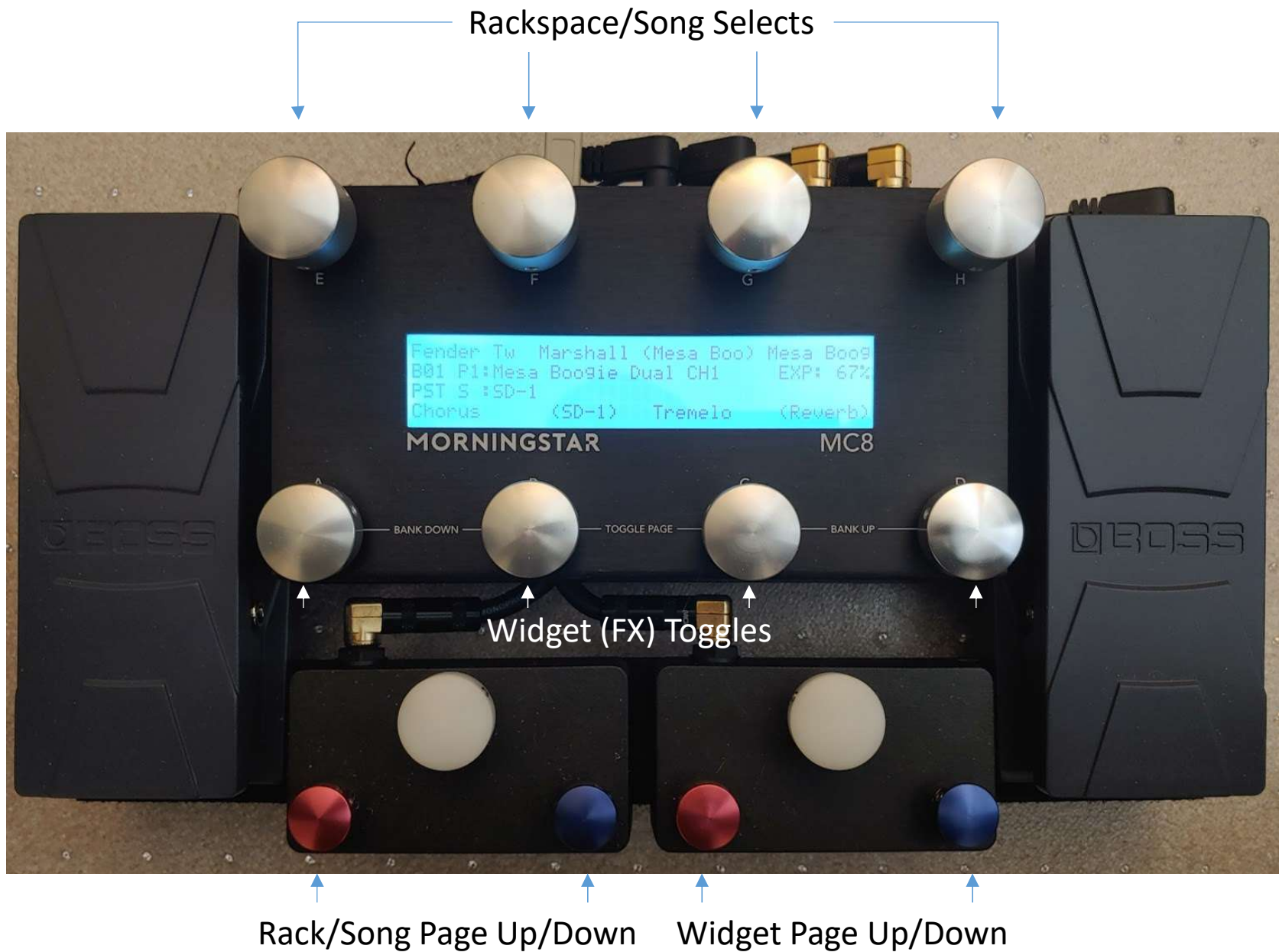


The Morningstar MC8 GigPerformer Extension

A Typical Setup



Design Goal of the MC8 Extension

The goal for the extension was to make controlling GigPerformer with the MC8 as simple, reliable, and flexible as possible while keeping all configuration entirely within GigPerformer. Specifically, to allow the MC8 to automatically stay in sync with any changes made in Rackspaces, Variations, Songs, Setlists, etc.

This extension requires setting one bank on the MC8 to communicate with this GigPerformer extension. I use bank 30, but any bank can be used for this purpose. When controlling GigPerformer with this extension the MC8 will always be operating with that single bank.

With that bank the user will be able to assign any number of toggle widgets to MC8 foot switches, switch among Rackspaces and Variations, or Songs and Songparts, and the MC8 display will always reflect which button is controlling what, and what its present state is.

This extension utilizes the MC8 as a simple button and display interface. The MC8 will display what the extension tells it to display, and the MC8 will send the same messages whenever a specific button is pressed. All of the logic for controlling the displays and reacting to foot presses resides in the extension, not in the MC8. The bank on the MC8 that is used to control GigPerformer will never change, no matter how many racks, songs, variations, or widgets you choose to control with it.

This means you can dedicate one “bank” on the MC8 to GigPerformer and continue to use all other banks as you see fit.

Controlling GigPerformer with the MC8 Extension

All configuration of how GigPerformer interacts with the MC8 is done through widgets in GigPerformer.

This document is divided into two major parts:

- Conceptual relationship between the physical displays and buttons on the MC8 and the logical structure of the extension
- Configuring widgets to produce the results you are looking for.

Overview of the MC8 Extension

The Morningstar MC8 can be viewed as a control surface having two rows of four foot switches, and display sections corresponding to those switches. We will refer to these as the Top row and the Bottom row.

Each control row can be cycled between showing widgets, Rackspace selection, Variation selection, Song selection, or Songpart selection. The example at the start of this document showed Rackspace selection active on the top row and Widget toggles on the bottom row.

A short press of a button will select what is currently assigned to that button. In the case of widgets, it will toggle them.

Long presses of specific buttons can be used to page backward and forward through whatever is shown on a row (e.g., through rackspaces or songs), or cycle through the options for what is shown on the row (e.g., switching from widgets to racks to variations).

The MC8 is structured with two Pages in each bank. If you choose to make use of this you can use long presses to switch between pages and have different items assigned to widgets on different pages. For example, you could have page one set to display widgets on both the top and bottom rows, and page two display Songs on the top row and Songparts on the bottom row.

Pedals and Extra Switches

The four Omniports on the MC8 can be used for expression pedals or additional foot switches (Aux switches). By default Omniports 3 and 4 are set up for expression pedals, which can be linked to widgets through this extension. Omniports 1 and 2 are set up to accommodate six additional buttons to control paging through what is displayed on the top and bottom rows.

Using additional buttons to control paging is a convenience and reliability feature. The same functionality is available whether you choose to use additional buttons through the Omniports or not. The only difference is whether you trigger these functions through long presses of the MC8's buttons, or regular presses on the Aux buttons. I prefer the additional buttons.

The next few pages illustrate the control layouts for using the MC8 without extra buttons, and with extra buttons.

A single switch press will toggle or select whatever is assigned to (and displayed near) that switch. This can be anything assigned to a GigPerformer widget (most often FX toggles), or to select between Rackspaces/Songs or Variations/Songparts depending on whether GigPerformer is in Setlist Mode. (e.g., if you are showing Rackspaces and go into setlist mode the row will change to Songs.)

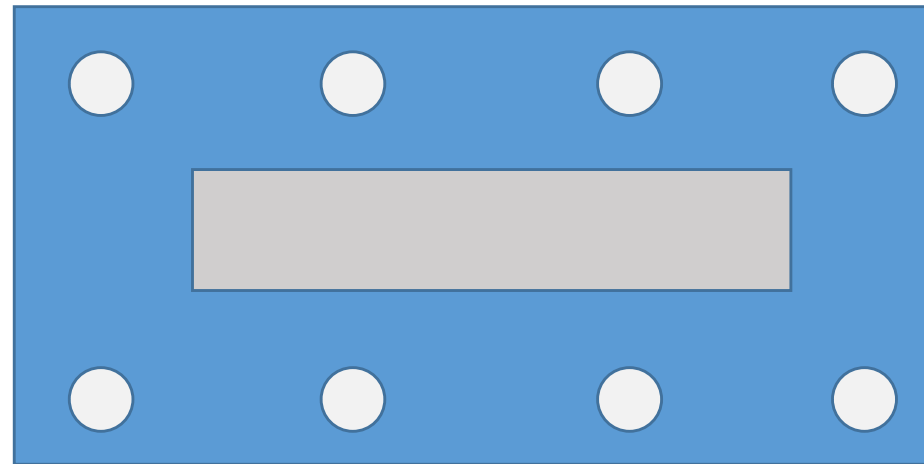
Any number of Widgets may be designated to be linked to a row of switches. This is done in banks of up to four widgets. If, for example, you have 16 Widgets you want to be able to toggle in one Rackspace you could assign those in four banks of four. You can choose to have all four of those banks assigned to the same row (requiring you to bank switch among them to access them all) or with one bank on each row (e.g., top row and bottom row of page one, top and bottom row of page two). How they get arranged on the MC8 will depend on how you name the widgets in GigPerformer.

The following diagram shows how Long Press actions change what is shown on the bottom row. The Long Press functions are the same for the top row, except the Page select. Page select is a toggle that is always on the third button of the bottom row.

Long Press actions for controlling the extension

Bank Down:

Shifts to previous group of what is displayed on this row. e.g., prior four Widgets, Racks/Songs, or Variations/Songparts →



Bank Up:

Shifts to next group of what is displayed on this row. e.g., next four Widgets, Racks/Songs, or Variations/Songparts ←

Select:

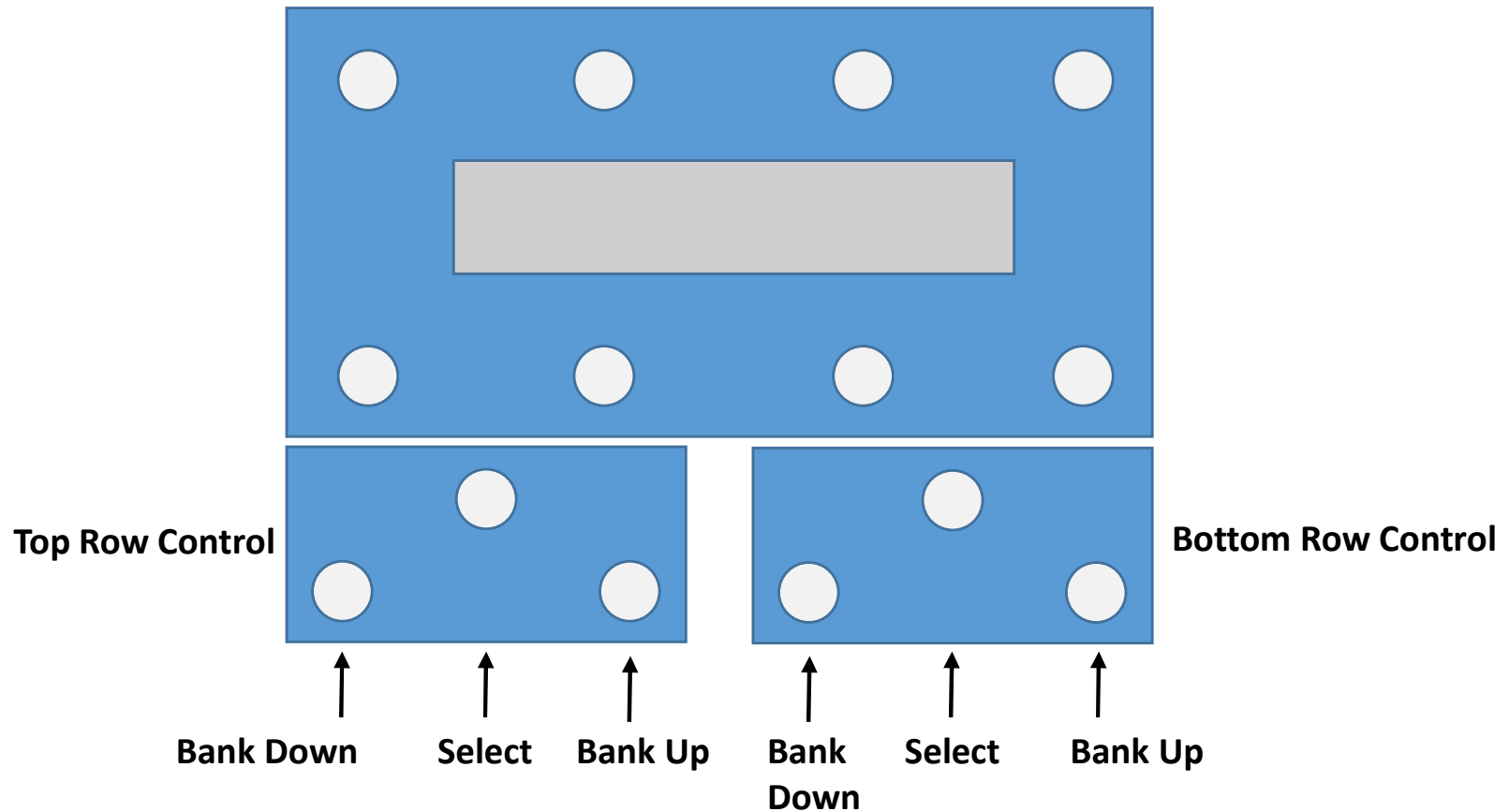
Rotates what is active on row between Widgets, Racks/Songs, Or Variations/Songparts

Page:

Moves to Page 2 of the bank, or back to Page 1

I generally use two additional sets of three buttons connected through MC8 Omniports 1 and 2.

The diagram below illustrates how these are assigned. They replicate the Long Press functions without requiring a long press for those of us who believe life is too short for Long Presses.



Widget Configuration

The MC8 extension identifies GigPerformer widgets the user would like to use by their OSC/GPScript Name as set in the Advanced tab of the widget editor.

To be controlled by the MC8 extension the GPScript Name must conform to the following format:

mcx_rowID_bankID_[position]

“mcx” identifies it as a widget of interest to the MC8 extension

“row ID” indicates which row of buttons and display this widget will appear on. The choices are:

- t = top row
- b = bottom row
- t2 = top row of page 2
- b2 = bottom row page 2
- e3 = expression pedal
- e4 = expression pedal

“bank ID” is any arbitrary name for this group of widgets that is meaningful to you. Typical examples would be:

- fx = general effect toggles
- fx2 = a second bank of fx toggles
- eq = toggles for different bands on a parametric eq
- sys = widgets assigned to GP functions like Play/Stop, Tap Tempo, etc.

“position” must be in the range of 0 – 3 to indicate which switch/display position the widget will appear in. Zero is the leftmost position, 3 is the rightmost

A typical bank of four widgets to appear on the bottom row of page one could be:

mcx_b_fx_0 mcx_b_fx_1 mcx_b_fx_2 mcx_b_fx_3

A second bank of four widgets to appear on the same row would be:

mcx_b_fx2_0 mcx_b_fx2_1 mcx_b_fx2_2 mcx_b_fx2_3

The bankIDs can be rotated through in alphabetical order using the “bank up” and “bank down” controls.

Special Configuration Widgets

Several “configuration” widget groups are available to control how widgets appear on the display. These include:

- “Parameter” widgets, which control the names that appear on the MC8 display when values are on or off
- “Indicator” widgets, which can be used to remember which widget banks are active on the MC8 screen when rackspaces, or variations are changed
- “row_configuration” widgets, which tell the extension what functions you want to appear on each of the four MC8 display rows by default (the four rows being “b, t, b2, and t2” as explained previously)

Parameter widgets are optional and correspond 1:1 with the basic mcx widgets previously described. The naming format is identical but with the addition of a “p” character appended to the row_ID.

- Our prior example illustrated basic widgets named as: mcx_b_fx_0 mcx_b_fx_1 mcx_b_fx_2 mcx_b_fx_3
- The corresponding “p” widgets would be named: mcx_bp_fx_0 mcx_bp_fx_1 mcx_bp_fx_2 mcx_bp_fx_3
- If a widget with the appropriate GP Script Name is found it will control what appears on the MC8 display based on the text of the widget Caption as set by the user on the General tab of the widget editor.
- The format of the Caption will be interpreted as “Off name_On name_Long name”
- A typical example would be “Chorus_(Chorus)_Boss CE-2 Chorus”
- This will the name “Chorus” on the MC8 when the widget is off and “(Chorus)” when on. The “Long name” part of the caption is not presently used and may be excluded
- These widget are usually hidden in GP for aesthetic reasons, using the “Hide” option on the General tab in edit mode
- In the absence of a “p” widget the MC8 will display the Caption of the underlying widget when the widget is off, and place a * in front of the same when it is on.
- For example, if the mcx_b_fx_0 widget had a caption of “Chorus” this would appear on the MC8 display as “Chorus” when the widget was toggled off, and “*Chorus” when toggled on
- Note that allowing GP to change the widget caption based on the widget value or temporary values communicated by the underlying VST may produce unreliable results on the MC8 display due to timing issues around reading widget values, particularly during Rackspace and Variation changes

Indicator widgets are optional and correspond to widget banks. They use an “i” in place of the position indicator.

- Our prior example illustrated basic widgets named: mcx_b_fx_0 mcx_b_fx_1 mcx_b_fx_2 mcx_b_fx_3
- The corresponding “i” widget would be named: mcx_bp_fx_i
- Indicator widgets have two purposes: visually indicating on the GP Rackspace screen which banks of widgets are currently displayed on the MC8, and controlling which are active when a Rackspace or variation is changed.
- The MC8 extension will set this widget to a value of 1 when the indicated bank is actively being displayed on the MC8, and to a value of 0.3 when it is not.
- In our prior example we showed two banks of effects control widgets. These were
 - mcx_b_fx_0 mcx_b_fx_1 mcx_b_fx_2 mcx_b_fx_3
 - mcx_b_fx2_0 mcx_b_fx2_1 mcx_b_fx2_2 mcx_b_fx2_3
- In the absence of any “i” widgets the MC8 extension will display the group with the bankID that comes first in alphabetical order. In this case the bank “fx” comes before “fx2” and bank “fx” would be displayed whenever the Rackspace is entered.
- If the Rackspace also contains widgets named “mcx_b_fx_i” and “mcx_b_fx2_i” then the extension can remember which bank you prefer to have selected when you enter a Rackspace or Variation.
- These widgets are often hidden for aesthetic reasons, unless the user has a specific desire to reflect these values on the main Gig Performer screen.
- These are usually created as Text widgets. Because the extension switches the “Value” of these widgets between 0.3 and 1.0 depending on whether they are being shown on the MC8, the apparent brightness of the widgets will change on the GP screen.
- I tend to use Text widgets instead of Shape widgets for this because the Text widget can do everything a Shape widget does, plus some additional features (such as text) than can be hidden (via transparency in the color selector) if not utilized
- See the example Rackspace in the example Gigfile to see one way these can be utilized visually

Row Configuration widgets are optional and come in two types that correspond to their required names:

- `mcx_initial_row_config` = control over what functions appear on each row when a Gig file is loaded
- `mcx_row_config` = same as above but may appear in multiple Rackspace to configure them differently
- Each of these widgets must have a GP Script Name as indicated above and a caption as follows:
 - The Caption is split into four parts with “_” delimiters between them
 - Each segment must contain one of: “racks”, “variations”, or “buttons”
 - Racks and Variations will be displayed when not in Setlist mode and will automatically switch to Songs and Songparts when GP is put into Setlist mode
 - The four positions correspond to what are effectively the four display rows and button banks on the MC8
 - The order is `b_t_b2_t2` corresponding to bottom and top rows of Page 1 followed by page 2
 - My usual configuration is: `buttons_variations_racks_buttons` which is also the default if no row configuration widget is specified
- If a `mcx_row_config` widget is present in a Rackspace the MC8 will reflect that configuration upon Rackspace entry
- The user can actively change what is shown on each row at any time using the Select footswitch buttons
- If the Rackspace is then changed to a different Rackspace that does not have a `mcx_row_config` widget the MC8 will retain the current row configuration
- Note that if an `mcx_row_config` widget exists in the global Rackspace the extension will pick that up upon every Rackspace change
- The `mcx_initial_row_config` widget exists for users that want the MC8 switch rows to start with something other than the defaults at Gig loading, but not reset to that default every time a Rackspace is changed. For this reason this configuration widget should be placed in the Global Rackspace if it is used
- Personally, I do not use this widget. I prefer to include an `mcx_row_config` widget in each Rackspace instead

Expresion Pedals

Expression pedals in the four Omniports can be assigned to widgets using the same general naming convention as the button widgets.

The format is:

mcx_e1_bankID_0 mcx_e2_bankID_0 mcx_e3_bankID_0 mcx_e4_bankID_0

I put my expression pedals on ports 3 & 4. This is how the MC8 template bank is constructed. Unless you want more than two expression pedals connected, I suggest using ports 3 & 4 unless you want to manually change the MC8 bank template. If you are well versed in using the MC8 to control other hardware then you should not have any difficulty editing the template.

If you want to use more than two expression pedals you will have to edit the MC8 bank template. By default the extension expect buttons on Omniports 1 and 2, which is why the expression pedals are normally on 3 and 4.

This extension currently provides no bank switching mechanism for expression pedals. If you create more than one bank for any of the four expression pedals there is no way to access anything except the first bank alphabetically. I have not had a practical need for it, so have not made an effort to implement expression pedal bank switching.