Overview of the Novation SL-MK3 Extension

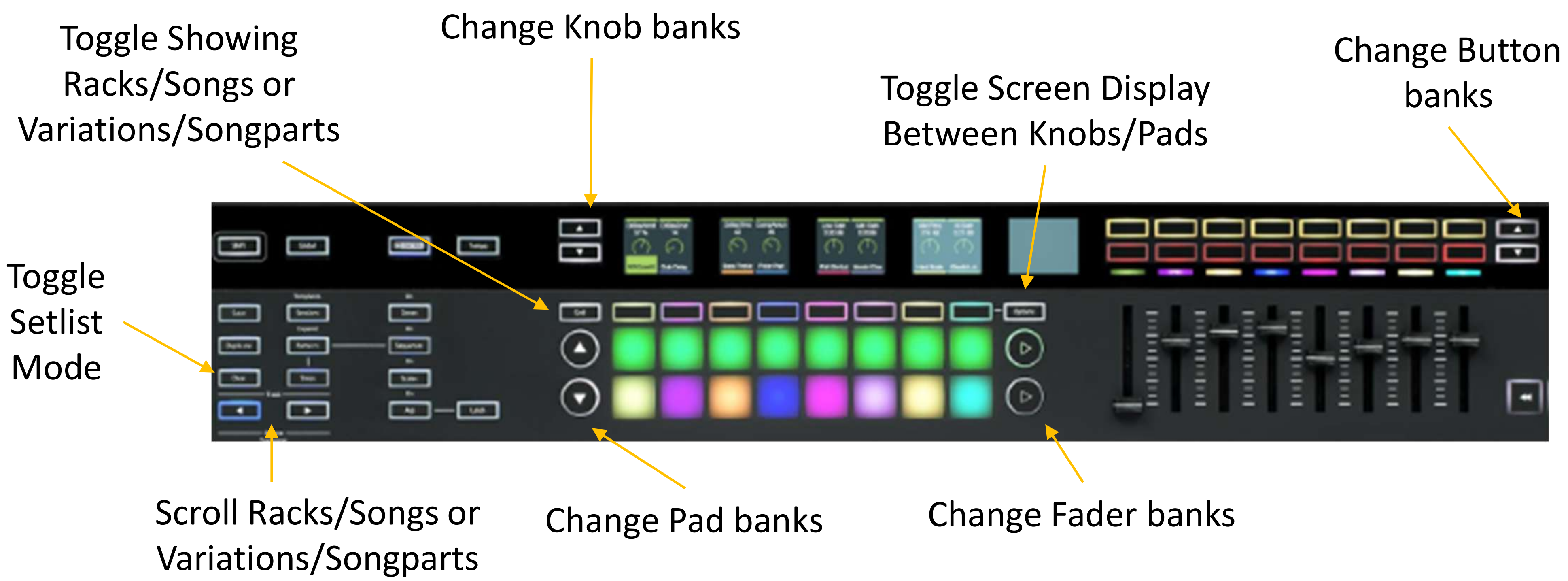
This extension works with the SL-MK3 by taking control of the displays, buttons, and pads when "InControl" mode is enabled using the "InControl" button on the SL-MK3.

Control Organization

The SL-MK3 extension works with control surface items in groups by type: knobs, buttons, pads, and faders.

The extension is built around the concept of "banks" of each control group. You can have multiple knob banks, pad banks, etc.

Next to each row of controls there are up/down arrows, which the extension can utilize to "bank switch" through as many banks of such controls as you care to have in your Rackspaces.


Toggle Showing Racks/Songs or Variations/Songparts
Change Knob banks
Change Button banks
Toggle Screen Display Between Knobs/Pads
Toggle Setlist Mode
Scroll Racks/Songs or Variations/Songparts
Change Pad banks
Change Fader banks

Getting Started with the SL-MK3 Extension

Follow these simple steps to get started:

- Download the zip file for your platform (Mac or Windows) from [Releases · WidnerM/GP-SL-MK3 (github.com)]
- From the zip file install the extension (.dylib for Mac, .dll for Windows) to the appropriate GigPerformer folder
    - /Users/Shared/Gig Performer/Extensions on Mac
    - C:/Users/Public/Documents/Gig Performer/Extensions on Windows
- Load the Demo gigfile from the zip file in GigPerformer
- Make sure your controller is in InControl mode by pressing the InControl button
- You may need to exit and restart GigPerformer after the first time you enable the extension

Setting the MIDI Ports

The extension should automatically find your controller's MIDI ports using their default names. If it does not, you can tell the extension which MIDI ports to use by creating two text widgets in the global rackspace and setting them appropriately.

These widgets must be named "sl_midiin" and "sl_midiout" (using the Advanced tab in the "OSC/GPScript Name" field).
Set the Caption for these widgets (on the General tab) to the MIDI port names exactly as they appear in the GP MIDI ports list.

Basic Operations and Troubleshooting

Without any widgets configured you should see your Rackspaces / Variations / Songs / Songparts listed on the bottom row of the display. You can switch between them using the Grid and Clear keys and cycle through using the arrows toward the lower left of the keyboard.

The naming conventions for other widgets is the main subject of the remainder of this document.

To sync your Rackspace with the SL-MK3 after adding new control widgets you should change Rackspaces, then come back.
This is necessary because the extension scans widget names upon Rackspace entry. It doesn't know you renamed widgets until you change Rackspaces and return.

Controlling GigPerformer with the SL-MK3 Extension

Configuration of how the SL-MK3 interacts with GigPerformer is done through widgets.

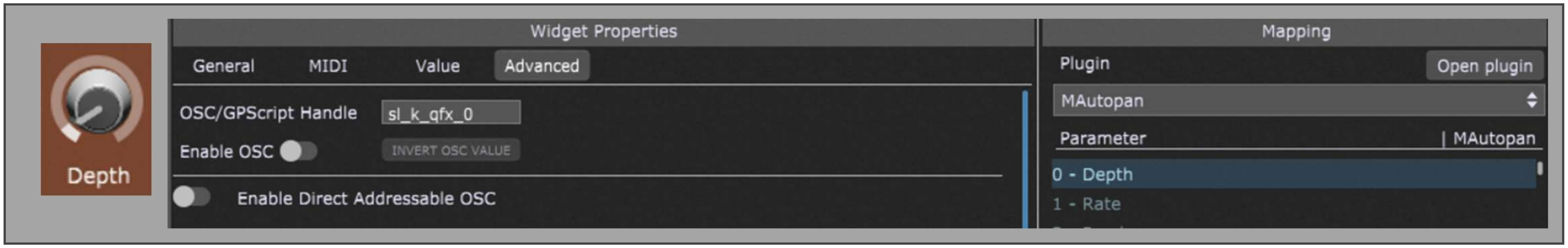
On the "Advanced" tab of each GigPerformer widget there is a field called "OSC/GPScript Name". Any widget with a name that begins with "sl_" will be examined by the extension for information about how it should be utilized by the extension.

Every button, pad, display section, and LED on the SL-MK3 can be programmed with RGB colors. Because only so much information can be attached to a knob widget, for example, the extension will also look for information on "parameter widgets" to control how they are displayed on the SL-MK3.

Widgets utilized by this extension fall into three broad categories:

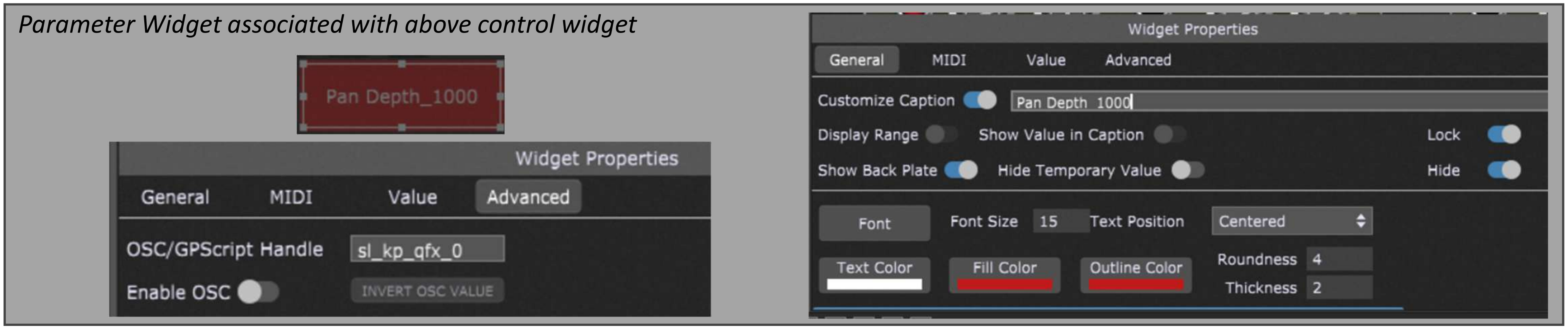
- Control Widgets – knobs, button, etc. widgets typically linked to plugin parameters or system functions
- Parameter Widgets – used primarily to determine how Control Widgets are displayed on the SL-MK3 (e.g., colors)
- Indicator Widgets – provide feedback in GP showing which banks of Control Widgets the SL-MK3 is actively controlling

Control Widget example



OSC/GPScript Handle: sl_k_qfx_0
Enable OSC
Enable Direct Addressable OSC
Plugin: MAutopan
Parameter | MAutopan
0 - Depth
1 - Rate

Parameter Widget associated with above control widget



OSC/GPScript Handle: sl_kp_qfx_0
Enable OSC

Customize Caption: Pan Depth_1000
Display Range
Show Value in Caption
Lock
Show Back Plate
Hide Temporary Value
Hide
Font Size: 15
Text Position: Centered
Roundness: 4
Thickness: 2

Control Widgets

“Control Widgets” are typically knobs, faders, buttons, etc. linked in GP to plugin parameters. To be utilized by the SL-MK3 extension the GPScript Name must conform to the following format:

sl_type_bankID_position

“sl” identifies it as a widget of interest to the SL-MK3 extension

“type” indicates which row of SL-MK3 controls the widget will be associated with. The choices are:

- k = knobs
- p = pads
- b = buttons
- f = faders

“bankID” is any arbitrary name for this group of widgets that is meaningful to you. Typical examples:

- “Pan” for panning controls of a mixer plugin
- “solomute” if you wanted two RGB button rows to control solo/mute on a mixer plugin

“position” must be in the range of 0 – 15 to indicate which control position the widget is linked to

Knobs and faders are numbered 0-7, buttons and pads 0-15, arranged left to right

For example, a bank of eight knob widgets you want to use to control panning in a mixer could be named:

sl_k_pan_0   sl_k_pan_1   sl_k_pan_2   sl_k_pan_3   . . .   sl_k_pan_7

The bankID “pan” in this example is entirely arbitrary. I generally use bankID’s that are descriptive, but you could just as easily name your banks things like “bank1” or “xyz” if you’re so inclined.

The purpose of this bankID is so that you can select among multiple banks. You can define as many banks as you want of each control type.

The up/down arrows next to the control groups on the SL-MK3 will scroll through banks in bankID alphabetical order.

Parameter Widgets

"Parameter Widgets" can be used to specify how "Control Widgets" appear on the SL-MK3. Parameter Widgets are not strictly necessary, but without them everything will appear using default colors and labels on the SL-MK3.

Parameter Widgets associated with buttons and pads control the RGB "on" and "off" colors on the SL-MK3. Parameter Widgets for knobs control the knob color, the bar color above the knob, and optionally the resolution of the knobs.

Parameter Widgets can be created at the bank level (for controlling colors of the entire bank) or at the individual widget level.

Parameter Widgets use the same general naming format as Control Widgets, but append a "p" to the "type" field.
    e.g., Control Widget "sl_k_pan_0" can have an associated Parameter Widget "sl_k**p**_pan_0"

Parameter Widgets intended to operate at the bank level append "p" to the "type" and drop the "_position" part of the name
    e.g., Parameter Widget "sl_kp_pan" would apply to all of the knobs with BankID "pan"

Individual parameter widgets take priority over bank parameter widgets if both exist
    e.g., widget "sl_kp_pan" will control the default knob color for the "pan" knob bank, but widget "sl_kp_pan_3" can be used to override that color for that knob

Parameter Widgets should generally be created as Text Widgets in GigPerformer, and often should be hidden.

The information the SL-MK3 extension looks for in Parameter Widgets is:

- Caption – sets the label that will appear on the SL-MK3 display for the widget
- Fill Color – controls the "on" color of buttons/pads or the color of knobs on the display
- Outline Color – controls the "off" color of buttons/pads or the top bar color above knobs
- Knob resolution – for knob widgets you can change the resolution by appending an "_" after the label in the Caption field followed by an integer. e.g., a Caption of "Volume_200" would result in a knob label of "Volume" and a resolution of 200 "ticks" to move the widget from 0.0 to 1.0. The default resolution is 1000.
- Pad & Button Caption - format is "Label_OffText_OnText" with optional "_m" appended make it momentary contact
- Note that the alpha (transparency) of the Fill and Outline color is not used by the extension. Setting the Fill and Outline alphas to zero in GigPerfomer is an alternative to hiding these widgets. This can be useful if you want to see the label on the Rackspace but not the color. This can also be very confusing when you're trying to remember where you set the color, so I usually hide them instead.

Indicator Widgets

“Indicator Widgets” are similar to Parameter Widgets, but instead of controlling the colors of widgets on the SL-MK3 they are used to show on the Rackspace screen which widget banks are actively being controlled by the SL-MK3.

For example, if you have three separate banks of knobs in a Rackspace (e.g., instrument parameters, reverb parameters, and pans) it can be helpful to see on the GigPerformer screen which ones are actively linked to the knobs on the SL-MK3. You can change banks using the up/down arrow keys next to the control row on the SL-MK3 and the Indicator Widgets will reflect which bank is currently active

Indicator Widgets are named with the same format as Control and Parameter widgets but with “_i” in place of the _position.
e.g., sl_k_pan_i

Indicator Widgets are generally created as Text widgets, most often with the text itself being blank or with the Text Color alpha set to zero so that the text itself does not appear on the Rackspace screen.

When the bank specified is “active” the extension will set the value of the widget to 1, which will raise its visibility on the Rackspace screen. When the bank is not active the value will be set to 0.3, which will reduce its visibility. A common use of these widgets is as an outline and background behind the associated set of Control Widgets.

The Caption of Indicator Widgets for knob and button banks controls the text that appears in the label areas on the right-most display on the SL-MK3, or temporarily in the Notify area when bank switching pad banks. Text for knob and button banks appears on two lines, which should be separated in the Caption by the “_” character. e.g., a Caption of “Solo_Mute” for a button bank would show the label “Solo” next to the top row of buttons, and “Mute” next to the bottom row.

Note – GigPerformer will remember Indicator Widget values when switching between Rackspaces/Variations/etc. As a result, it will remember which banks you were controlling when you last used or saved a Variation, and when you return to it those same banks will be “active” again on the SL-MK3.

Note – the background color of an Indicator Widget is used for coloring the up/down bank select arrows on the SL-MK3. If you make each bank a different color, the up/down arrows will indicate which bank is next/previous as you bank select.

BankID Linking

If the same bankID is used for different control rows (e.g., Faders and Knobs) then when that bankID is selected to be active for one control row (e.g., Knobs or Faders) it will be selected for all rows that have a bankID of that name.

For example, if you want to be able to bank select between three separate 8 channel mixers plugins and have the entire control surface move together between them you would use Widget names like:

sl_f_mix1_[0..7]	sl_k_mix1_[0..7]	sl_p_mix1_[0..15]	sl_b_mix1_[0..15]
sl_f_mix2_[0..7]	sl_k_mix2_[0..7]	sl_p_mix2_[0..15]	sl_b_mix2_[0..15]
sl_f_mix3_[0..7]	sl_k_mix3_[0..7]	sl_p_mix3_[0..15]	sl_b_mix3_[0..15]

Named as such, when you bank select using the Knob bank select keys you will automatically also select the corresponding set of Faders, Pads, and Buttons.

In contrast, if you want to use the same three mixer plugins and be able to independently select which mixer plugin the Knobs, Faders, and Pads are actively controlling you must use different bankIDs for each widget row. A simple example would be:

sl_f_volume1_[0..7]	sl_k_pan1_[0..7]	sl_b_mute1_[0…7]
sl_f_volume2_[0..7]	sl_k_pan2_[0..7]	sl_b_mute2_[0…7]
sl_f_volume3_[0..7]	sl_k_pan3_[0..7]	sl_b_mute3_[0…7]

I sometimes use a mix of both approaches in the same Rackspace.

Additional Controls

The buttons immediately below the display are always assigned to select Rackspaces, Variations, Songs, or Songparts. You can scroll through the them using the left and right arrow keys toward the bottom left of the control surface.

Other button assignments are:
- “Play” – starts the playhead
- “Stop” – stops the playhead
- “Clear” – toggles in and out of Setlist mode
- “Grid” – toggles between displaying Rackspaces/Variation or Songs/Songparts if you are in Setlist Mode
- “Option” – toggle the SL-MK3 displays between showing Knobs and showing Pad assignments.

Fader Controls

The faders on the SL-MK3 are not motorized, which usually means that any time you change a Rackspace, Variation, Song, or Songpart the physical position of the faders will most likely not match the widgets they are linked to.

The SL-MK3 extension uses a “catch” approach before establishing a link between an individual fader and the widget that it is controlling. The LED above each fader indicates the status of each fader. The color codes are:
- Unlit – following Rackspace/Variation/Song/Songpart change it is unknown if the fader is in sync with the widget
- Green – the fader on the SL-MK3 is materially higher than the value of the widget
- Red – the fader is too low relative to the value of the widget
- Other – the light changes to the fader bank color on the Indicator or Parameter widget when fader is in sync

Fader bank switching is done using the round Scene buttons between the pads and faders.

Key Lights

The SL MK3 key lights can be lit to show zone assignments. Zones and their color are defined with widgets conforming to the same standard as other widgets.

The format is:

sl_zone_zoneID_0 and sl_zone_zoneID_1

The widget values map to the first and last MIDI notes of the zone. The bottom note on the 61SL-MKII is 36, which corresponds to a widget value of 28.4. You can calculate this as [note number / 127] * 100.

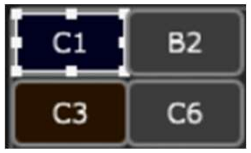
The fill color of the “0” widget will be used as the RGB zone color. You can create as many “zoneID” banks as you want. Each bank will be displayed with its own color.

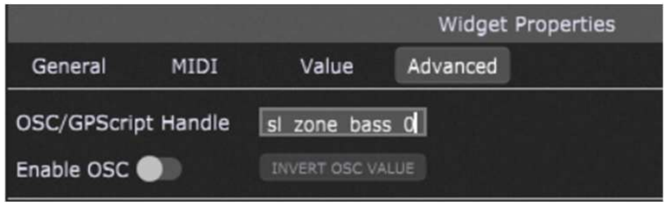
If these widgets are linked to GP’s Midi In blocks the “0” widget should be linked to the Min Note parameter and the “1” to the Max Note parameter.

If two zones overlap the color of the key LED will be the mathematical sum of the two colors. These may not look the way you expect or want.

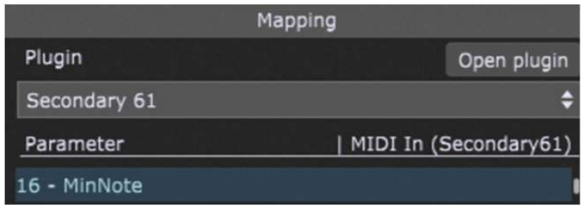
- Tip: if you look at the hexadecimal values of the three color channels (RGB) in the color picker, keep each of the channels below 0x80 and overlapping colors should look more like you’d expect.
- Tip: if you want complete control over the color of overlapping zones you can un-link your zone widgets from the MIDI in blocks, or create separate MIDI in blocks and widgets for any overlap portions.

Note: the key light for the highest key on the 61SL-MK3 cannot be lit through this interface. This is a limitation of the Novation “In Control” protocol.





OSC/GPScript Handle: sl zone bass 0



Plugin: Secondary 61

Parameter | MIDI In (Secondary61)

16 - MinNote

Additional Comments

The only Transport buttons integrated into the extension are the Play and Stop buttons. The others are not used.

You can open the Script Logger window in GP and see some level of debugging detail that might help if you encounter odd behavior. Probably not, though, because I try to keep it to a minimum in the released versions.

Configuring the details of how things are displayed can take a lot of extra widgets. I usually hide these in my rackspaces, which means they're only visible in Edit mode. I tend to put them near (or even right on top of) the widgets they're associated with.

At times I've considered storing the configuration data in a separate file rather than extra widgets. I always conclude that this would make things less reliable, more confusing, more difficult for the user, and require more user effort to keep files in sync.