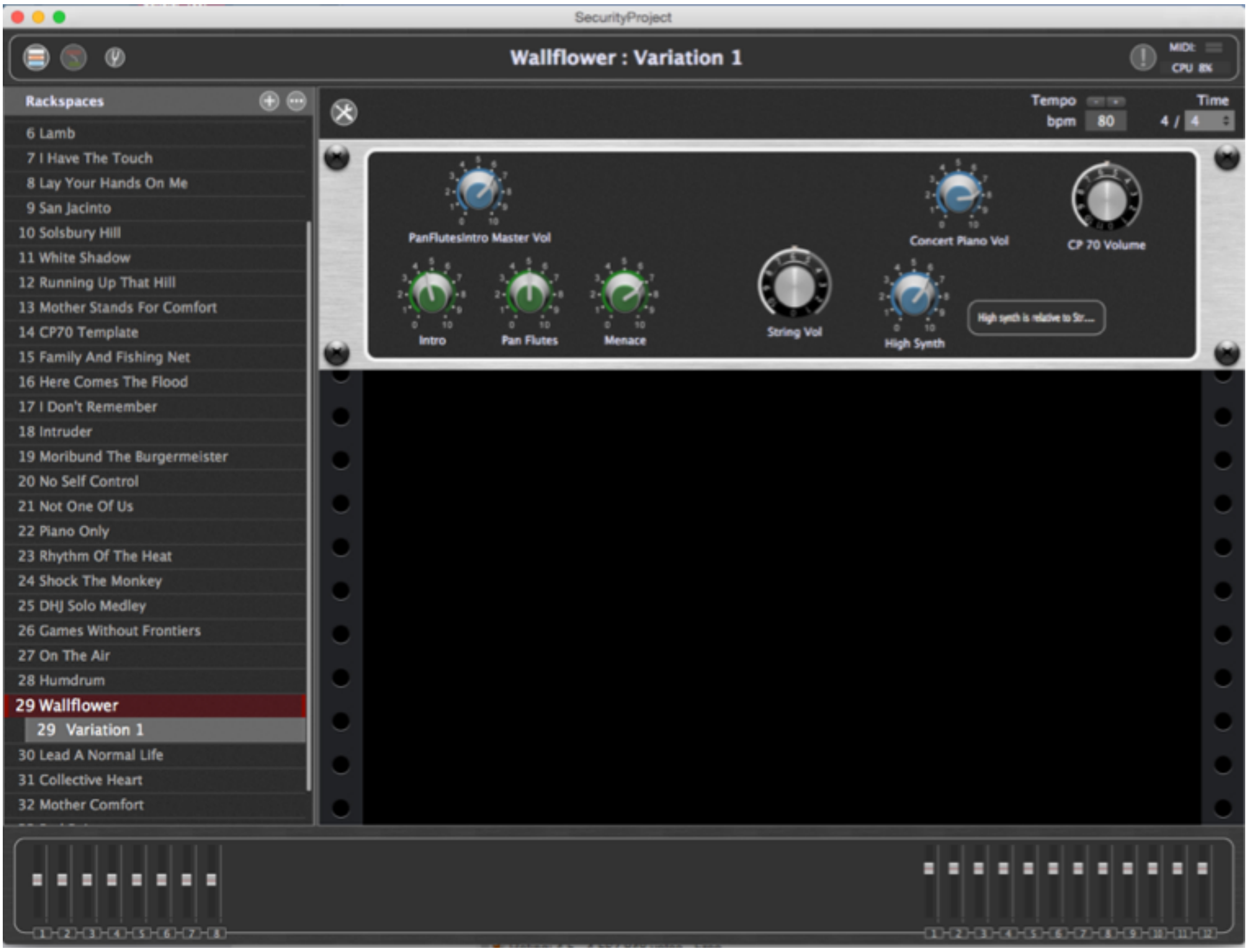
Wallflower : Variation 1

Rackspaces

6 Lamb
7 I Have The Touch
8 Lay Your Hands On Me
9 San Jacinto
10 Solsbury Hill
11 White Shadow
12 Running Up That Hill
13 Mother Stands For Comfort
14 CP70 Template
15 Family And Fishing Net
16 Here Comes The Flood
17 I Don't Remember
18 Intruder
19 Moribund The Burgermeister
20 No Self Control
21 Not One Of Us
22 Piano Only
23 Rhythm Of The Heat
24 Shock The Monkey
25 DHJ Solo Medley
26 Games Without Frontiers
27 On The Air
28 Humdrum
29 Wallflower
29 Variation 1
30 Lead A Normal Life
31 Collective Heart
32 Mother Comfort

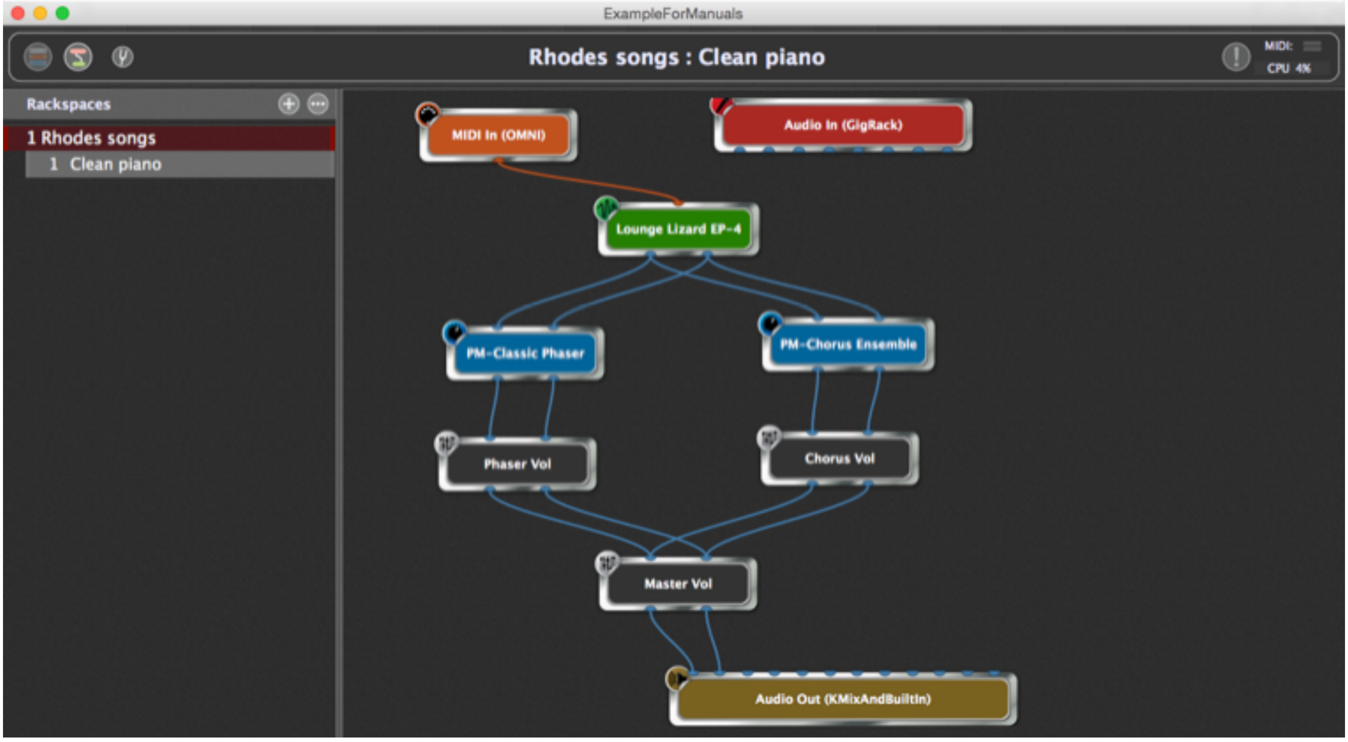
For example, here is a list of the rackspaces used by one of our team members with his band. The selected rackspace is called Wallflow and we're looking at the connections view.

2 Panel widgets



Switching to the performer view for the same rackspace you can now see the panel widgets that are configured for that rackspace. Note that there does not have to be a one-to-one correspondence between a rackspace and a song. That decision depends totally on the way you want to approach your rig design and because of Gig Performer's ability to switch instantly from one rackspace to another, you may choose to spread your song out over multiple rackspaces.

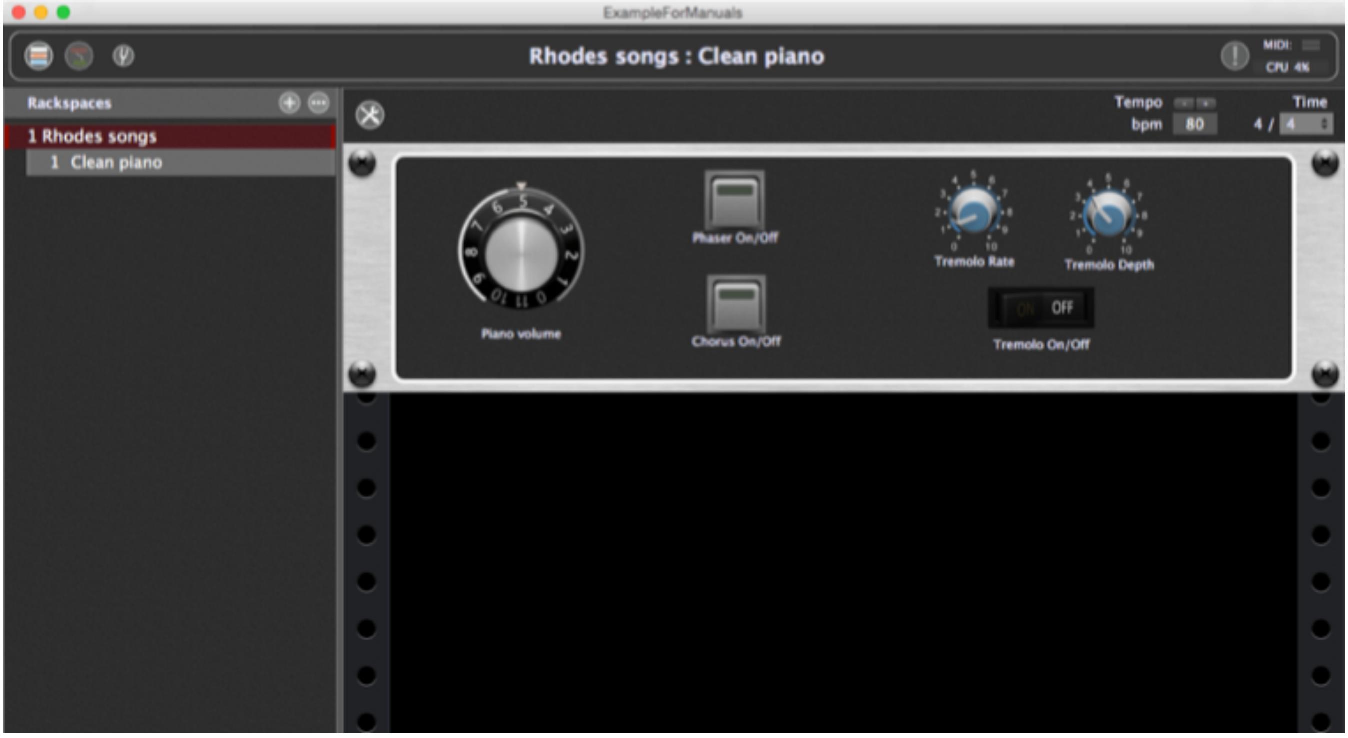
3 Variations



However, there are other circumstances where it makes sense to use exactly the same plugin setup but with different settings. For example, suppose you are performing a number of songs, all of which use an electric piano with a chorus pedal, phaser pedal and for different songs you just want to enable/disable one or both of those pedals or you just want to change the phase rate or the piano's tremolo amount.

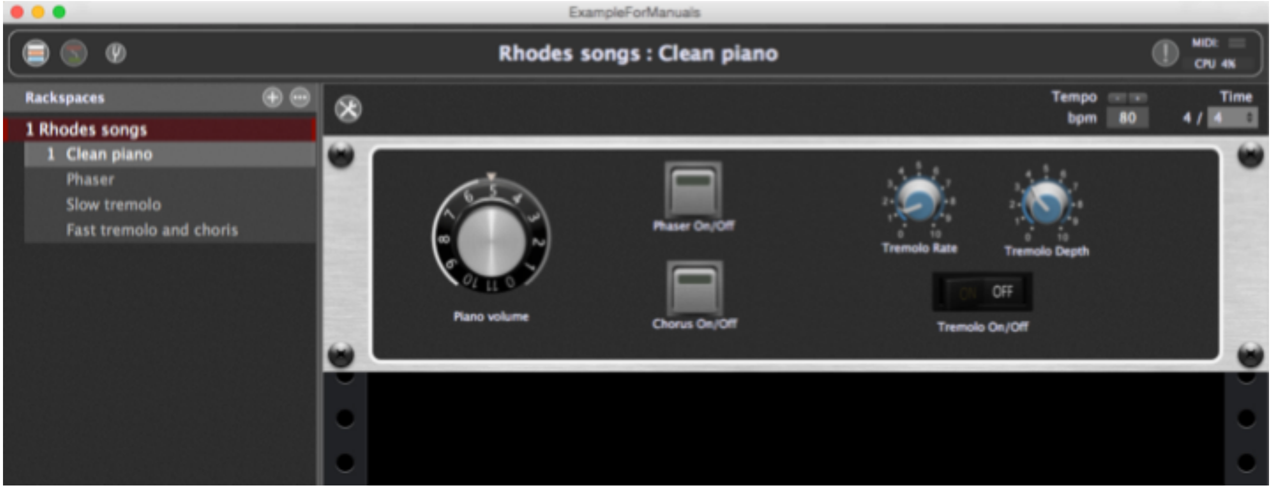
Here, we have created a setup like the one shown here consisting of the Lounge Lizard electric piano, a phaser, a chorus, and some gain controls. There is a single rackspace called Rhode songs and there is a default "variation" called Clean Piano

4 Setting widget values in the default variation



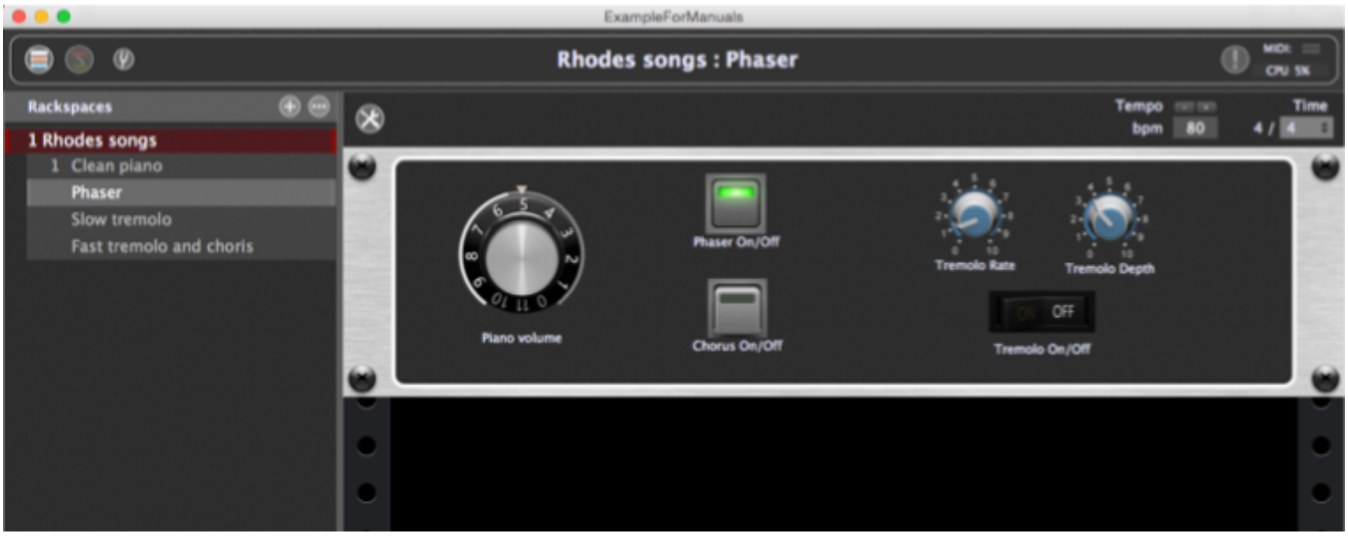
In the performer view, we have created some widgets and attached them to the appropriate parameters of the plugins so that we can control them. Here, we have set the master volume to 5 (a little less than half since this particular volume knob goes up to 11!) and the phaser, chorus and tremolo effects are turned off.

5 Creating more variations



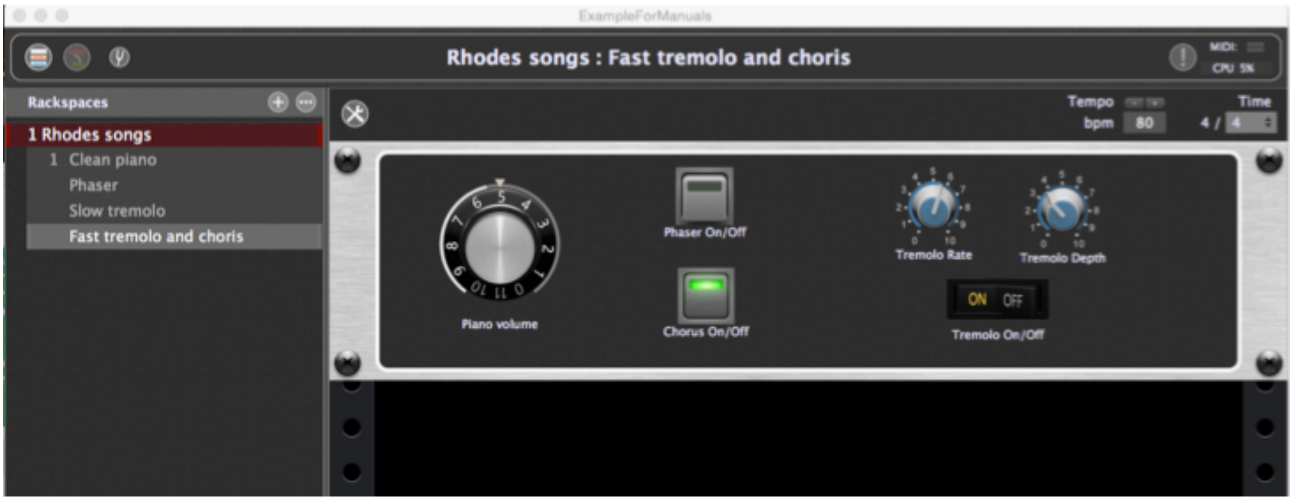
Now create some variations by pressing CMD-N and naming them as desired. Here, we now have four variations defined.

6 Adjusting widget values in different variations



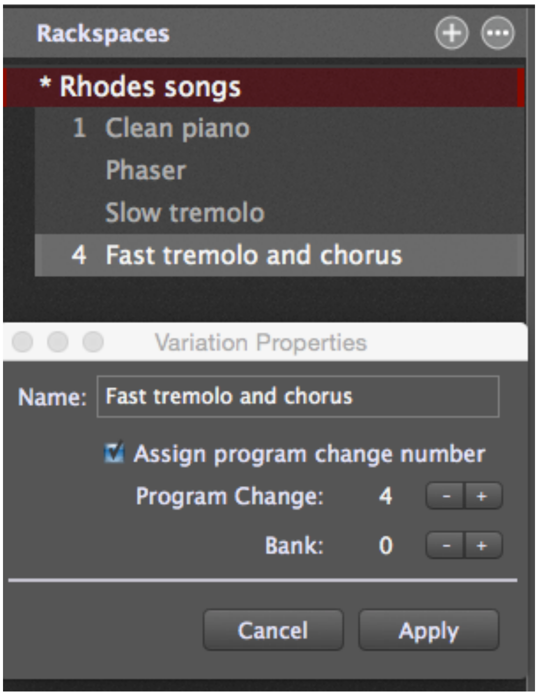
Click on the Phaser variation and then click the "Phase On/Off" button to turn it on. You will now find that if you switch between Clean piano and Phaser, that button will turn off or on respectively as you do so. Note that if you are currently playing a chord on your keyboard, the chord will continue playing though this change.

7



Continue adding variations as needed. Here you can see that the "Fast tremolo and chorus" variation is selected and the widgets reflect those settings. Note that variations do not save the complete state of all the plugins. What they actually save is the value of each widget (Knob position, button on/off and so forth). However, since those widgets are connected to plugin parameters, whenever the widgets change, the parameter values will also.

8 Associating Program Changes with variations



Name: Fast tremolo and chorus

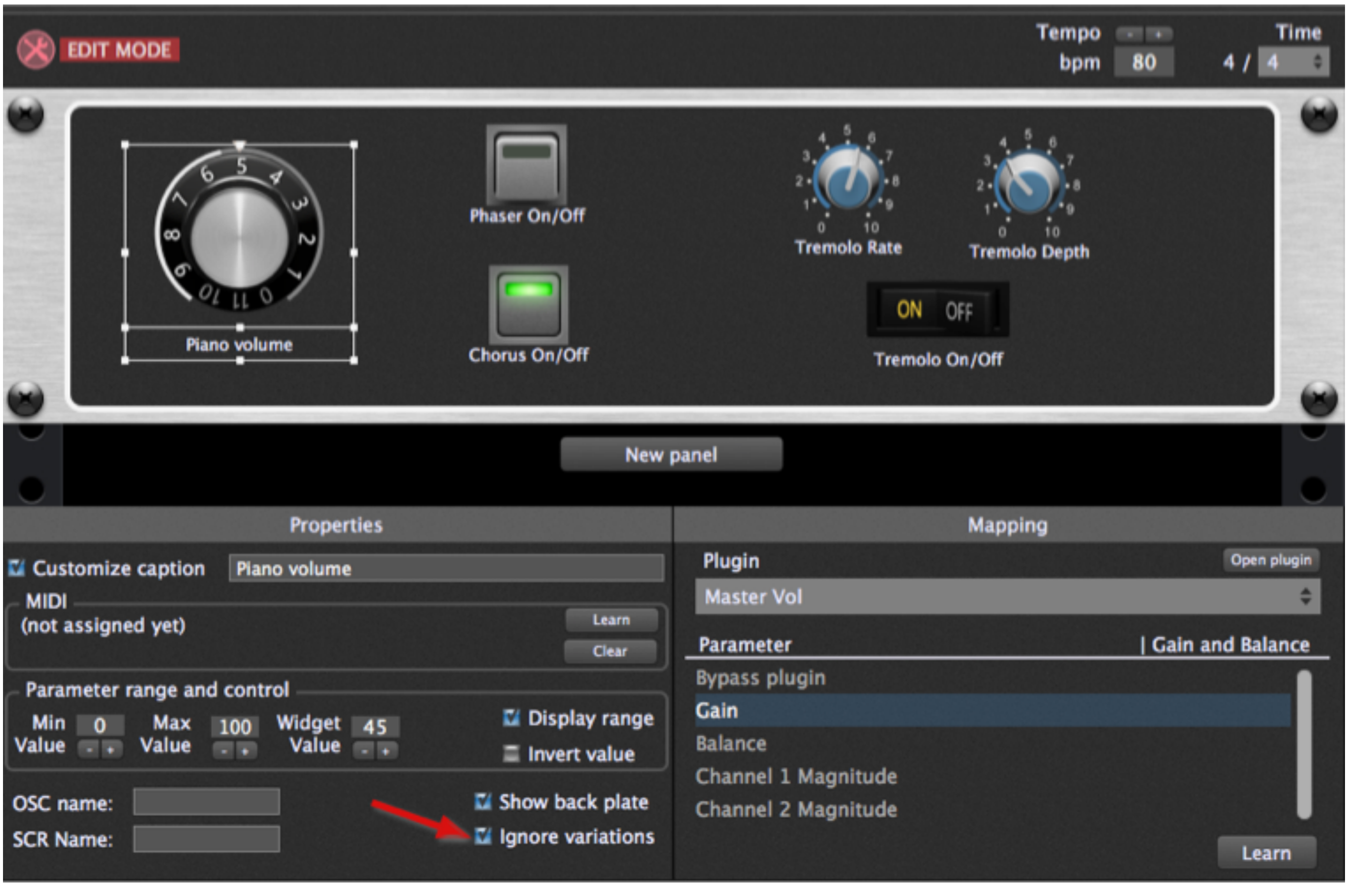
☑ Assign program change number

Program Change: 4

Bank: 0

By the way, you can associate an explicit program change with each variation. Just double-click on the variation and the Variation Properties dialog will appear. Here we have associated program change 4 with the Fast Tremolo and chorus variation. Note that since you must always select a variation, even if there's only one, there is no program change associated with the rackspace itself.

9 Allowing certain widgets to ignore variations



☑ Customize caption: Piano volume

MIDI
(not assigned yet)

Parameter range and control
Min Value 0 | Max Value 100 | Widget Value 45
☑ Display range
☐ Invert value

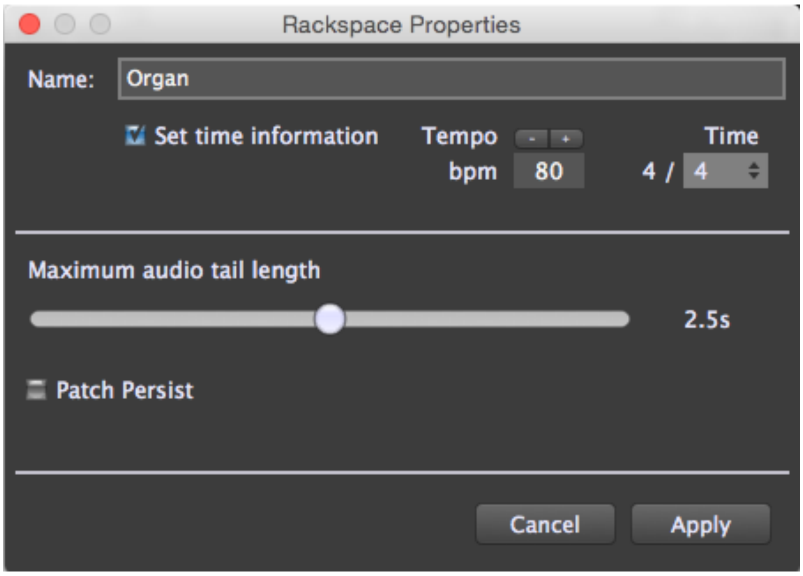
OSC name:
SCR Name:
☑ Show back plate
☑ Ignore variations

Plugin: Master Vol

Parameter | Gain and Balance
Bypass plugin
Gain
Balance
Channel 1 Magnitude
Channel 2 Magnitude

As we mentioned the goal of variations was to allow instant lightweight changes without having to duplicate an entire plugin configuration. This works extremely well but as we used this feature ourselves we realized that it is extremely useful to be able to exclude specific widgets from variation changes. For example, even though you are changing effects like phaser, chorus and so forth, you may not want your volume to change. To prevent the Variations mechanism from being applied to particular widgets, select the widget and check the "Ignore variations" field that is near the center bottom of the widget inspector.

10 Rackspace properties



Name: Organ

☑ Set time information
Tempo bpm 80
Time 4 / 4

Maximum audio tail length 2.5s

☐ Patch Persist

Double-clicking on a rackspace name opens a dialog to display rackspace properties.

Rename
To rename the rackspace, simply type a new name in the name field

Tempo
A rackspace can have its own tempo value. If you do not set explicit information here, the rackspace will

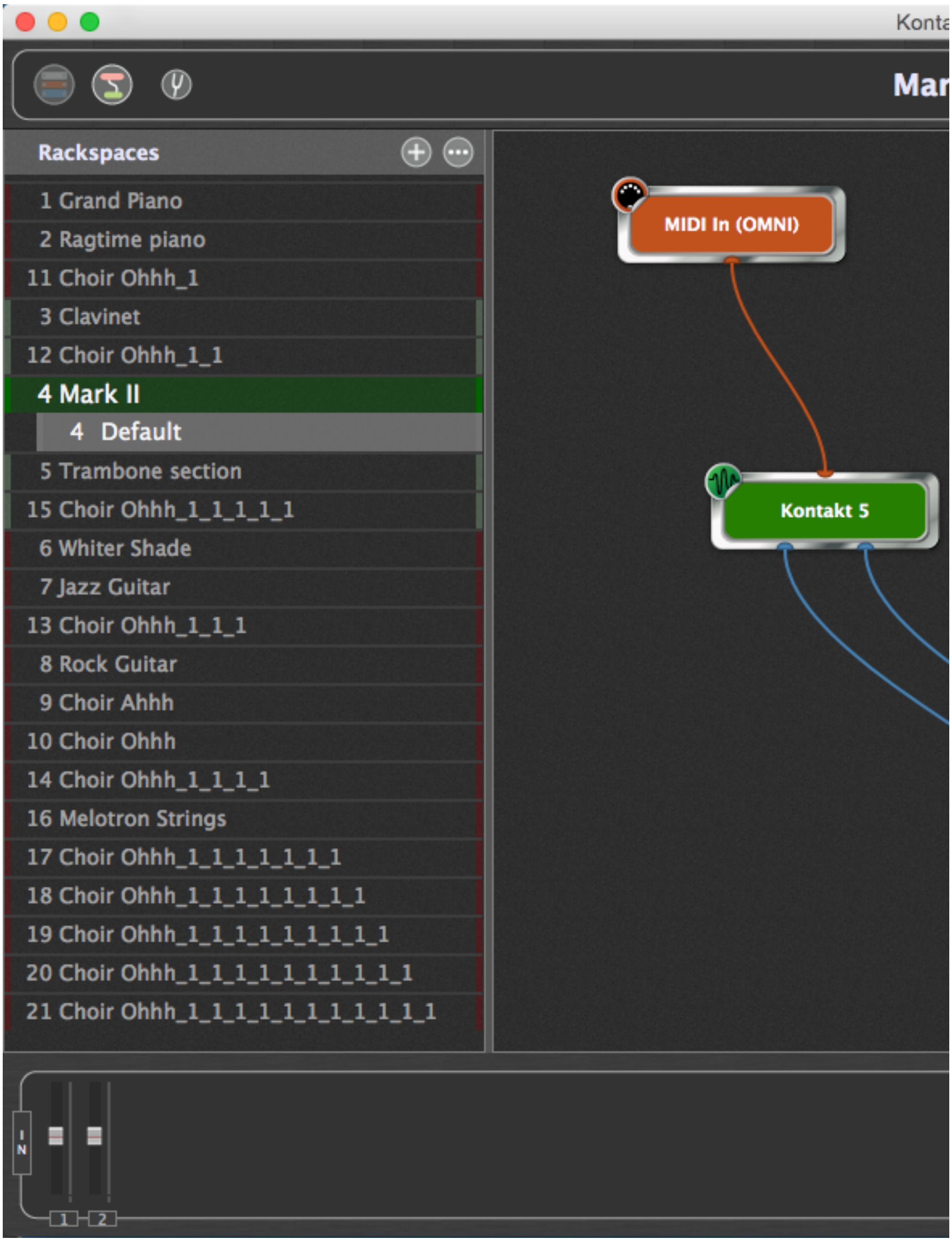
use the global tempo information

Tail length
You can control the maximum time before a sound will be cut off when you switch rackspaces. Often known as the audio tail, this feature is useful for effects such as reverb or echo. The maximum length is 5 seconds

Patch Persist
When you switch from one rackspace to another, all synth plugins for the old rackspace are immediately silenced even if you are holding notes down or have pressed the sustain pedal. If you enable Patch Persist, then notes that were held down will continue to sound until you release the keys for those notes.

Press Apply when you are happy with all your changes

11 Predictive loading



One of the biggest problems for live performers is how much memory and CPU cycles are required to hold all the desired plugins. 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 instantly from one rackspace to another.

Typically, the alternative approach is to load just one set of plugins at a time. This will certainly reduce RAM and CPU requirements considerably but at the cost of having to wait while a song is loaded.

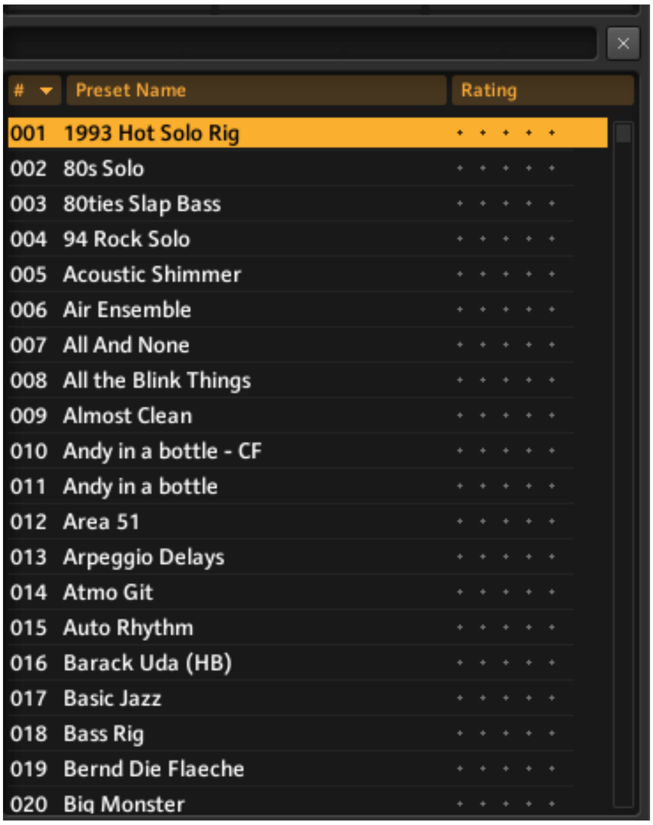
With Gig Performer, there is now an exciting alternative to these extremes which will be very attractive to most users with significant audio hosting requirements.

Predictive Loading is a mechanism where just a few rackspaces are loaded. As you move from one rackspace to the next, the next few rackspaces are loaded in the background (you can control how many) and earlier rackspaces are unloaded in the background. There is absolutely no impact on audio, you still get glitchless switching and Patch Persist but the assumption is that you will stay on a rackspace for some minimum amount of time so that subsequent rackspaces can be loaded. You essentially trade off RAM against spontaneity. By giving up the ability to move to an arbitrary rackspace instantly, you can reduce your RAM requirements and CPU utilization as well. This approach will work extremely well for anyone following a set list.

Controlling a widget from a MIDI device

You can control widgets (knobs, sliders, buttons etc) from your physical devices. Gig Performer makes it easy to learn the appropriate MIDI commands for this purpose

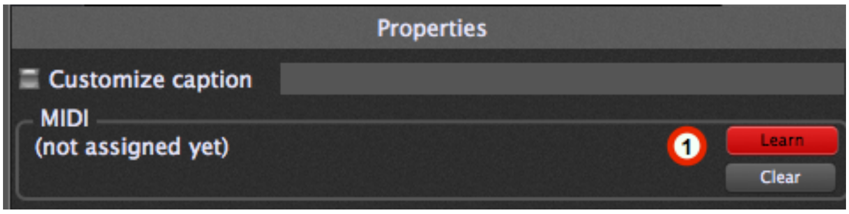
1



#	Preset Name	Rating
001	1993 Hot Solo Rig	
002	80s Solo	
003	80ties Slap Bass	
004	94 Rock Solo	
005	Acoustic Shimmer	
006	Air Ensemble	
007	All And None	
008	All the Blink Things	
009	Almost Clean	
010	Andy in a bottle - CF	
011	Andy in a bottle	
012	Area 51	
013	Arpeggio Delays	
014	Atmo Git	
015	Auto Rhythm	
016	Barack Uda (HB)	
017	Basic Jazz	
018	Bass Rig	
019	Bernd Die Flaeche	
020	Big Monster	

In widget edit mode, drag a knob on to a panel (we assume you know how to do this by now). Note there is a section called MIDI and underneath it says (not assigned yet)

2



Properties

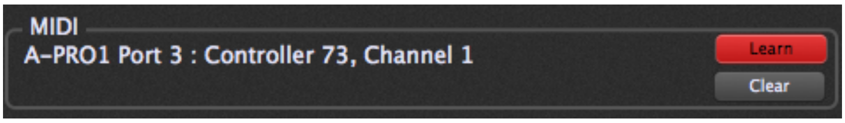
□ Customize caption

MIDI

(not assigned yet)

Click the Learn button (1)

3



MIDI

A-PRO1 Port 3 : Controller 73, Channel 1

Next, move a slider or knob on your keyboard. In this example, I moved one of sliders on a Roland A800 keyboard controller. The device is called "A-PRO1 Port 3" and the slider is sending CC number 73 in MIDI channel 1

Note also that the widget (slider or knob) that you placed on the panel is now also moving as you move the physical slider.

If you wish, while Learn is still enabled, you can move a different physical knob or slider and the information below MIDI will switch to show the new information.

When you're done, click Learn again to turn off Learn mode.

Learning aftertouch messages

Learning aftertouch messages requires some extra care. Normally, when you press a key, Gig Performer will recognize and learn a Note On event. If Learn is still enabled and you release the key, then Gig Performer will recognize and learn a Note Off message. However, aftertouch messages (also known as channel pressure messages) occur in between a Note On and Note Off, as you push the key harder before releasing it. To learn the aftertouch message itself, you should do the following:

a) Go into learn mode
b) Press a note
c) Start pushing harder so that Aftertouch messages are now seen
d) Turn off Learn mode BEFORE you release the note

Internal plugins

Gig Performer only comes with a few plugins of its own. We made the decision to not design our own general purpose plugins as there are so many available plugins, both free and commercial. We fully expect most users to want to use their own favorite plugins. Having said that, we will add more internal plugins to address functionality that is not readily available from other developers and vendors. Our current list of internal plugins are

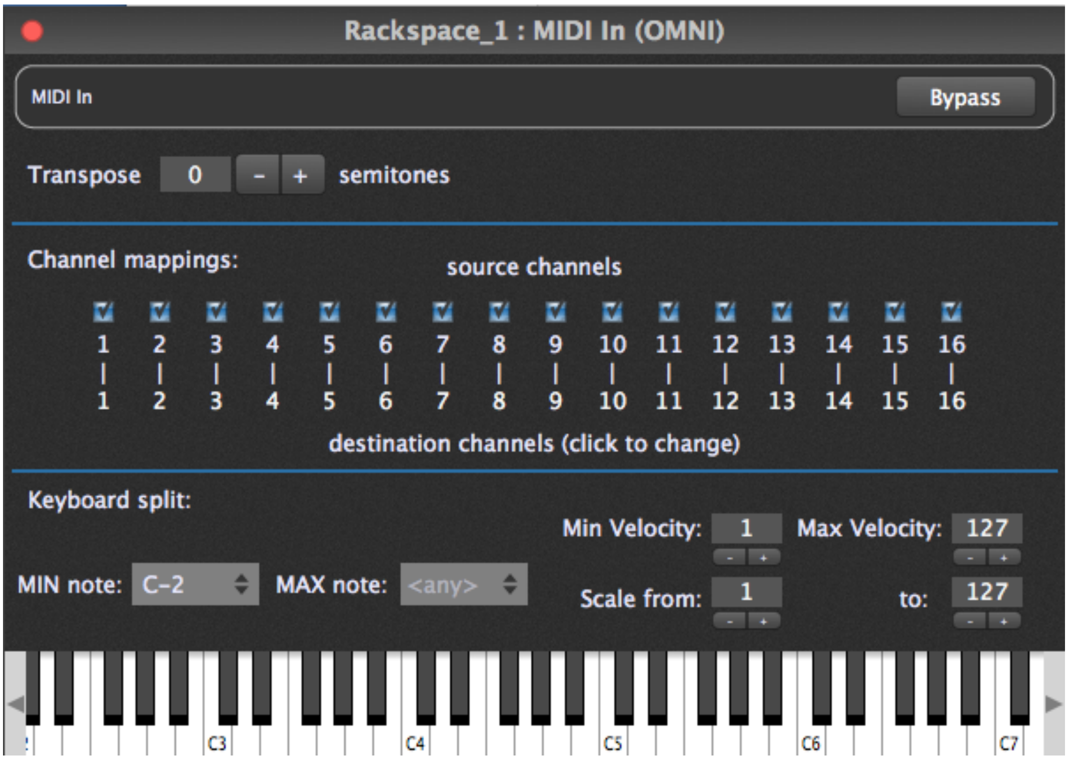
Audio In - represents the current physical device used to receive audio
Audio Out - represents the current physical device used to send audio out
These plugins cannot be removed from your rackspace.

MIDI In - receive MIDI messages from external devices and from OSC sources. There are several versions of this plugin. The OMNI version, the default in any new rackspace, will forward MIDI events from ANY device in your system. You can also create Midi In plugins that only forward incoming MIDI messages from specific MIDI devices.
MIDI Out - send MIDI messages to external synthesizers. You can also configure this plugin to automatically send out program changes each time rackspace containing it is activated.
MIDI Filter - block or remap MIDI messages

Gain/Balance - adjust the gain and the balance of an incoming stereo signal
Gain Control (Mono) - as above but for a mono audio source
Gain/Balance (Mono to Stereo) - allows you to pan an incoming mono source across two channels

As with external plugins, our internal plugins have parameters that can be used for automation and remove control

1 Midi In



Transpose 0 semitones

Channel mappings:

source channels

☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

destination channels (click to change)

Keyboard split:

Min Velocity: 1 Max Velocity: 127

MIN note: C-2 MAX note: <any>

Scale from: 1 to: 127

The MIDI In plugin supports channel mapping, min/max note values (for keyboard splits), transpose and "sticky notes".

Channel mapping

You can block a channel or reroute incoming messages from a certain channel to another channel

Transpose
You can optionally transpose all incoming note messages.

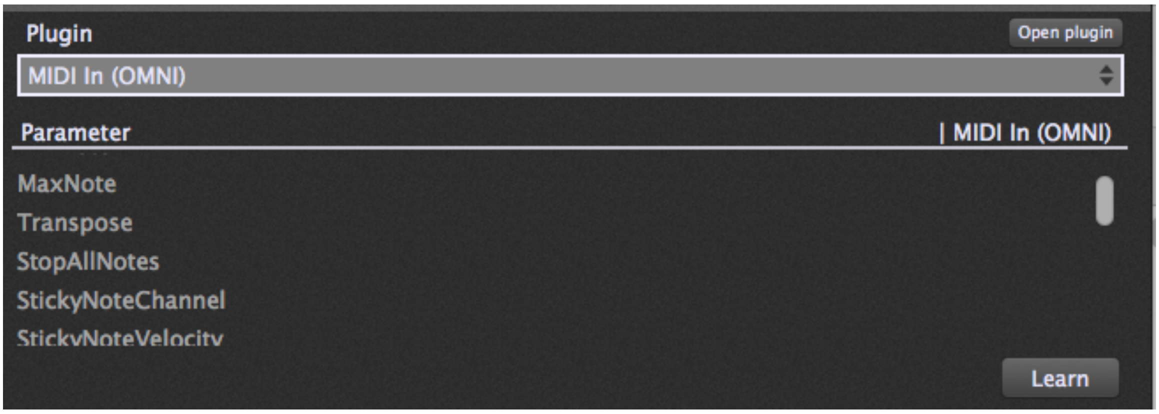
Keyboard splits
You can select a range of notes that should be allowed through, all others will be blocked.

Velocity scaling
You can scale the velocity of all incoming Note messages to within a desired range, including a single value. For example, by setting both the Scale from: and to: fields to 64, all incoming notes will have their velocity changed to 64

Velocity layering
You can use velocity ranges to determine whether particular notes should be played based on how loud (or soft) you press a key. For example, you could connect a Midi In plugin to a piano synth but only hear the piano if you press keys hard. This is particularly useful if you want to have multiple sounds available but control when some are heard based on how you play.

Using velocity scaling with velocity layering
After you have defined a velocity range inside of which notes will be played, you may want to change the actual velocities that are sent to your synth plugin. For example, you may have created a piano sound that you only want to be heard if the velocity value you play is great than 80 (say). However, you may still want the piano to respond as if it was being played quietly. For example, set the min/max velocities to 80/127 respectively so that you only hear the piano when you hit the keys hard. Then select the Scale from/to values to 1/50 respectively. Notes sent in with velocity 80 will produce notes with velocity 1. Notes with velocity 127 will produce notes with velocity 50. Notes in between will be mapped somewhere in between as well.

2 All Notes Off



Plugin

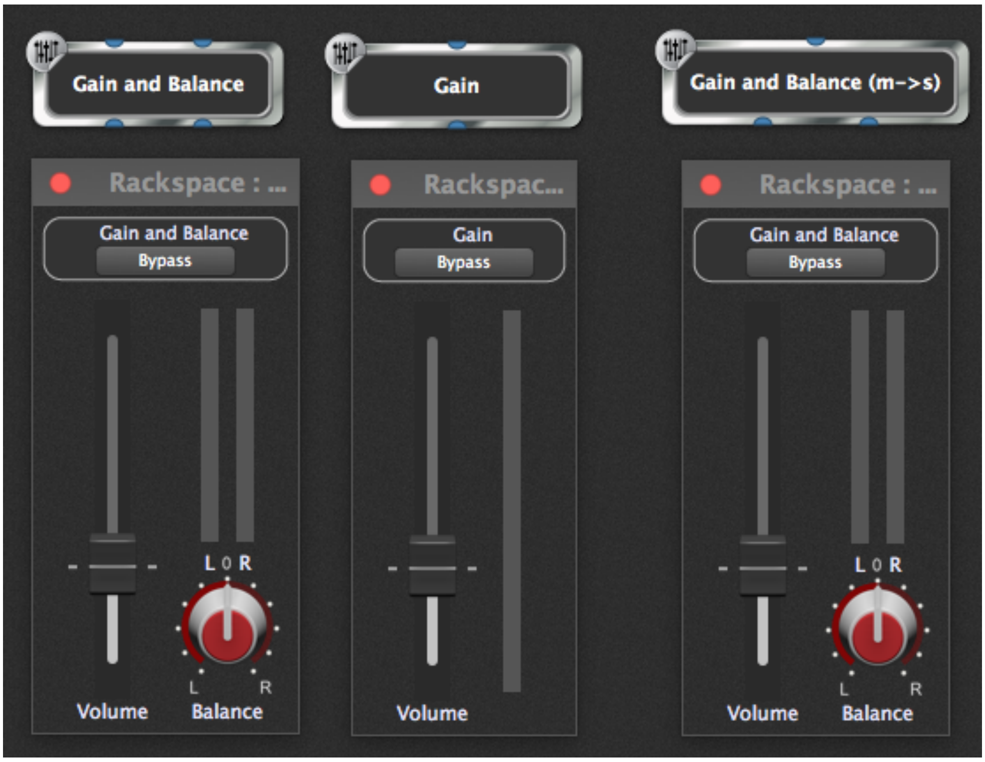
MIDI In (OMNI)

Parameter | MIDI In (OMNI)

MaxNote
Transpose
StopAllNotes
StickyNoteChannel
StickyNoteVelocity

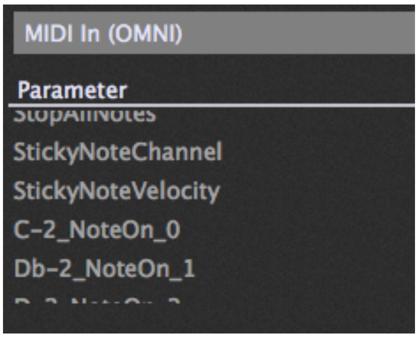
There is also an explicit "StopAllNotes" command which can be used to turn off MIDI notes when switching variations. Although normally one wouldn't do this because the whole point of variations is to allow you to continue holding notes down while changing effect values, it is useful if you want to use 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, a sustain OFF message is sent and a Center PitchBend message is sent. Just associate it with any widget, typically a button, and it will trigger automatically when you change variations.

3 Gain and Balance



The Gain/Balance plugins lets you adjust the gain, the location in the stereo field (except for the Mono version!) and also provides instantaneous magnitudes that can be used to audio drive meter widgets

4 Sticky Notes



MIDI In (OMNI)

Parameter

StopAllNotes

StickyNoteChannel

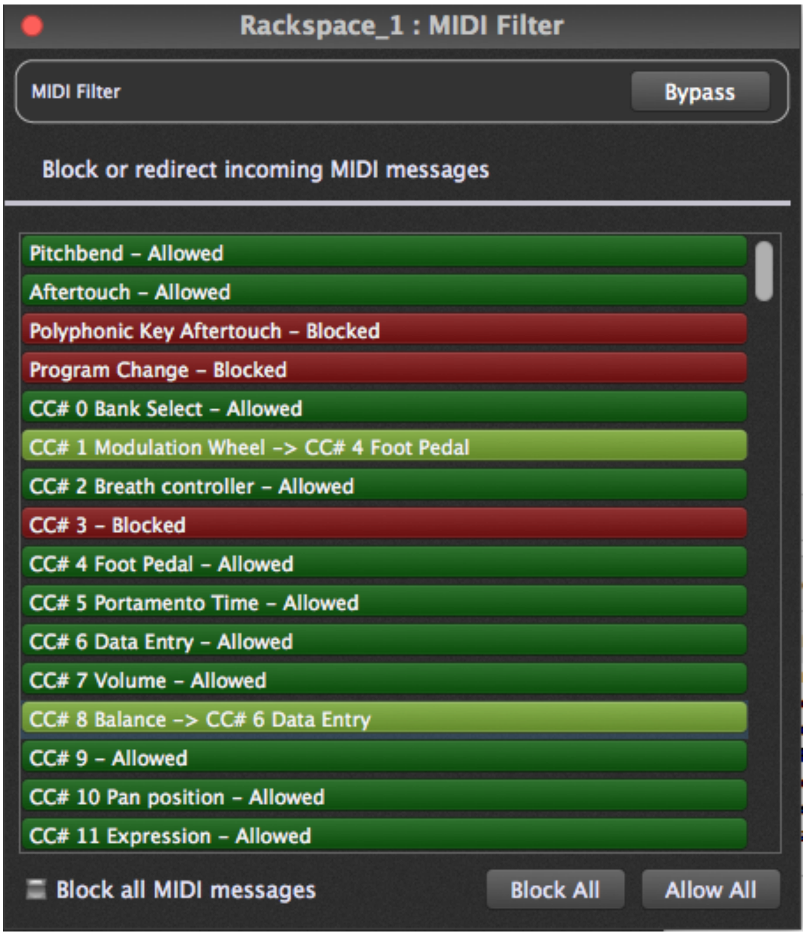
StickyNoteVelocity

C-2_NoteOn_0

Db-2_NoteOn_1

Sticky notes are designed to allow loops or sample playback by sending Note On events only, as if you played a note but didn't release it. There are parameters for setting the MIDI channel and the default velocity and then there is a parameter for each MIDI note. You can select one and associate it (typically) with a button that, when pressed, will send the associated note. When released, the note off event will be sent to stop the sound.

5 Midi Filter



Rackspace_1 : MIDI Filter
MIDI Filter
Bypass
Block or redirect incoming MIDI messages
Block All
Allow All

Pitchbend – Allowed
Aftertouch – Allowed
Polyphonic Key Aftertouch – Blocked
Program Change – Blocked
CC# 0 Bank Select – Allowed
CC# 1 Modulation Wheel -> CC# 4 Foot Pedal
CC# 2 Breath controller – Allowed
CC# 3 – Blocked
CC# 4 Foot Pedal – Allowed
CC# 5 Portamento Time – Allowed
CC# 6 Data Entry – Allowed
CC# 7 Volume – Allowed
CC# 8 Balance -> CC# 6 Data Entry
CC# 9 – Allowed
CC# 10 Pan position – Allowed
CC# 11 Expression – Allowed

□ Block all MIDI messages

The Midi Filter plugin allows you to block certain kinds of messages and also to map certain messages to other messages. Common uses of this filter might to block sustain pedal messages, convert mod wheel messages into volume messages and map aftertouch messages to into mod-wheel messages. Clicking on any line causes a popup menu through which you can make your choice.

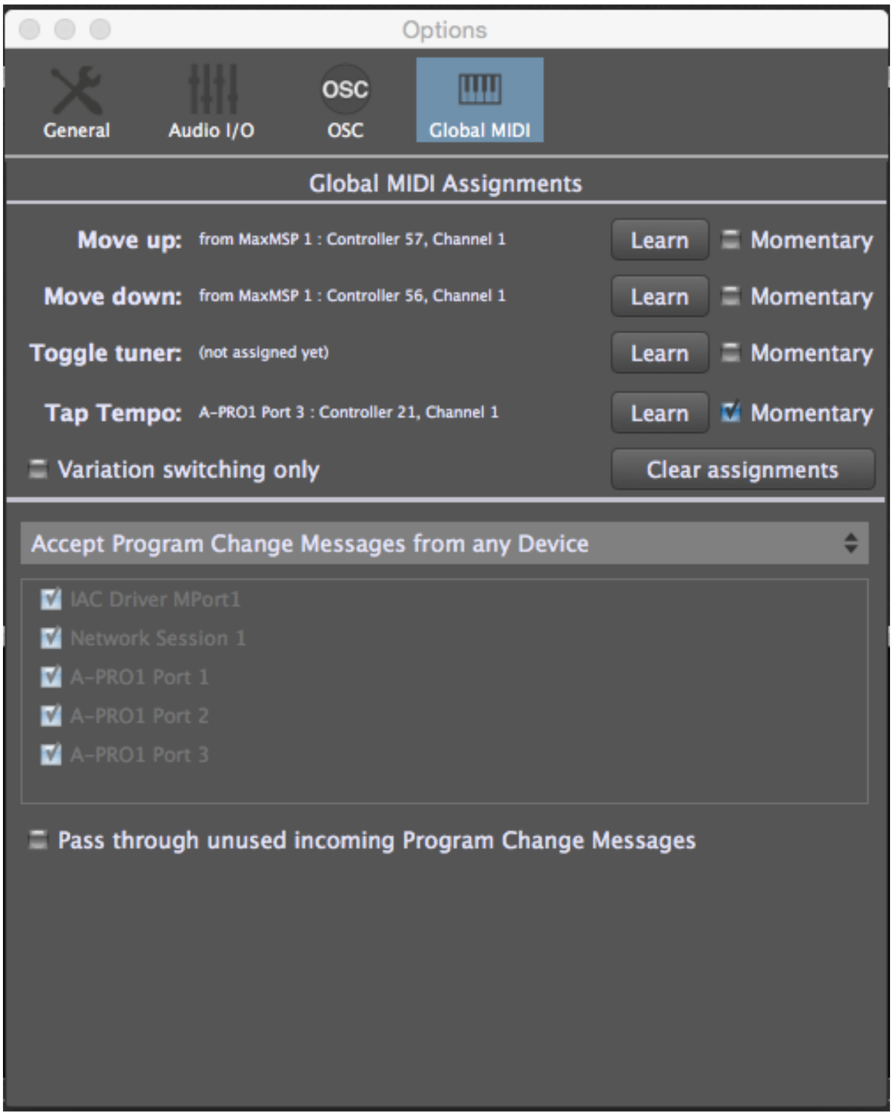
Allowed: Messages of this type are passed through
Blocked: Messages of this type are blocked
-> : Messages of this type are converted into messages of another type

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.

Tap Tempo

Gig Performer supports Tap Tempo. You can use this in conjunction with plugins such as delays or echoes so that their delay values can be synced to whatever tempo in which you are playing.

1 Associating physical control with Tap Tempo



Global MIDI Assignments

Move up: from MaxMSP 1 : Controller 57, Channel 1 — □ Momentary

Move down: from MaxMSP 1 : Controller 56, Channel 1 — □ Momentary

Toggle tuner: (not assigned yet) — □ Momentary

Tap Tempo: A-PRO1 Port 3 : Controller 21, Channel 1 — ☑ Momentary

□ Variation switching only

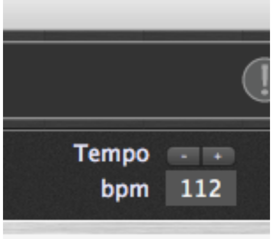
Accept Program Change Messages from any Device

☑ IAC Driver MPort1
☑ Network Session 1
☑ A-PRO1 Port 1
☑ A-PRO1 Port 2
☑ A-PRO1 Port 3

□ Pass through unused incoming Program Change Messages

You can associate a button or a pedal with the Tap Tempo mechanism through the Global MIDI page of the Options dialog. Click on the Learn button for Tap Tempo and press the button or pedal you want to use for detecting tempo. You will most likely need to enable the Momentary option as well.

2 Responding to tempo



Tempo bpm 112

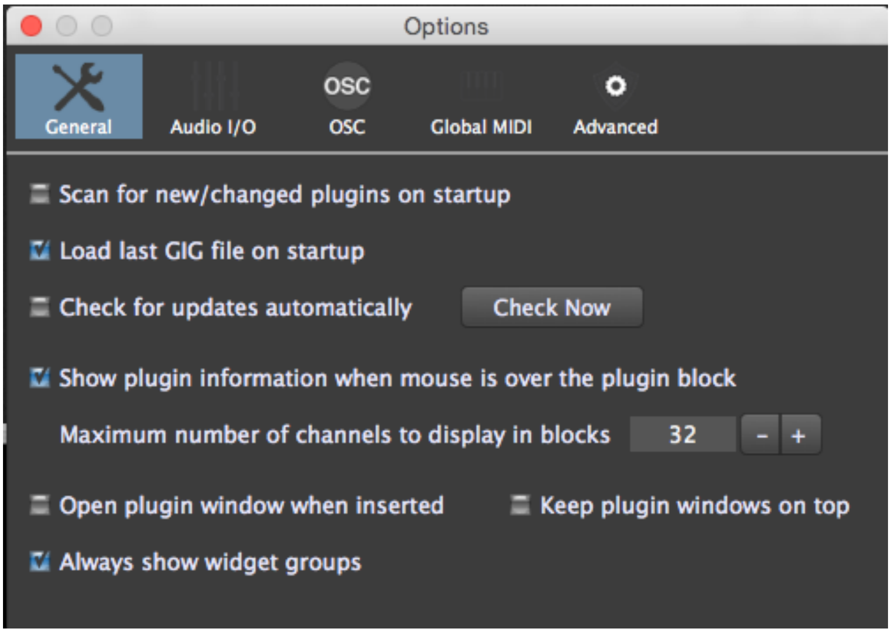
Once you have learned the MIDI message and you start tapping, the BPM display at the top right of the main window will show you the current tempo that it has detected. All plugins that respect global tempo will now synchronize to this tempo.
It may be necessary to enable 'sync' to external tempo on individual plugins. Please check your plugin documentation for more information on this topic

Options and preferences

Gig Performer has numerous preferences to customize its behavior. The options dialog is divided into four sections:

General
Audio I/O

1 General



☐ Scan for new/changed plugins on startup

☑ Load last GIG file on startup

☐ Check for updates automatically

☑ Show plugin information when mouse is over the plugin block

Maximum number of channels to display in blocks 32

☐ Open plugin window when inserted ☐ Keep plugin windows on top

☑ Always show widget groups

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 you are not updating plugins.

Load last GIG file on startup
When enabled, Gig Performer will attempt to load the last gigfile that was used.

Show plugin information when mouse is over the plugin block
When enabled, moving your 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
Some plugins have a huge number of audio channels most of which will never be used but it makes the plugin block very unwieldly. This option prevents plugins from showing too many audio channels. Note that the restriction does not apply to the Audio In and Audio Out blocks.

Open plugin window when inserted
When checked, the plugin editor window will open automatically whenever you insert a plugin. Disable this feature if you prefer to create many plugin blocks first and edit them all later.

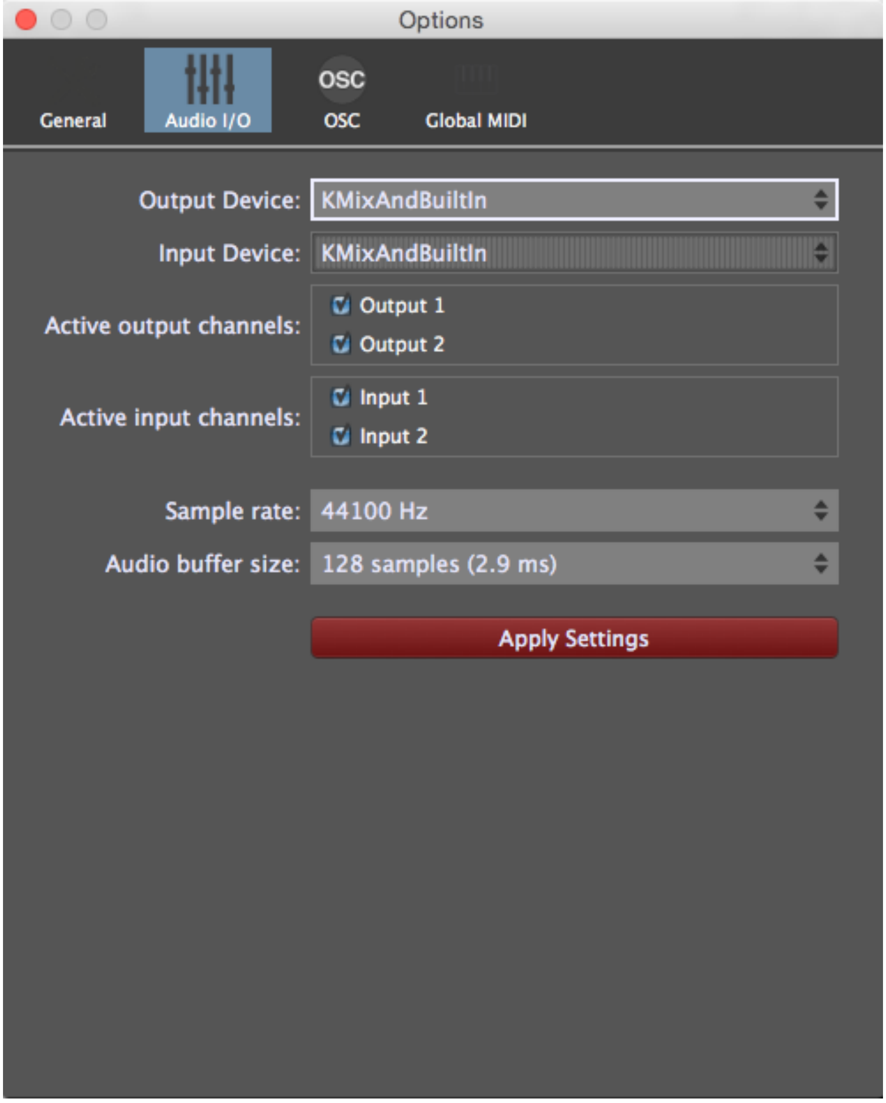
Keep plugin windows on top
When enabled, open plugin window editors will always appear on top of other windows

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

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.

2 Audio I/O



Output Device: KMixAndBuiltIn

Input Device: KMixAndBuiltIn

Active output channels:
☑ Output 1
☑ Output 2

Active input channels:
☑ Input 1
☑ Input 2

Sample rate: 44100 Hz

Audio buffer size: 128 samples (2.9 ms)

Output Device
Select the audio output device you want to use. Note: changing the audio output device will cause your entire gig to be reloaded immediately. In general, it is more expedient to change the output device when you just have a single new Gig.

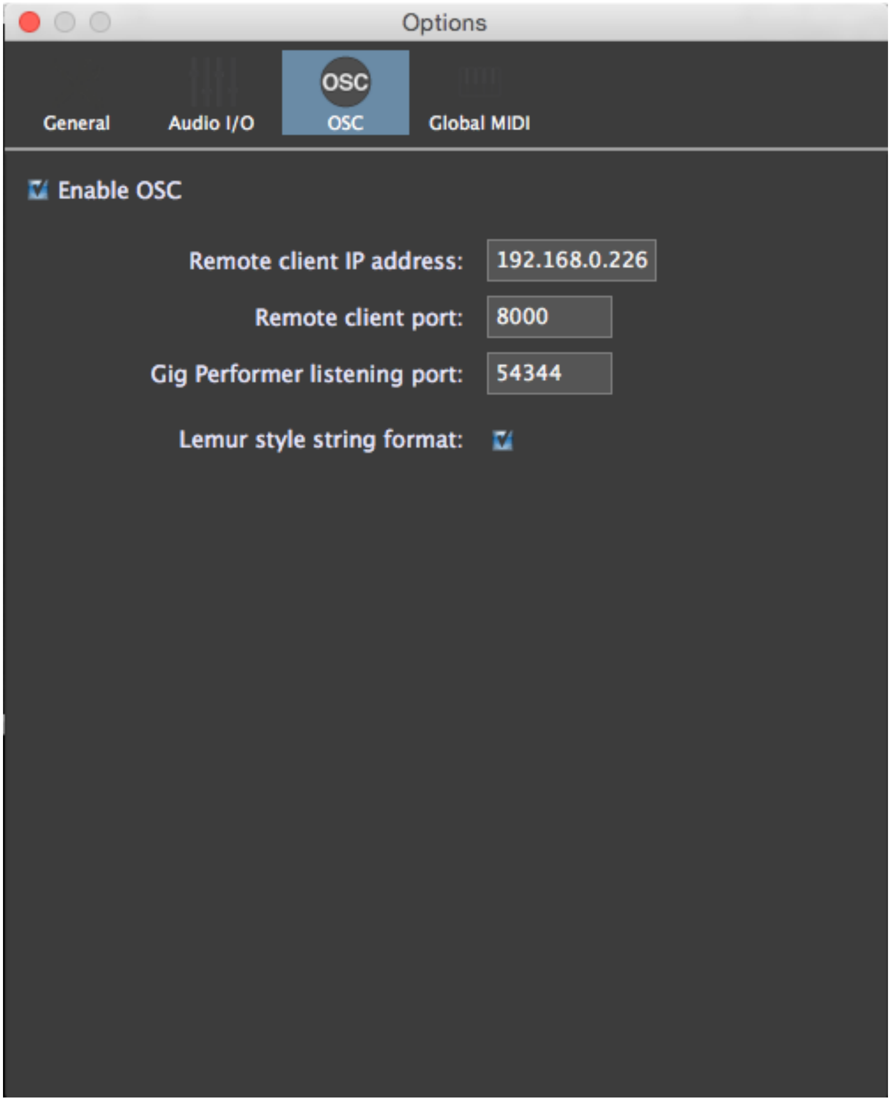
Input Device
Select the audio input device you want to use. Note: changing the audio input device will cause your entire gig to be reloaded immediately. In general, it is more expedient to change the input device when you just have a single new Gig.

Active output channels/Active input channels
If you have an audio interface with many channels, you may not want to expose all of them to Gig Performer. For example, many interfaces have many more channels available than are accessible through analog connections. 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.

Audio buffer size
The buffer size determines how often samples are actually processed. The smaller the buffer size, the better the latency but at the cost of higher CPU utilization. Depending on your plugins and your computer, you will 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 noises in the output

3 OSC



☑ Enable OSC

Remote client IP address: 192.168.0.226

Remote client port: 8000

Gig Performer listening port: 54344

Lemur style string format: ☑

Enable OSC
Check this to turn on OSC. Please review the section on OSC later on in this document for more information about available OSC commands. Note that a complete tutorial on OSC is beyond the scope of this document. We assume that if you know you want to use OSC then you already have some knowledge/experience with it and/or are using another application through which you want to communicate with Gig Performer over OSC

Remote client IP address
The address of the remote device that is running an OSC application

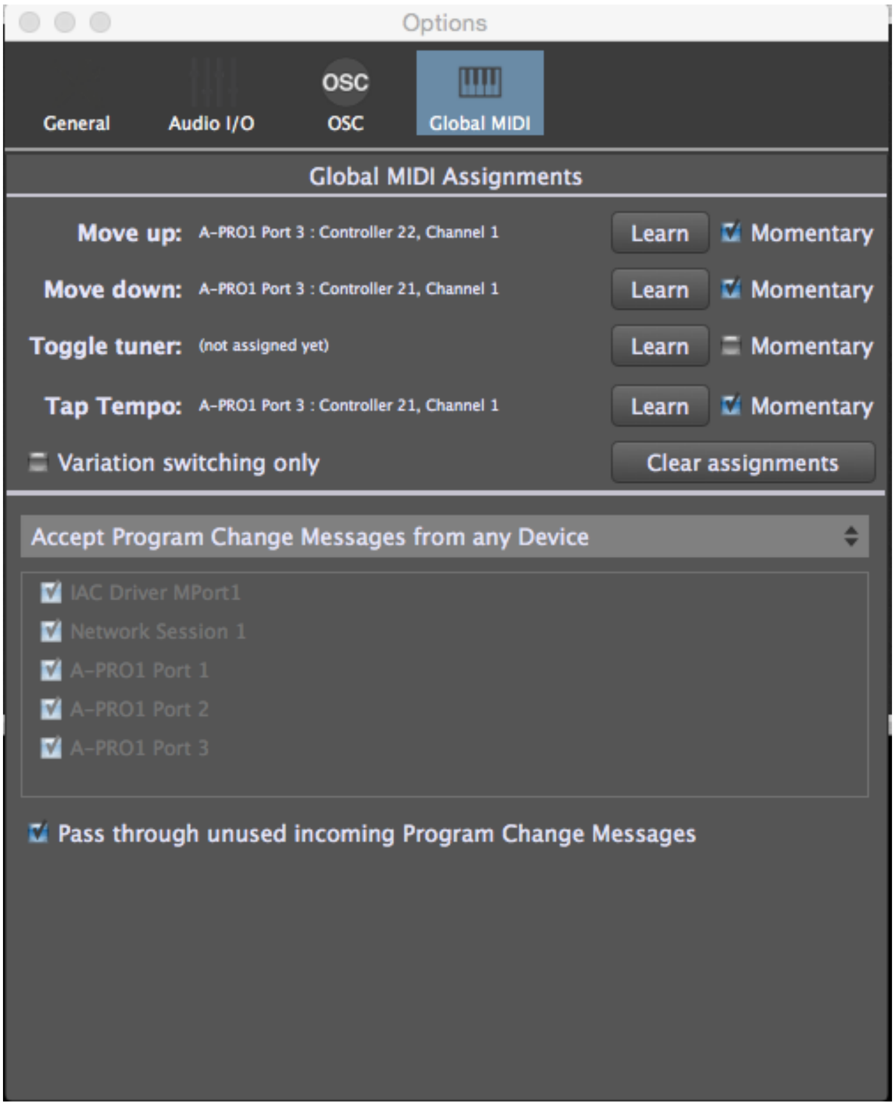
Remote client port
The port on that remote device that is listening for OSC messages over UDP

Gig Performer listening port
The port on which Gig Performer is listening for incoming OSC messages over UDP

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

Note: if you change any settings in this OSC section, you will need to restart Gig Performer. This restriction will be removed in a future version.

4 Global MIDI Assignments



Global MIDI Assignments

Move up: A-PRO1 Port 3 : Controller 22, Channel 1 ☑ Momentary

Move down: A-PRO1 Port 3 : Controller 21, Channel 1 ☑ Momentary

Toggle tuner: (not assigned yet) □ Momentary

Tap Tempo: A-PRO1 Port 3 : Controller 21, Channel 1 ☑ Momentary

□ Variation switching only

Accept Program Change Messages from any Device

☑ IAC Driver MPort1
☑ Network Session 1
☑ A-PRO1 Port 1
☑ A-PRO1 Port 2
☑ A-PRO1 Port 3

☑ Pass through unused incoming Program Change Messages

You can use MIDI Messages to control some behaviors in Gig Performer. To learn an event, click the Learn button for the behavior in which you are interested and then turn the knob/slider or key on your physical device. Check the Momentary box if you have a momentary button that you want to treat as if it were latching.

In this dialog, Gig Performer can learn MIDI messages for the following

Move up
This action will move to the previous variation or to the previous rackspace if you are at the first variation

Move down
This action will 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
This action switches you to or from tuner view

Tap Tempo
Gig Performer will measure incoming events and determine the desired tempo based on the rate the events are received

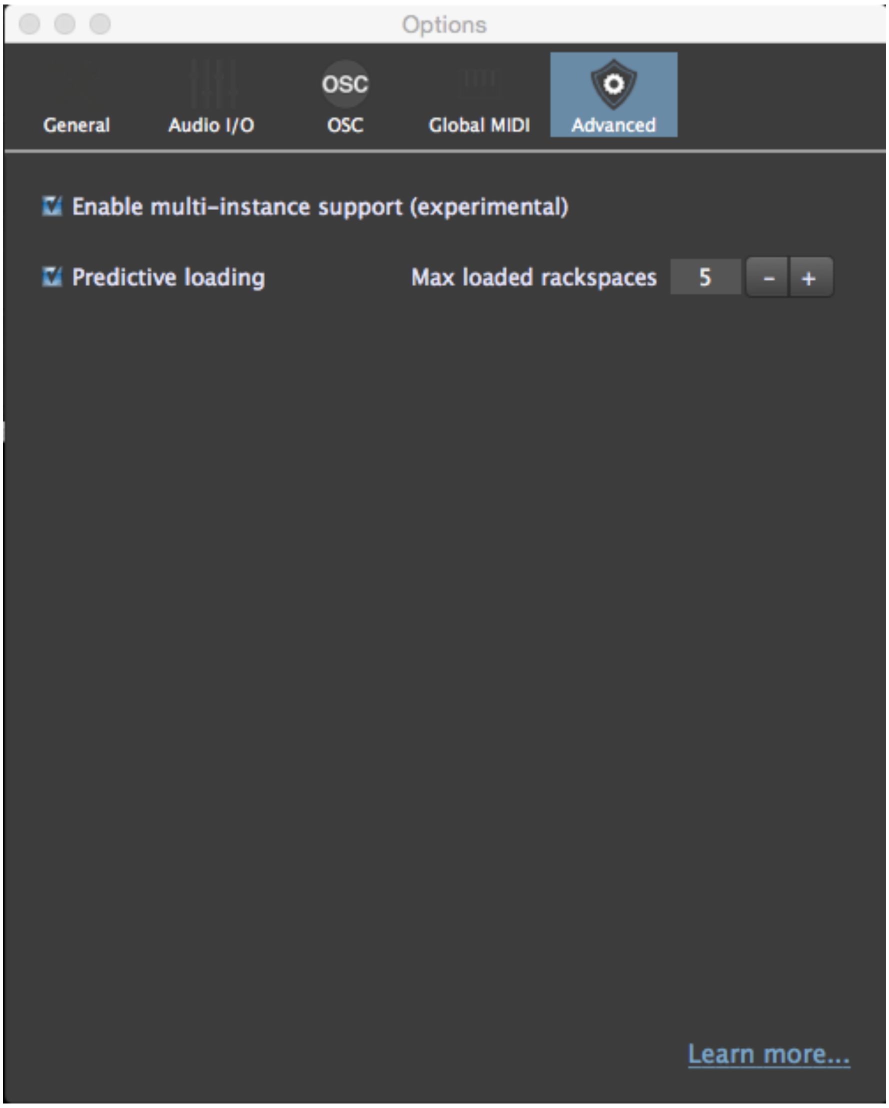
Variation switching only

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

Accept Program change Messages from any Device
Gig Performer can be configured to respond to Program Change messages from your external devices. Depending on your needs, you may want to allow Program Changes to come from any external device or only from designated devices. Use this option to designate the desired devices

Pass through unused incoming Program Change messages
When enabled, if a program change is received for which there is no corresponding rackspace, the program change message will be sent to the plugin that is connected to a MIDI In block that is allowed to receive program changes

5 Advanced



☑ Enable multi-instance support (experimental)

☑ Predictive loading   Max loaded rackspaces 5 − +

Enable multi-instance support (experimental)
Checking this option enables Gig Performer to support multiple independent Gig concerts simultaneously. This mechanism allows you to support multiple musicians (each can have separate control) or to keep some sounds around permanently while continuing to change sounds in a different Gig concert

Predictive loading
Please see our online documentation for more information on this feature. Your needs will determine whether you want this feature enabled

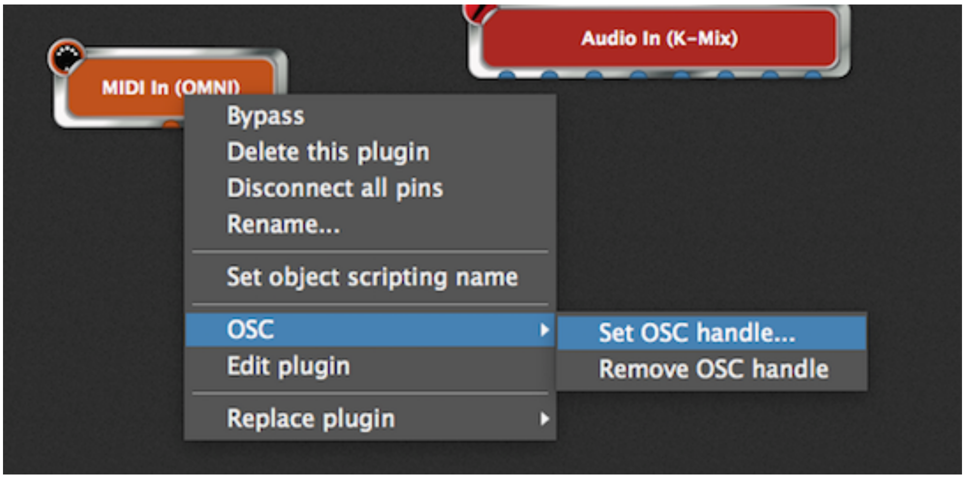
OSC

Gig Performer supports OSC (Open Sound Control) for controlling and responding to widget changes. You can also send MIDI messages over OSC and you can select a desired rackspace and variation via OSC.

MIDI over OSC

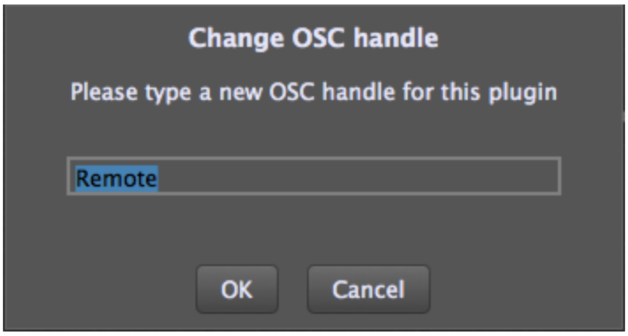
You can send various MIDI messages to a MIDI Input plugin from a remote MIDI source.

1 Enabling OSC



Before you can send MIDI events to a MIDI in plugin over OSC, you need to associate an OSC handle with that plugin. OSC handles must be unique within a single rackspace but you can use the same OSC handle. An OSC handle is just a word comprising of characters only. The handle name is case sensitive. This handle will be used as part of the OSC address target to which you will send your messages

2 Sending MIDI messages over OSC



Change OSC handle

Please type a new OSC handle for this plugin

Remote

For our examples, the OSC handle for the MIDI In plugin is called Remote

/Remote/NoteOn n v c
Send a NoteOn message with the specified note number, velocity and channel (channels are specified from 1 to 16). A velocity value of 0 will be treated as if it were a NoteOff message
E.g.,

/Remote/NoteOn 60 64 1
Sends MIDI NoteOn event with note number 60, velocity 64 on MIDI channel 1

/Remote/NoteOff n r c
Send a NoteOff message with the specified note number, release value and channel (channels are specified from 1 to 16). A velocity value of 0 will be treated as if it were a NoteOff message

/Remote/Note n v c d
Convenient way to send a NoteOn message with a duration in milliseconds. The associated NoteOff

message will be triggered automatically after d milliseconds

/Remote/CC n v c
Send a CC message

/Remote/PB v c
Send a pitch bend message

/Remote/AT v c
Send an aftertouch message

/Remote/PT n v c
Send a polypressure message

/Remote/Midi <int> [<int> [<int>]]
This message may have between 1 and three integer arguments representing the explicit bytes of the various MIDI channel messages. This OSC message is typically for special purpose use, perhaps with algorithmic note generators.

Plugins

You can associate OSC handles with plugins and then set or receive OSC messages to control or query a plugin. Each plugin in your rackspace must have a unique OSC handle (you CAN reuse the same OSC handle in another rackspace though)

You can get a list of all plugins that support OSC by sending the following OSC message into Gig Performer

/Rackspace/ListOSCPlugins

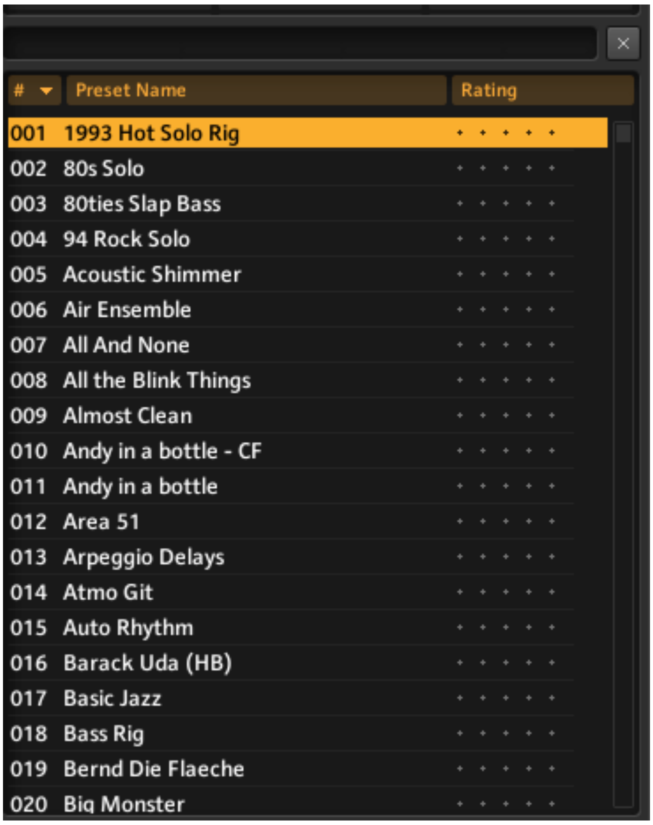
You will receive a list of OSC messages back consisting of a header, the actual names and a terminator

/PluginListStart

  /PluginName <string>
  /PluginName <string>
  /PluginName <string>

/PluginListEnd

1



#	Preset Name	Rating
001	1993 Hot Solo Rig	• • • • •
002	80s Solo	• • • • •
003	80ties Slap Bass	• • • • •
004	94 Rock Solo	• • • • •
005	Acoustic Shimmer	• • • • •
006	Air Ensemble	• • • • •
007	All And None	• • • • •
008	All the Blink Things	• • • • •
009	Almost Clean	• • • • •
010	Andy in a bottle - CF	• • • • •
011	Andy in a bottle	• • • • •
012	Area 51	• • • • •
013	Arpeggio Delays	• • • • •
014	Atmo Git	• • • • •
015	Auto Rhythm	• • • • •
016	Barack Uda (HB)	• • • • •
017	Basic Jazz	• • • • •
018	Bass Rig	• • • • •
019	Bernd Die Flaeche	• • • • •
020	Big Monster	• • • • •

If you have given a plugin the name K1 then the following OSC messages are available

/K1/GetParamCount
Requests the total number of parameters in K1

/K1/SetParam <int x> <float f>
Set 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 host, the above OSC message will be sent out automatically as well

/K1/GetParam <int x>

Explicitly request the value of parameter with index x. The result will be /K1/SetParam x f   i.e, you get back the index you requested along with its value

Widgets

/RackSpace/ListOSCWidgets
Send this message to Gig Performer to receive a list of all OSC widgets in the current rackspace

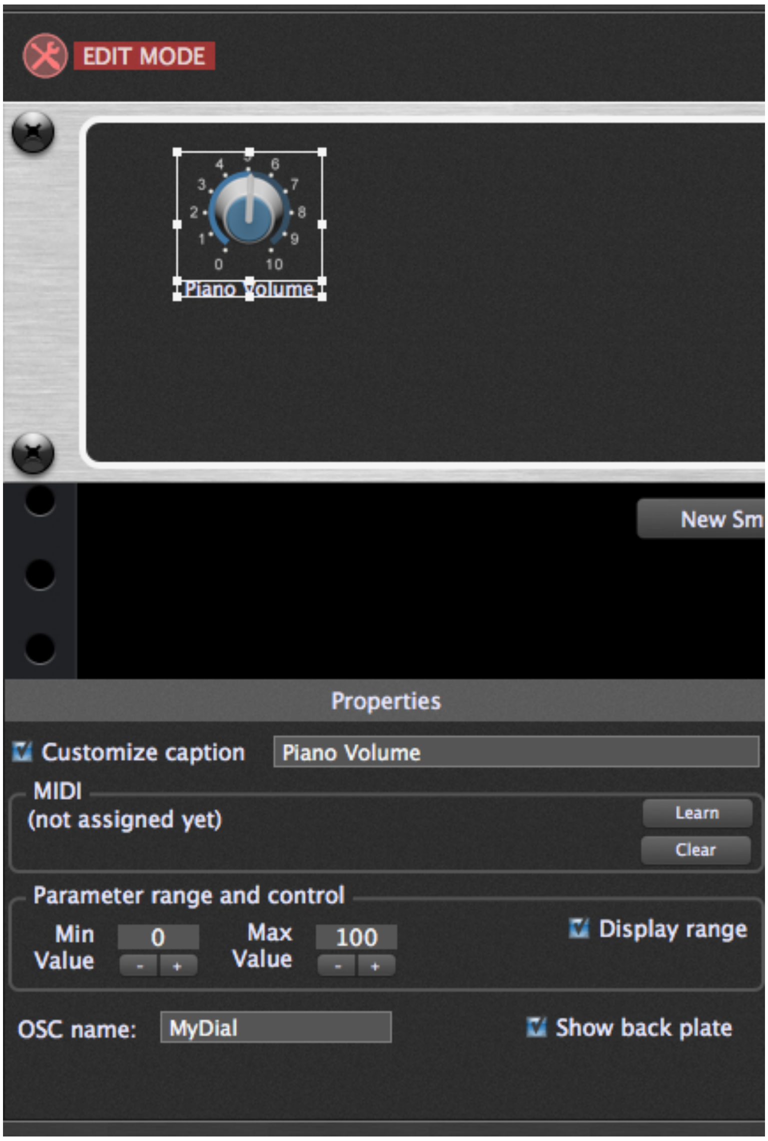
You will receive a list of OSC messages back consisting of a header, the actual widget names, their types (knob, slider, meter, etc) and a terminator

/WidgetListStart

/Widget <string> <string>
/Widget <string> <string>
/Widget <string> <string>

/WidgetListEnd

1 Controlling widgets with OSC



☑ Customize caption: Piano Volume

MIDI: (not assigned yet)

Parameter range and control: Min Value 0, Max Value 100, ☑ Display range

OSC name: MyDial ☑ Show back plate

You can associate an OSC handle with any widget and then control the value from OSC using SetValue and a parameter between 0.0 and 1.0
E.g.
/MyDial/SetValue 0.5

Sets the position of the knob with OSC handle "MyDial" half way

Sending the message above to a label widget controls the brightness of the label

You can request the current value of a widget using the message

/MyDial/GetValue

However, if you are listening to incoming messages, you will receive messages automatically when a dial is changed and you can of course cache that information rather than explicitly asking for a value

You can also set the caption of a widget (including the text of a label) using the following message with a string parameter

/MyDial/SetCaption "abc"

Sets the caption of MyDial to 'abc'

Switching rackspaces and variations

Switch to another rackspace. You can specify the rackspace by its index (the first one will always be 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>]

Change variation. The first variation is always 0
/RackSpace/SetVariation <int>

Move up - Moves to the previous variation or rackspace
/GigPerformer/MoveUp

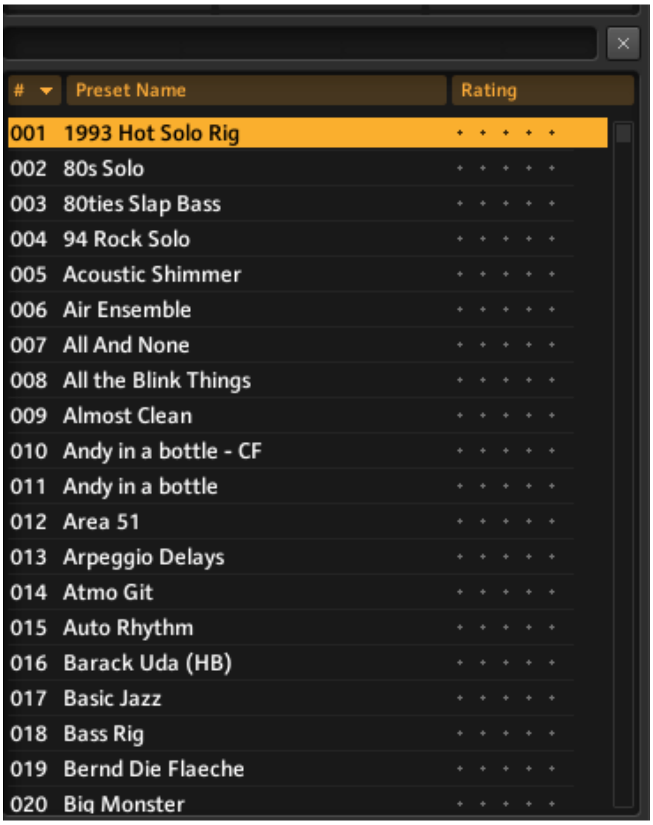
Move down - Moves to the next variation or rackspace
/GigPerformer/MoveDown

Gig Performer will send out OSC information when the rackspace or variation is changed. This can be used to keep a remote control surface in sync.
As well as 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 labels will be sent out.

The rackspace and variation will be sent out using the following OSC message
/RackspaceName string

You can also send the message **/Refresh** into Gig Performer. This is useful if you're using a wireless connection and/or UDP packets are occasionally getting lost. This message will cause Gig Performer to resend the entire state again.

Whenever Gig Performer sends out the entire state, it will first send the message
/ClearAll
which can be used by Lemur, for example, to clear all the previous values (you will need to create a script to do that)



#	Preset Name	Rating
001	1993 Hot Solo Rig	
002	80s Solo	
003	80ties Slap Bass	
004	94 Rock Solo	
005	Acoustic Shimmer	
006	Air Ensemble	
007	All And None	
008	All the Blink Things	
009	Almost Clean	
010	Andy in a bottle - CF	
011	Andy in a bottle	
012	Area 51	
013	Arpeggio Delays	
014	Atmo Git	
015	Auto Rhythm	
016	Barack Uda (HB)	
017	Basic Jazz	
018	Bass Rig	
019	Bernd Die Flaeche	
020	Big Monster	

If your plugin supports presets (such as GuitarRig from Native Instruments, in this image) then you can send an OSC message to set the desired preset.
If the OSC handle of a Guitar Rig instance is GR then the following will set Guitar Rig to the first preset (presets start at 0)

```
/GR/SetPreset 0
```

Tap Tempo

The simple OSC message

```
/Tap
```

can be sent to GigPerformer which will measure the rate at which the messages arrive, using that information to set the global tempo, which can be referenced by various plugins that sync to the current BPM

The OSC message

```
/SetBPM <value>
```

can be sent to set the current BPM

E.g.,

```
/SetBPM 120
```

Tuner control

You can switch to the tuner view via an OSC command.

Send the message

/TunerView/SetValue 1.0

from your touch surface.

To exit the tuner view send the message
/TunerView/SetValue 0.0

When you are in Tuner view (and OSC is enabled), Gig Performer will send OSC messages representing the current note name and the offset.

/TunerNoteName <string>
/TunerCentsOffset/SetValue <float>

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

Tips and tricks

1) Hold the shift key down while moving a widget for more fine-grained control
2) Hold the shift key down and touch the output pins of one plugin to the input pins of another to make a quick stereo connection. If there is already a connection, it will be removed instead.
3) Many more to come

Keyboard shortcuts
Keyboard modifiers

Keyboard shortcuts

As you get more comfortable with Gig Performer you will find it very convenient to use keyboard commands for more rapid access to certain tasks

File management
CMD-O: Open a new gig
CMD-S: Save current gig
SHIFT-CMD-S: Save current gig with new name

SHIFT-CMD-I: import rack space from a file
SHIFT-CMD-E: Export the current rackspace to a file

Window management
CMD-T : cycle between the performer view and the connections view
CMD-G: toggle widget edit view

Rackspace management
SHIFT-CMD-N: Add a new rackspace
CMD-D: Duplicate the current rackspace
SHIFT-CMD-BACKSPACE: Delete the current rackspace

CMD-N: Add new variation to current rackspace
CMD-BACKSPACE: Delete the current variation

Connections management
CMD-P: Insert a new plugin in the connections view (Gig Performer will switch to the connections view automatically if necessary)

CMD-, : Show preferences

Scripting (experimental, not available in all versions of Gig Performer
CMD-R: reload script

Keyboard modifiers

In certain cases, if the Shift key and/or the Alt key are being pressed during some operation, the behavior of that operation will be changed

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. This is mostly for testing purposes to save time but we mention it here in case you wonder why your last gig file didn't get loaded

Dragging plugins:
If you hold the Shift key down and you drag one plugin slightly over another so that audio pins "touch", a stereo connection will be made automatically, or if the two plugins are already connected, the stereo connection will be broken.

Customizing parameter labels for widgets:
Normally the widget caption will be assigned the parameter name only. 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.

Toggling channels in the Midi In plugin
You can enable/disable individual MIDI channel by clicking on their respective checkboxes. However, if you hold the Shift key down while clicking a checkbox, 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

Glossary of terms

1) Plugin architecture: a collection of plugins connected together in a specific way

2) Connections view: the area where you create the desired plugin architecture

2) Performer view: the area containing knobs, sliders and buttons that you use to control your sounds during performance

3) Variation: the combination of settings of all knobs, sliders and buttons associated with a single plugin architecture. You can have multiple variations

4) Rackspace: The combination of a plugin architecture and all its variations

5) Widget: a generic term for a knob, slider or button that is used to control plugin parameters from a front panel or remotely from a keyboard or control surface

Appendix - 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 SOON 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 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, 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.