

MMcL Mackie Control Surface for Cakewalk by BandLab

History

The MMcL Mackie Control Surface is an enhancement to the MackieControl.dll shipped with Cakewalk by BandLab.

When Cakewalk Inc. initially developed support for hardware surfaces for SONAR, they made the Cakewalk Control Surface API available to the public so that hardware manufacturers could develop support for their hardware for use within SONAR.

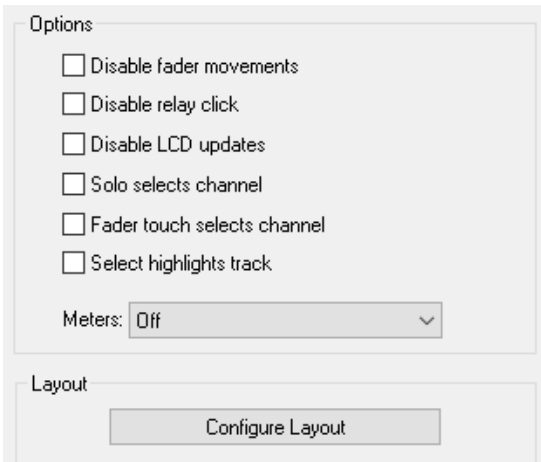
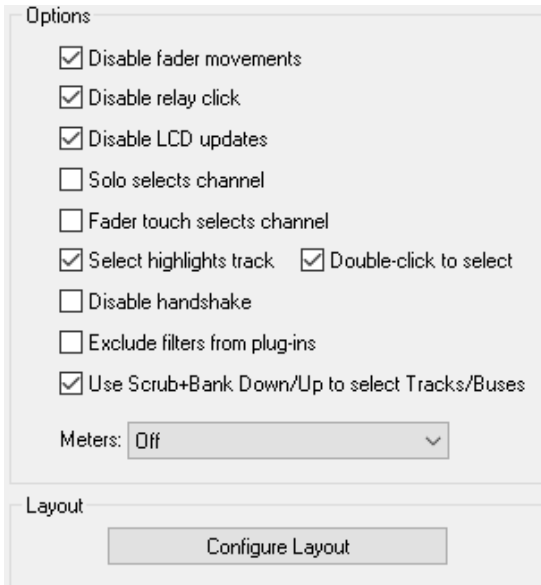
The original API can be found on GitHub: <https://github.com/Cakewalk/Cakewalk-Control-Surface-SDK>

Azslow3 was the first to make enhancements to the API with his own fork on GitHub: <https://github.com/Azslow3/Cakewalk-Control-Surface-SDK>. Azslow3 added support for the ProChannel EQ/Compressor, some fixes for the C4, and greater control over the initial handshake.

Msmcleod has taken Azslow3's changes and added further enhancements. Msmcleod's version of the API can be found on GitHub: <https://github.com/msmcleod/Cakewalk-Control-Surface-SDK>

Enhancements

The MMcL Mackie Control surface has some new options:

Original Cakewalk MackieControl Surface	MMcL MackieControl Surface
	

Double-click to select

This option is aimed at users of the Korg nanoKONTROL Studio control surface, and is used in conjunction with the “Select highlights track” option.

Although the nanoKONTROL’s track select buttons operate properly when the “Select highlights track” option is checked, the highlighted track is also changed every time a fader is moved. This is because the Korg nanoKONTROL sends a “select track” message not only when a track select button is pressed, but also when a track’s fader is moved.

When “Double-click to select” is checked, fader movements no longer change the highlighted track, however the user can “double click” the select button (i.e. press it twice) to change the highlighted track.

Disable Handshake*

This option disables the initial handshake with the hardware control surface when Cakewalk/SONAR loads the control surface.

Earlier SONAR would hang if the control surfaces weren’t plugged in at the time of the initial handshake, or the hardware in question didn’t fully support the MCU handshaking protocol. Enabling this option prevents this behaviour.

**Originally developed by Azslow3*

Exclude filters from plug-ins*

By default, support is enabled for the ProChannel EQ, Track Compressor and Bus Compressor, when using the EQ or Dynamics buttons on the MCU or C4.

Checking this option disables this support.

**Originally developed by Azslow3*

Use Scrub+Bank Down/Up to select Tracks/Buses

This option is aimed at users of the Korg nanoKONTROL Studio.

The Korg nanoKONTROL has no buttons to switch between controlling Tracks and Buses, and is set to control Tracks by default.

Enabling this option gives the following behaviour:

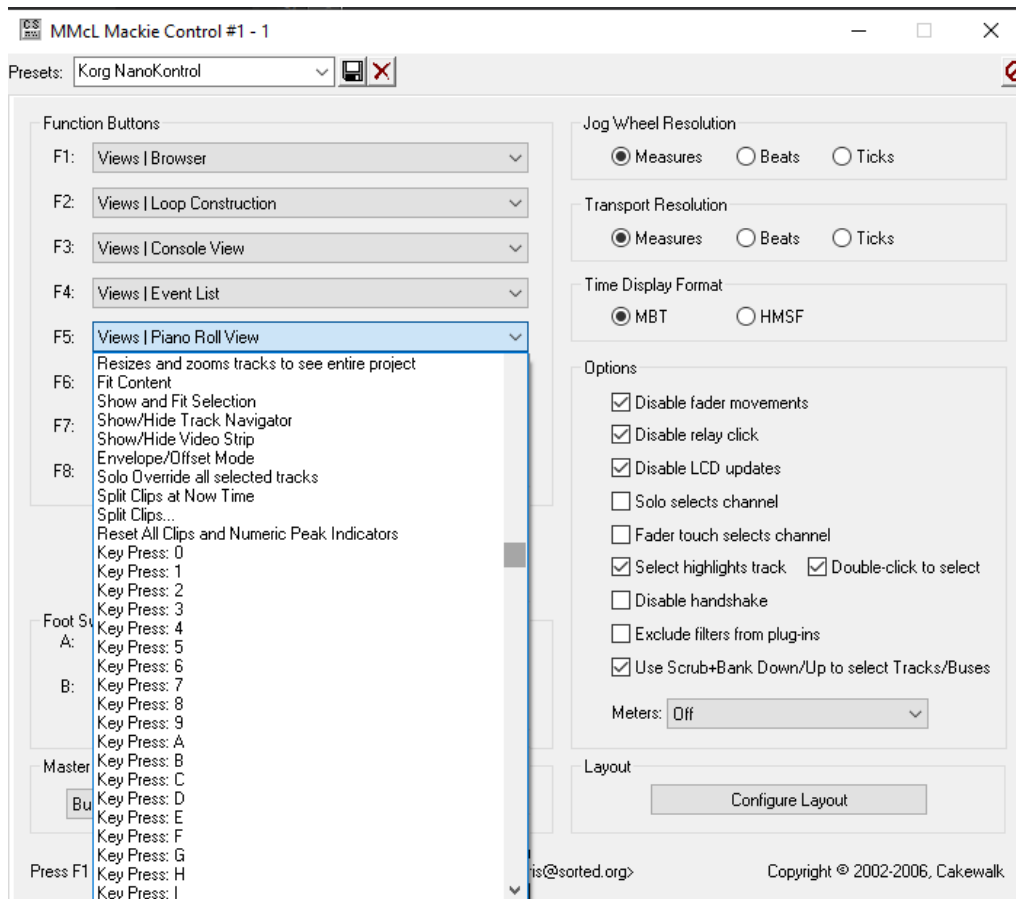
Scene button + Bank Down button : Control Tracks

Scene button + Bank Up button : Control Buses

NOTE: For this to work, the Korg nanoKONTROL must be configured to use the Scene button as the Scrub button, by enabling the “Use Scene Button as Scrub” Global Parameter, using the KORG KONTROL Editor software.

Key Press support

In addition to the standard list of commands, the MCU function buttons and foot switches can now be configured to send a key press to Cakewalk:



When used in conjunction with the Key Bindings within Cakewalk, this can greatly enhance the functionality of the function buttons over the standard list of commands, e.g. by configuring an MCU function button to run a CAL script.

Additional Button Mapping:

F1/CUT, F2/COPY and F3/PASTE buttons now have extra modifier behaviour (in red below):

		M1	M2	M3 (beta)	M4
F1/CUT	F1	Cut	Cut Special		Reload Plugin Mappings
F2/COPY	F2	Copy	Copy Special	Export track template & save as __mcu-temp-track-template__.cwx	Copy Special & press OK
F3/PASTE	F3	Paste	Paste Special	Imports template __mcu-temp-track-template__.cwx	Paste Special & press OK

WARNING: Saving / Importing the track templates using the M3 modifier will clear the current clipboard.

Additional modifiers can now be used with the Save and Tracks buttons:

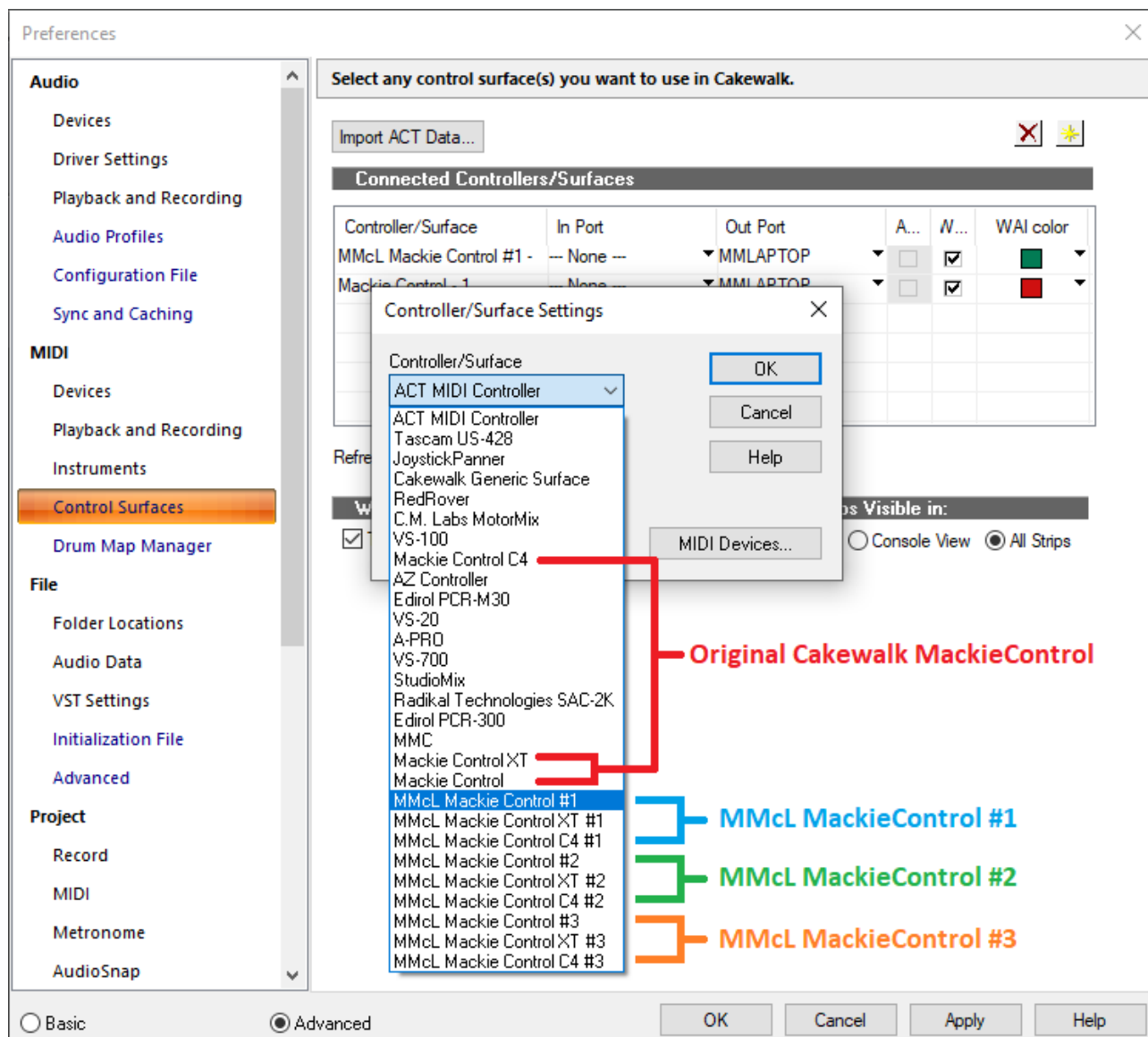
		M1	M2	M3	M4
Save	Save	Save As	Save Copy As	Export Audio	
Track	Control Tracks	Export Track Template	Import Track Template	Bounce to Tracks	Bounce to Clip(s)

The following behaviour was added for the Korg NanoKONTROL:

Cycle + Rewind	Go to Start
Cycle + Record	Undo

Multiple Surface Support

Up to three independent instances of the MMcL Mackie Control surface are now available, so that multiple control surfaces can be used to independently control the same instance of Cakewalk:



This can be useful if:







- You want one control surface to control your tracks, whilst the other controls your buses
- You have another control surface away from your main desk (e.g. in a vocal booth) and wish to control your tracks/buses and/or transport from there.

NOTE: Both the original Cakewalk Mackie Control surface and the MMcL Mackie Control surfaces share the same settings within `MackieControl.ini`.

Installer / Uninstall Support

The MMcL Mackie Control now comes as an installer, which automatically installs and registers the new control surfaces. It is no longer necessary to copy over the original Cakewalk MackieControl.dll, or run regsvr32. The new control surfaces will now live alongside the original Cakewalk version.

Once installed, the following files are added to your Control Surface directory in addition to this guide (normally [C:\Program Files\Cakewalk\Shared Surfaces](#)):

 UninstallMMcLMackieControl.exe	06/04/2019 04:57	Application	116 KB
 UninstallMMcLMackieControl.ini	06/04/2019 04:57	Configuration sett...	3 KB
 MackieControl-MMcL-3.dll	06/04/2019 04:56	Application extens...	245 KB
 MackieControl-MMcL-2.dll	06/04/2019 04:56	Application extens...	245 KB
 MackieControl-MMcL-1.dll	06/04/2019 04:55	Application extens...	245 KB
 UnregisterMMcLMackieControl.bat	05/04/2019 00:43	Windows Batch File	1 KB

To uninstall, click on the uninstall program highlighted above.

Do not delete any of these files manually, as this will cause the control surfaces to look like they are still available, and could cause Cakewalk to crash.

If you do get your system into this state, re-install the MMcL Mackie Control Surface and use the uninstall program highlighted above.

Fader Reset on Close

Motorised faders now return to zero on close of project or when closing Cakewalk.

Editing MackieControl.ini

Note: The following section was taken from an archive of an article written by Chris Boucher (the original developer of the MackieControl.dll). The original article can be found here:

<https://web.archive.org/web/20120625023520/http://www.chrisboucher.com/cakewalk/mackiecontrol/>

The MackieControl.ini file defines how plugin parameters are mapped across the VPots of the control surface. If a plugin is not listed in the file, its parameters are simply mapped in the order the plugin presents them to Cakewalk. By adding a plugin to the ini file, the parameters can be remapped into a consistent order.

In addition to rearranging the parameters, the ini file also allows the data type/display format (for the Halo of LEDs around each VPot) and default values to be defined for each parameter. A plugin can also be identified as an Equaliser or Dynamics processor for quick access via the EQ and DYN buttons on the control surface.

Finally, for Equaliser plugins, the special frequency/gain mode where the faders control level and the VPots control frequency, can be defined.

How To Add a Plugin

Let's say we want to add a new (fictitious) plugin called EqX, a simple two band equaliser. Firstly, add the plugin to a Cakewalk track and press PLUG-INS twice, to enter SINGLE CHANNEL mode where the plugin parameters are mapped across the VPots.

Using the BANK and CHANNEL buttons (in EDIT mode), we find that there are 9 parameters available:

Parameter Number	Parameter Name
0	Bypass
1	Band 1 Enable
2	Band 1 Gain
3	Band 1 Frequency
4	Band 1 Q
5	Band 2 Enable
6	Band 2 Gain
7	Band 2 Frequency
8	Band 2 Q

Note that the parameters are numbered starting at zero.

The MackieControl.ini file is located in the "Cakewalk\Shared Surfaces" directory, in my default install this is "C:\Program Files\Cakewalk\Shared Surfaces". It's a simple text file, so you can edit it using it using notepad.

Before going any further, take a backup of this file, just to be safe

The first section at the top of the file lists the plugins defined in the file, and looks something like:

```
[Plugins]
0=FxChorus
1=FxDelay
2=FxEq
3=FxFlange
4=FxReverb
5=CompressorX (Sonar XL)
6=Equalizer (Sonar XL)
7=Track Eq
8=Bus Eq
9=ProChannel EQ
10=Track Compressor
11=Bus Compressor
```

The first step is to add a new entry for our plugin, so we add the line:

```
12=EqX
```

to the end of the [Plugins] section.

Note: the name of the plugin must exactly match the name as shown in Cakewalk and on the desk (if you press PANS and then PLUG-INS the name will be displayed for a couple of seconds).

Now we need to add a new section, defining how the parameters are to be mapped. Add the line:

```
[EqX]
```

to the end of the file.

To start with, we're just going to reorder the parameters. We'll be using 9 VPots (since there are 9 parameters), so add the line:

```
NumVPots=9
```

And now we'll add the parameters themselves, but change their order:

```
[EqX]
```



```

NumVPots=9

VPot0=3                ; Band 1 Frequency
VPot1=7                ; Band 2 Frequency
VPot2=4                ; Band 1 Q
VPot3=8                ; Band 2 Q
VPot4=2                ; Band 1 Gain
VPot5=6                ; Band 2 Gain
VPot6=1                ; Band 1 Enable
VPot7=5                ; Band 2 Enable
VPot8=0                ; Bypass

```

Everything after the ; is simply a comment. Like parameters, the VPots are numbered starting at zero.

So now VPot0 (the first VPot) is set to control parameter 3, which in this case is Band 1 Frequency.

Pressing M4-F1 on the Mackie causes the control surface module to reload the ini file and rebind the VPots - very useful for checking that everything is reordered correctly.

We now need to define what type of data each VPot/parameter controls - this information is used to control the Halo of LEDs around the VPots and what happens when a VPot is pressed. The options are: **level**, **pan**, **freq**, **switch**, **boost/cut** and **spread**.

The data type is added after the parameter number, separated by a comma:

[EqX]

```

NumVPots=9

VPot0=3,freq           ; Band 1 Frequency
VPot1=7,freq           ; Band 2 Frequency
VPot2=4,spread         ; Band 1 Q
VPot3=8,spread         ; Band 2 Q
VPot4=2,boost/cut      ; Band 1 Gain
VPot5=6,boost/cut      ; Band 2 Gain
VPot6=1,switch         ; Band 1 Enable
VPot7=5,switch         ; Band 2 Enable
VPot8=0,switch         ; Bypass

```

For data types other than **switch**, a default value can also be defined. This is the value assigned to the parameter when the VPot is pressed. Note: the value must be in the range 0.0 - 1.0:

[EqX]

```

NumVPots=9
VPot0=3,freq,0.25      ; Band 1 Frequency
VPot1=7,freq,0.6       ; Band 2 Frequency
VPot2=4,spread,0.2     ; Band 1 Q
VPot3=8,spread,0.2     ; Band 2 Q
VPot4=2,boost/cut,0.5  ; Band 1 Gain
VPot5=6,boost/cut,0.5  ; Band 2 Gain
VPot6=1,switch         ; Band 1 Enable
VPot7=5,switch         ; Band 2 Enable
VPot8=0,switch         ; Bypass

```

The **switch** data type doesn't have a default value, instead pressing the VPot toggles the parameter on and off.

As you turn a VPot, the parameters are adjusted in steps of 0.05. For some parameters this may be too coarse so, finally, the step size for each parameter may be defined. Here I've reduced the step size for the frequency controls to 0.02:

```

[EqX]
NumVPots=9
VPot0=3,freq,0.25,0.02 ; Band 1 Frequency
VPot1=7,freq,0.6,0.02  ; Band 2 Frequency
VPot2=4,spread,0.2     ; Band 1 Q
VPot3=8,spread,0.2     ; Band 2 Q
VPot4=2,boost/cut,0.5  ; Band 1 Gain
VPot5=6,boost/cut,0.5  ; Band 2 Gain
VPot6=1,switch         ; Band 1 Enable
VPot7=5,switch         ; Band 2 Enable
VPot8=0,switch         ; Bypass

```

Before going further, we should also define the type of plugin. There are three options 0 (everything else), 1 (equaliser) and 2 (dynamics). If a plugin is type 1, then it can be accessed by pressing the EQ button. Similarly, type 2 plugins can be accessed via the DYN button.

EqX is an equaliser, so:

```

[EqX]
PluginType=1
NumVPots=9
VPot0=3,freq,0.25,0.02 ; Band 1 Frequency

```

```
VPot1=7,freq,0.6,0.02 ; Band 2 Frequency
VPot2=4,spread,0.2 ; Band 1 Q
VPot3=8,spread,0.2 ; Band 2 Q
VPot4=2,boost/cut,0.5 ; Band 1 Gain
VPot5=6,boost/cut,0.5 ; Band 2 Gain
VPot6=1,switch ; Band 1 Enable
VPot7=5,switch ; Band 2 Enable
VPot8=0,switch ; Bypass
```

You can also rebind the VPots for each of the four MODIFIERS (M1 - M4). For equalisers the convention I've used so far is that if the unmodified VPot controls frequency, then pressing M2 makes it control the corresponding Q, M3 controls Gain and M4 controls the Enable:

[EqX]

```
PluginType=1
NumVPots=9
VPot0=3,freq,0.25,0.02 ; Band 1 Frequency
VPot1=7,freq,0.6,0.02 ; Band 2 Frequency
VPot2=4,spread,0.2 ; Band 1 Q
VPot3=8,spread,0.2 ; Band 2 Q
VPot4=2,boost/cut,0.5 ; Band 1 Gain
VPot5=6,boost/cut,0.5 ; Band 2 Gain
VPot6=1,switch ; Band 1 Enable
VPot7=5,switch ; Band 2 Enable
VPot8=0,switch ; Bypass
M2VPot0=4,spread,0.2 ; Band 1 Q
M2VPot1=8,spread,0.2 ; Band 2 Q
M3VPot0=2,boost/cut,0.5 ; Band 1 Gain
M3VPot1=6,boost/cut,0.5 ; Band 2 Gain
M4VPot0=1,switch ; Band 1 Enable
M4VPot1=5,switch ; Band 2 Enable
```

You can rebind M1 too, but as it's normally used to enter fine adjustment mode, it's generally better if you don't.