# The Morningstar MC8/MC6 GigPerformer Extension

A Typical Setup

## Getting Started with the MCx Extension

Follow these simple steps to get started:

- Download the zip file for your platform (Mac or Windows) from Releases · WidnerM/GP-MC8 (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
- Install the appropriate bank preset from the zip file into any open bank your controller using the Morningstar editor
- Load the Demo gigfile from the zip file in GigPerformer

All behavior of the extension is controlled by widgets in GigPerformer. This can get rather extensive, as the rest of this document describes. It may be easier to start simple before trying to make things too complex.

## Important Configuration Widgets

The extension needs to know how to communicate with your controller. It looks for this information in three text widgets that should be created in the Global Rackspace.

These widgets are named (on the Advanced tab in the "OSC/GPScript Name" field) as follows:

- "mcx_midiin" – the midi in port exactly as it appears on your system and under the Options -> MIDI Ports menu item
- "mcx_midiout" – as above for the out port
- "mcx_layout" – the type of device you have as either "mc8", "mc6", or "mc6 pro"

The extension should find your controller's MIDI ports by default, but if it does not you should set those widgets.

As long as the mcx_layout widget alread exists in the Global Rackspace you can change it through the GP menus by selecting Extensions -> MC8 Extension -> [your device type].

A widget named "mcx_initial_row_config" is optional but suggested. It will control what appears on the top and bottom rows of your controller by default. This is described in more detail on page 10 of this document.

## Basic Toggle Controls

If you're starting from a new Rackspace (i.e., not one from the demo gigfile) assigning button-type widgets to your footswitches is very simple. All you have to do is edit the properties of the widget.

To assign widgets to the switches on the bottom row:

- Edit the widget and select the "Advanced" tab
- In the "OSC/GPScript Name" field type "mcx_b_fx_0"
- This will assign the widget to the bottom left switch
- Change the 0 to 1, 2, or 3 for the other bottom row switches

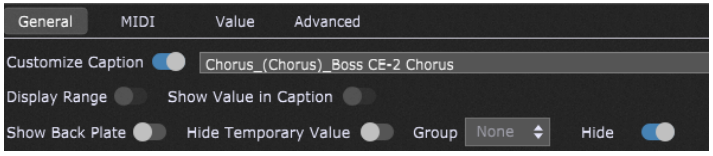

***To sync your Rackspace with the foot controller you must change Rackspaces, then come back***. This is necessary because the extension scans widget names upon Rackspace entry. It doesn't know you renamed widgets until you change Rackspaces.

The name that will display for each widget on the controller is (by default) the widget's caption

- If you let the underlying plugin change the caption it may not sync properly with the controller display. This is because GigPerformer runs audio and scripting in different threads, so we can't know which will execute first

For full control of what appears on the controller display when the widget is on and off you can use separate text widgets

- The OSC/GPScript names for the text widgets parallel the button widgets but with a "p" added to the name
- e.g., "mcx_bp_fx_0" will control the label for the bottom left switch
- Use a Customized Caption in the format "OffName_OnName_LongName"
- I generally hide these widgets by selecting the "Hide" option on the General tab of the text widget


Assigning widgets to the top row is the same, except the first part of the name must be "mcx_t". (i.e., t=top, b=bottom)

- If you didn't change the "mcx_initial_row_config" widget then the top row will select variations by default
- To change what is being displayed on the top row, long-press the second button on the top row to rotate between Rackspaces, Variations, and widgets

## Design Goal of the MCx Extension

The goal for the extension was to make controlling GigPerformer with the MC8 simple, reliable, and flexible 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. It has been adapted to work with the MC6 and MC6 Pro. Collectively we'll refer to these as MCx.

This extension requires setting one bank on the MCx to communicate with this GigPerformer extension. Any bank can be used for this purpose. When controlling GigPerformer with this extension the MCx will always be operating with that single bank. To minimize risk of unintentionally overwriting your other banks, your MCx should be set to this GigPerformer bank before starting GigPerformer using this extension.

With this GP bank the user will be able to assign any number of toggle widgets to MCx foot switches, switch among Rackspaces and Variations, or Songs and Songparts, and the MCx display will stay in sync to show which button is controlling what and what its present state is.

This extension utilizes the MCx as a simple button and display device. The MCx will display what the extension tells it to display, and the MCx will send the same messages whenever a apecific button is pressed. All of the logic for controlling the displays and reacting to foot presses resides in the extension, not in the MCx. The bank on the MCx 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 MCx to GigPerformer and continue to use all other banks as you see fit.

## Controlling GigPerformer with the MCx Extension

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

This document is divided into two major parts:

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

## Overview of the MCx 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. On the MC6 there are three buttons per row, not four, but otherwise everything works the same way.

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 on the top row and Widget toggles on the bottom row.

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

Long presses of different 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 the controller effectively has four rows of buttons: a top and bottom row in each page. Long presses of specific buttons can switch between pages, and you can 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 MCx can be used for expression pedals or additional foot switches (Aux switches). The extension looks for extra switches on Omniports 3 and 4 with the MC6 Pro, and Omniports 1 and 2 on the MC8. The other two Omniports can be linked to knob/fader/pedal widgets through this extension.

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.

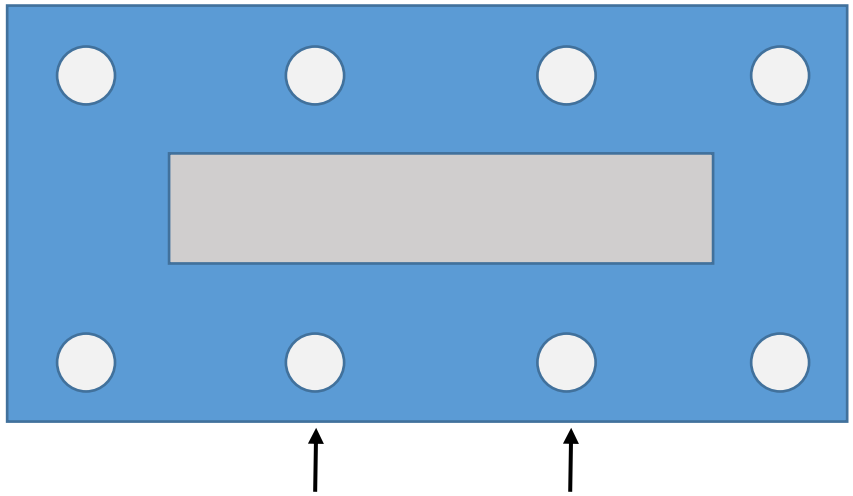
A single switch press will toggle or select whatever is assigned to (and displayed near) that switch. This can be anything assigned to a GigPerformer widget (e.g., FX toggles), or to select between Rackspaces/Songs or Variations/Songparts depending on whether GigPerformer is in Setlist Mode. (e.g., if 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

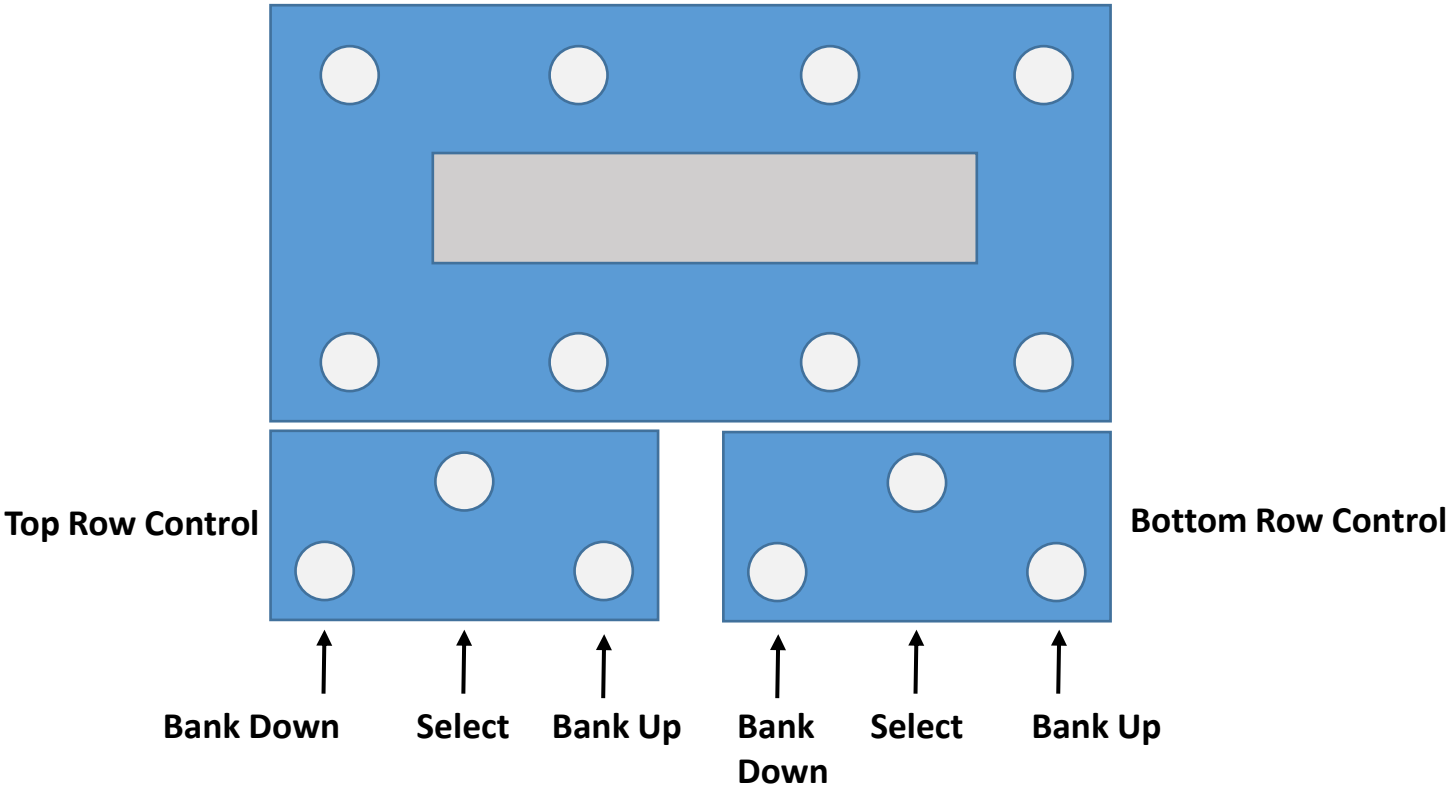
**Select**:
Rotates what is active on row between Widgets, Racks/Songs, Or Variations/Songparts

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

***Note: on the MC6 Pro the "Page" buttons effectively don't exist***

I generally use two additional sets of three buttons connected through MC8 Omniports 1 and 2 set up as Aux. On the MC6 Pro you must use Omniports 3 and 4 set up as Aux switches.

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




***NOTE*** *– the term "Bank" in this context refers to banks of GigPerformer widgets, songs, or racks as described in the next section. It has nothing to do with Banks as defined on the MCx. The MCx extension will always operate within a single Bank as defined on the MCx.*

## 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 MCx extension the GPScript Name must conform to the following format:


"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

- e1 = expression pedal
- e2 = expression pedal
- 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

MC6 units will only display and control widgets with a position of 0 – 2

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 Rackspaces 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 Rackspaces 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 MCx 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 on every Rackspace change

- The 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

Note that if you have Aux switches connected to on Omniport then you can't also have an expression pedal on the same Omniport. The extension does not know what you actually have connected to the MCx, so use the names as appropriate for the Omniports your expression pedals are actively connected to.

**This extension currently provides no individual BankID switching mechanism for expression pedals**. If you create more than one bank for any of the four expression pedals the extension will initially attach to the first BankID in alphabetical order.

It is possible to have the same expression pedal control different things at different times within the same Rackspace by using the same BankIDs as widgets assigned the button rows, but this will only be controlled using "Bank Synchronization" as described in the next section.

If you want your expression pedals to always control the same thing in a Rackspace (e.g., master volume) then you should use a unique BankID for that.

## Bank Syncronization

Often it is preferable to have fully independent control of what is on the bottom and top button rows, but other times it is preferable to have the two linked. Linking the two rows tends to be convenient when selecting Songs or Rackspace changes, as it allows you to page between 8 (for the MC8) or 6 (for the MC6) options at a time rather than half that number.

For Racks, Songs, Variations, and Songparts it works like this:

- If both the top and bottom Rows are toggled to show the same thing (e.g., Racks, Songs) then those rows will automatically become temporarily linked if you page forward or back through them
- You can use long-presses or Aux switches to page forward on either row, and both rows will advance
- If you manually select something different to show on either row they will become de-linked and each row will resume being paged independently
- Note that if you change Rackspaces (including by changing songs) and the new rackspace contains a "mcx_row_config" widget the row and button assignments will automatically change on rackspace activation

For Widgets

- Each time you change which widgets are shown on a row (cycling through them by BankID) the extension will automatically look for widgets using that same BankID on other rows. If it finds any, it will switch that row to them
- If you never want this to happen, then don't use the same BankIDs on your different rows.
- e.g., if you have widgets "mcx_t_fx_0..3" and "mcx_b_fx_0..3" then if both rows are currently showing widgets, and you page either row to the "fx" bank then both rows will switch to the "fx" bank
- This includes expression pedals, which offer no other mechanism to switch banks. If you want your expression pedals always controlling the same thing in a Rackspace, give them unique BankIDs

Forced Sync

- Using Aux switches, generally the middle button is assigned to toggle what is displayed on a row (e.g., widgets, Racks, Variations). A long press of the middle button will force both rows to whatever is being toggled to
- A typical use of this would be when a Song is over and another is to be chosen you can long-press either row select button to get into displaying songs on both rows quickly. If you have a "mcx_row_config" widget on the Rackspace you switch to then the MCx will automatically go back to displaying your preferred controls after song selection
- If you don't use Aux switches this can be done using double-presses of the row select buttons

## Other Notes

**Pages** – the MC8 contains two pages, the MC6 Pro hardware contains four. This extension utilizes only two pages, and the "Page" functions will toggle between them.

**MC6 Pro Colors** –
Colors on the MC6 Pro are not fully controllable through the MCx extension at this time. The Morningstar developers expect this ability to be added soon, at which point the extension will be adjusted to allow widgets to control display colors on a preset basis.

Currently the extension has the ability to control the center bar color in a limited way. It is set up now so that the center bar becomes orange when GigPerformer is in Setlist mode and purple when it is not. Presets on page one are green, page two is red. This can be changed by editing the MC6 Pro preset. You should be able to figure this out and change them pretty quickly using the Morningstar Editor. I'm not going into detailed instructions at this point because I expect it all to change in the next version or two of the MC6 Pro firmware.

**MC8 Pro** – Morningstar currently expect an MC8 Pro to be released late in 2023, but that is speculative at this time.