

FCB1010 Manager



Version 2.1.0 for Windows
Version 2.0.0 for macOS

MANUAL

Copyright © 2009-2018 by Mark van den Berg

MOUNTAIN UTILITIES



<https://mountainutilities.eu/>

CONTENTS

| | |
|--|--------------------|
| 1. Overview | 3 |
| 2. Version history | 4 |
| 3. Computer requirements | 7 |
| 4. Installation of FCB1010 Manager | 8 |
| 5. MIDI setup | 10 |
| 6. The main window | 14 |
| 7. The MIDI input messages window | 26 |
| 8. Using the computer keyboard | 31 |
| 9. Known problems | 32 |

1. Overview

FCB1010 Manager is a utility by Mountain Utilities for setting up Behringer's FCB1010 MIDI Foot Controller. It is available for macOS and Windows. It is free, although donations are more than welcome.

Its main features are:

- Support for FCB1010 firmware versions 2.4 and 2.5.
- No support for Ossandust's 'UnO' firmware. (This might change in the future — but don't hold your breath.)
- Easy up/downloading of FCB1010 data.
- A preset table with cut/paste facilities.
- Allows preset and bank descriptions.
- An integrated virtual FCB1010 emulating the real FCB1010's output behavior. Very useful for testing.
- Various windows for MIDI input and output testing.

2. Version history

Version 2.1.0 (2018-06-14; Windows edition only):

- The instructions in the ‘Receive data’ and ‘Send data’ dialog boxes have been expanded.
- The exe file of the installer is no longer offered within a zip file, but directly.
- The installer automatically installs the edition of the application that matches the operating system: the 32-bit edition on 32-bit Windows, the 64-bit edition on 64-bit Windows. (Previously there was only a 32-bit edition, which was installed on 32- and 64-bit Windows alike.)
- Portable 32- and 64-bit editions are available. The 32-bit edition runs on 32- and 64-bit Windows, the 64-bit edition only on 64-bit Windows.
- The FCB1010 Manager update mechanism allows you to specify in which update types you are interested: alpha (development) versions, (‘public’) beta versions, release candidates and release versions.
- Since the Mountain Utilities website has changed from the HTTP protocol to HTTPS, all links to the Mountain Utilities website have changed to HTTPS too.
- In accordance with Apple renaming OS X to macOS, all occurrences of ‘OS X’ in this manual have been replaced with ‘macOS’.
- CopperLan has been added to the list of MIDI pipes in §5. (Thanks to JAPP for bringing CopperLan to my attention.)
- Some typos in this manual were corrected.

Version 2.0.0 (2016-02-13):

New features:

- This manual.
- *Windows edition only:* If a previous version of the application has been installed, the installer skips the dialog box in which you can set the destination folder and the dialog box in which you can set the program group in the Windows menu. So if you wish to install to a different folder or program group, you must uninstall the previous version first.
- The new Restart operation terminates FCB1010 Manager and automatically starts a new instance of it. There is also a version of Restart that restarts FCB1010 Manager with its default setup.
- Maintenance of multiple application setups (such as window positions and the enabled/disabled states of MIDI devices) via setup (‘.stp’) files. See the Setup submenu: on OS X this is in the FCB1010 Manager pull-down menu, on Windows in the File pull-down menu of the main window. Note that opening a previously saved setup file involves restarting the application. You can also start the application with a setup file as a command line parameter: *-s setupfile* on OS X, */s setupfile* on Windows.
- Maintenance of the application’s window positions via desktop (‘.dsk’) files; see View ⇒ Desktop in the main window. Unlike opening a setup file, opening a desktop file does *not* involve restarting the application.
- You can sanitize the lists of most-recently opened/saved files in the menus via two operations: ‘Remove absent files from list’ and ‘Clear list’.
- In the main window several menu items have received images and several buttons have been added to the toolbar.
- In the ‘Window list’ dialog box you can make all windows visible in one operation.
- In the MIDI input messages window and the Presets window: the grid updates its display immediately when you mouse-drag the ‘thumb’ in its horizontal or vertical scrollbar. (Previously these grids updated only when you released the mouse button.)
- If ‘Switch 1 mode’ or ‘Switch 2 mode’ in the Global Configuration window is ‘State’, the preset table now says ‘Off’ in the SW 1 or SW 2 column for switches that are unchecked in the preset window. (Previously *nothing* was shown in this situation, which obscured the fact that selecting

- such a preset does switch SW 1 or 2 off.)
- If ‘Switch 1 mode’ or ‘Switch 2 mode’ in the Global Configuration window is ‘Impulse’, the preset table now says ‘Imp’ in the SW 1 or SW 2 column for switches that are checked in the preset window. (Previously it said ‘On>Off’, which didn’t quite fit in the grid cells unless you manually widened the column.)
- New: the ‘MIDI controllers’ window, in which you can receive and send Control Change messages.
- New: the ‘MIDI program changer’ window, from which you can send Program Change messages, optionally prefixed by Bank Select MSB/LSB messages.

Bug fixes (OS X and Windows editions):

- In the preset window: if the FCB1010 firmware version is 2.5 and Direct Select is off, NOTE’s background color updates immediately when CNT 1 gets enabled or disabled.
- In the preset window: if CNT 1 has been enabled, its Controller and Value fields no longer get hidden when NOTE gets enabled or disabled.
- The Soft FCB1010 window no longer outputs a spurious Note On message with velocity 0 when the first digit of a new preset is being selected if Direct Select is on and NOTE is enabled in the old preset.

Bug fixes (OS X edition only):

- Incoming MIDI System Exclusive messages are now processed correctly. (Previously the terminating F7 could get cut off, leading to error message ‘Not enough MIDI SysEx input buffers’.)
- Non-SysEx messages coming in in quick succession are now processed correctly. (Previously such messages would get skipped and lead to error message ‘Non-SysEx MIDI input message too long’.)
- MIDI input messages window: the actual maximum number of messages in the grid now keeps matching the value stipulated in the Options dialog box.
- The menu bar now consistently shows the active window’s pull-down menus. So if the active window doesn’t *have* any pull-down menus, nothing is shown; previously in this situation, the pull-down menus of the previously active window would still be shown.
- Selecting a bank in the Banks window or a preset in the Soft FCB1010 window no longer leads to a wrongly highlighted cell in a previously selected row in the Presets window.

Version 1.2.2 (2015-04-30):

- First version also available for OS X.
- The menu of the Presets window has been simplified: the Disable, Enable, Fix and Increment submenus (which used to be part of the Edit pull-down menu) are now pull-down menus in their own right.
- Many internal improvements to MIDI input and output communication. In particular, the input system for SysEx messages has been redesigned completely, one consequence being that the maximum SysEx message length is now fixed at 65536 bytes, hence it is no longer editable via the Input tab of the MIDI devices dialog box.
- The MIDI input messages window can display message times in various formats: (((days:))hours:)min:)sec.)ms.
- FCB1010 Manager no longer refuses to start if system.ini doesn’t exist in the Windows system folder.
- FCB1010 Manager uses the new Mountain Utilities web site at mountainutilities.eu in links and its update mechanism.
- The Donate item in the main window’s Help pull-down menu no longer opens a dialog box, but makes your web browser open the Donate page at the Mountain Utilities web site.

Version 1.2.1 (2013-01-18):

- You can keep FCB1010 Manager's main window on top of FCB1010 Manager's other windows and other applications.
- Windows that were open when the previous session of FCB1010 Manager terminated reappear exactly where they were. (In previous versions of FCB1010 Manager, even a window on a secondary monitor always reappeared on the primary display, even when the secondary monitor was still available.)

This new behavior has one potentially problematic consequence: when you remove a monitor or reduce the screen resolution, windows may become invisible upon a restart of FCB1010 Manager. To remedy this, you can use the new 'Make fully visible' operation in the 'Window list' dialog box (accessible via the main window's View pull-down menu or the Alt+0 key combination).

- LoopMIDI has been added to the list of known virtual MIDI ports ('pipes').
As in previous versions, when you start FCB1010 Manager for the first time you must choose to disable or enable all MIDI pipes in FCB1010 Manager, and this now includes all loopMIDI ports, or rather those ports that have 'loopMIDI' in their names — as suggested by loopMIDI. However, loopMIDI allows you to change these names completely — if you do so, FCB1010 Manager won't recognize them as MIDI pipes, so then you can only enable them manually. i.e. via the MIDI devices dialog box.

Version 1.2.0 (2011-01-15):

- Unicode support: in most text edit boxes in the program (e.g. for file names) you can now enter any 'international' characters. (Technical note: FCB1010 Manager now saves its ini file in UTF-8 format, starting with the three-byte UTF-8 BOM (byte order mark); however, it can still read the plain ASCII/ANSI ini files of previous versions of FCB1010 Manager.)
- As a consequence of its new Unicode support, FCB1010 Manager no longer runs under Windows 95, 98 or Me, because these operating systems do not support Unicode. (If you require a version of FCB1010 Manager that runs under these operating systems, please send a message to the contact address at the Mountain Utilities web site.)

Version 1.0.2 (2009-11-02):

All references to the Mountain Utilities web site have been updated from 'home.hetnet.nl' to 'home.kpn.nl'. Hence the automatic update mechanism will work again.

Version 1.0.1 (2009-08-06):

- The package uses a different installer (Inno Setup instead of InstallShield). Consequently, the zip file's size has gone down from 4 MiB to 570 KiB! You may also note a few minor differences in the installation procedure.
- Facilities for automatic and manual checking whether an update of FCB1010 Manager is available from the Mountain Utilities web site.

Version 1.0.0 (2009-07-08):

First published version.

3. Computer requirements

To run FCB1010 Manager, your computer must comply with the following requirements:

- Processor: Any Intel 80486- or Pentium-compatible CPU. Processor speed is relatively unimportant.
- Operating system:
 - macOS: any version for the Intel x86-architecture.
 - Windows: as of version 1.2.0, FCB1010 Manager only runs under Windows operating systems that support Unicode, such as Windows 2000, XP, Vista, 7, 8 and 10.
- An SVGA-compatible graphical card and monitor:
 - The screen size should be at least 800×600 pixels.
 - For best results, the color depth should be at least 16 bits. (At a depth of only 256 colors, some colors aren't rendered as intended.)
- A mouse.
- Free hard disk space: about 26 MiB on macOS and 8 MiB on Windows.
- RAM: when running, FCB1010 Manager normally occupies roughly 18 MiB on macOS and 5 MiB on Windows.

4. Installation of FCB1010 Manager

To install FCB1010 Manager on your computer, proceed as follows:

macOS:

1. Download FCBMan*a.b.c*.dmg (where *a.b.c* stands for the actual version number) to your computer from the FCB1010 page at the Mountain Utilities web site (<https://mountainutilities.eu/fcb1010>).
2. Open the dmg file in Finder, and drag-and-drop the FCB1010 Manager icon on the Applications icon. If a previously installed version of FCB1010 Manager exists in the Applications folder, you are asked what you want to do: it's best to select Replace.
3. Right-click the dmg file's 'disc' icon on the right side of the desktop and run Eject from the local menu.

Windows:

On Windows, three editions are available: an installer, a 32-bit portable edition and a 64-bit portable edition:

Installer:

1. Download fcyman-*a.b.c*-install.exe (where *a.b.c* stands for the actual version number) to your computer from the FCB1010 page at the Mountain Utilities web site (<https://mountainutilities.eu/fcb1010>).
2. Run fcyman-*a.b.c*-install.exe and follow its instructions. The installer automatically installs the edition of the actual application (FCBMan.exe) that matches the operating system: the 32-bit edition on a 32-bit Windows system, the 64-bit edition on a 64-bit Windows system.

Note: the installation includes an uninstaller. It can be run from the Windows Start Menu via Programs → Mountain Utilities → FCB1010 Manager, or via Settings → Control Panel → 'Add or Remove Programs' (Windows XP) or 'Programs and Features' (Windows Vista and later).

Note that when you install a new version of FCB1010 Manager, you do *not* have to uninstall any previously installed version first: the old version will be replaced with the new version automatically.

Portable (32- or 64-bit):

1. Download fcyman-*a.b.c*-*xnn*-portable.zip (where *a.b.c* stands for the actual version number and *xnn* is 'x64' or 'x86') to your computer from the FCB1010 page at the Mountain Utilities web site (<https://mountainutilities.eu/fcb1010>). The 'x64' (64-bit) edition only runs on 64-bit Windows, the 'x86' (32-bit) edition on 32- and 64-bit Windows.
2. Unzip fcyman-*a.b.c*-*xnn*-portable.zip completely (maintaining the zip file's tree structure) to any folder to which FCB1010 Manager itself (FCBMan.exe) will have write-access (which is necessary for its configuration files). Crucially, in Windows Vista and later you should *not* unzip to a subfolder of C:\Program Files or C:\Program Files (x86), since these folders are subject to Windows' UAC (User Access Control), which means that FCB1010 Manager would *not* have write-access.

Running FCB1010 Manager itself

After installation, you can start FCB1010 Manager itself: FCB1010 Manager.app on macOS (from the Applications folder), FCBMan.exe on Windows (e.g. via the Windows start menu).

If you have never run FCB1010 Manager (in any version) from the installed operating system before, the program notifies you that it can't find your configuration. This is normal: the program

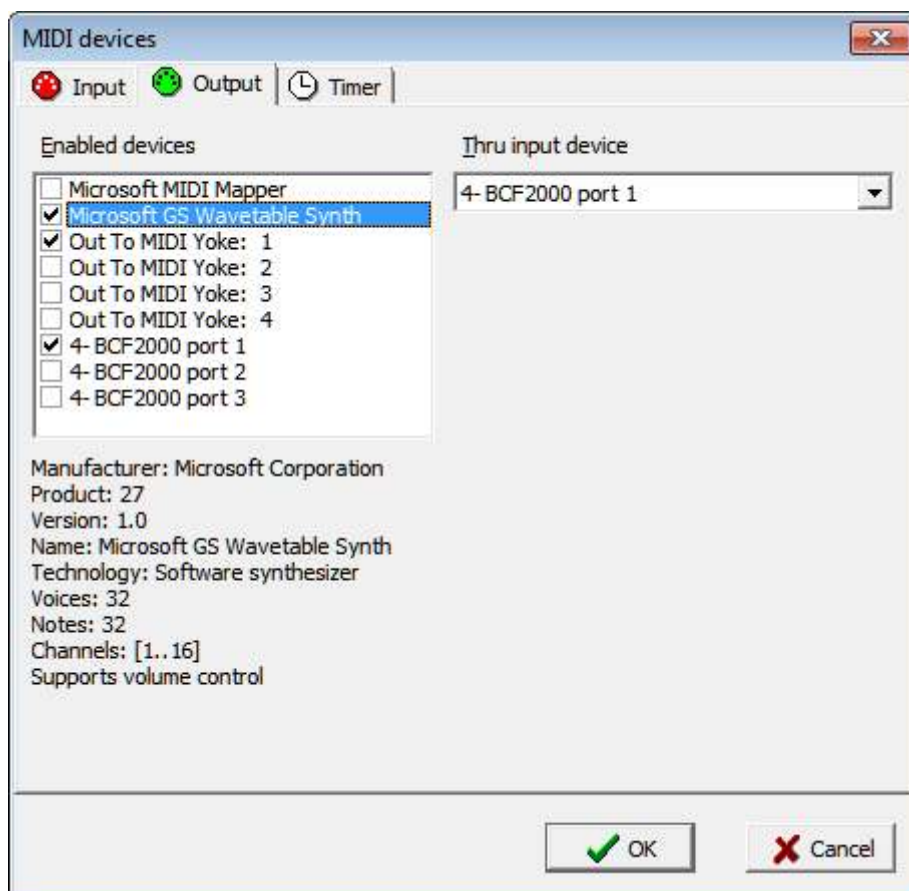
saves its configuration file (fcbman.ini in macOS, FCBMan.ini in Windows) whenever the program *terminates*, so obviously this configuration file doesn't exist yet when you start the program for the very first time. The program also notifies you if the configuration file does exist but belongs to a previous version; all existing settings are retained.

5. MIDI setup

FCB1010 Manager can only communicate with MIDI input and output devices that are *enabled*.

When FCB1010 Manager is started for the first time, it enables all MIDI input and output devices, except that the Windows edition skips the MIDI Mapper and any software synthesizers. However, you are advised to fine-tune this default setup to suit your particular needs. For instance, if you want to run other programs using MIDI devices simultaneously, it may be a good idea to keep as many MIDI devices disabled as you can in FCB1010 Manager, in order to avoid MIDI device access conflicts.

From the main window's Options pull-down menu, 'MIDI devices' opens a dialog box in which you can select the MIDI devices to which FCB1010 Manager connects:



Tip (Windows only): In this dialog box you can also set up a 'soft MIDI thru' link, by connecting an enabled MIDI input to an enabled MIDI output device. Any MIDI message received by the MIDI input device is then immediately passed on to the MIDI output device. In fact, you can set up as many links as there are devices; the only restriction is that each device can have only one connection.

MIDI pipes

Of particular interest is the enabling/disabling of 'MIDI pipes'. 'MIDI pipe' is my term for what is commonly known as a 'virtual MIDI device'. This is software that manifests itself as a virtual MIDI

output port plus a virtual MIDI input port: the driver passes any MIDI data sent to the output port to the input port. Hence, when a computer program sends data to the virtual MIDI output port, this data can be picked up at the corresponding input port by any other computer program. Thus, a MIDI pipe allows inter-program MIDI communication. If a MIDI pipe driver is ‘multi-client’, we can connect more than one program (up to a particular maximum) to the same virtual output or input port.

To my knowledge, the following free MIDI pipe drivers are available (please contact me if you know any others):

- **Hubi’s Loopback Device:** 4 multi-client pipes, but for Windows 9x only.
- **Sony/Sonic Foundry Virtual MIDI Router:** 4 single-client pipes. Windows 32-bit only.
- **Hurchalla Maple:** 12 single-client pipes. Windows 32-bit only.
- **LoopBe1:** only 1 multi-client pipe, so not very useful. (No, you can’t install more than one copy!) Windows 32-bit only.
- **LoopBe30:** 30 multi-client pipes, but the trial version only works for a brief period, and the full version is not free. Windows 32-bit only.
- **MIDI Yoke** (<http://www.midiox.com/>): the NT (/2000/XP/Vista/7/8(?)) version allows up to 16 multi-client pipes, so understandably this has been the most popular MIDI pipe driver for 32-bit Windows versions.
 - Problems:
 1. MIDI Yoke’s NT version, even though it is 32-bit, *can* be installed under 64-bit operating systems, but (reportedly) the pipes are only accessible to 32-bit DAWs, not to 64-bit DAWs.
 2. To work with MIDI Yoke correctly under Windows versions with UAC (User Access Control) you must apply a manual tweak:

The MIDI Yoke installer tries to create MIDI Yoke’s configuration file (MYOKENT.INI) in C:\Windows, but the operating system doesn’t allow this and actually creates it in C:\Users\Username\AppData\Local\VirtualStore\Windows. On the other hand, the MIDI Yoke configuration applet under Control Panel *does* have write access to C:\Windows and will create a second copy of MYOKENT.INI there when you change the settings. However, the latter file will never be seen by the *driver* (because the operating system keeps redirecting it to the copy in ...\VirtualStore\Windows); in other words, the driver ‘won’t listen to you.’

To fix this, you must manually remove MYOKENT.INI from C:\Users\Username\AppData\Local\VirtualStore\Windows or move it to C:\Windows, using administrator rights.
 3. As discussed below, the NT version of MIDI Yoke may slow down the termination of MIDI Tools.
- **CopperLan** (<http://www.copperlan.org/>): Available for macOS and Windows (32- and 64-bit). This is primarily a MIDI-over-Ethernet system, so it’s a bit of overkill if you only need local MIDI pipes. (CopperLan 1.4 for Windows installs *three* drivers, if I remember correctly!) It offers up to 32 virtual MIDI input ports and 32 virtual MIDI output ports; by default none of these are set up as ‘pipes’, but you can manually connect any output to any input (although the idiosyncratic user interface makes this much more difficult than it should be).

- **loopMIDI** (<http://www.tobias-erichsen.de/>):

This allows you to create and destroy any number of MIDI pipes on the fly.

Simple and effective, so probably the best choice on modern Windows versions (particularly 64-bit versions, given MIDI Yoke's problems on those).

If configured improperly, MIDI pipes can easily cause problematic MIDI signal paths. There are several dangers:

Duplication:

If there is first a *split* in the signal path, and then a *merge*, two or more copies of the same MIDI message arrive at the end of the signal path (i.e. the target MIDI device). This is time-consuming in all cases, but — even worse — it can mess up communication with certain MIDI devices.

Feedback:

In general, feedback involves the return of a sent MIDI message to the same MIDI hardware device or computer program that sent the message. Obviously this needlessly slows down operation, although it isn't necessarily disastrous. However, there may also be more sinister effects. For instance, feedback may interfere with FCB1010 Manager's communication with the device.

At the very least you should normally avoid enabling both the output port and the input port of the same MIDI pipe in the same program (e.g. FCB1010 Manager), because by definition anything you send to a MIDI pipe's *output* port (the pipe's starting point) is returned at the corresponding MIDI pipe's *input* port (the pipe's end point). So for instance, if you enable both 'Out To MIDI Yoke: 1' and 'In From MIDI Yoke: 1' in a program, then any MIDI data the program sends to 'Out To MIDI Yoke: 1' is returned to the program at 'In From MIDI Yoke: 1'. This type of feedback is usually undesired, except perhaps for monitoring purposes.

The most severe type of feedback occurs when the sender/recipient actually *re-sends* the returned MIDI message: this leads to an infinite loop, which may well grind the sender/recipient (and indeed the whole computer) to a virtual standstill.

In FCB1010 Manager this horror can happen if you enable the MIDI Thru feature in the MIDI devices dialog box for an input-output pair already exhibiting feedback. For instance, if you activate MIDI Thru from 'In From MIDI Yoke: 1' to 'Out To MIDI Yoke: 1' in FCB1010 Manager's MIDI devices dialog box, then any MIDI data sent to 'Out To MIDI Yoke: 1' not only comes back to FCB1010 Manager at 'In From MIDI Yoke: 1' (via MIDI Yoke's pipe 1), but is then automatically *re-sent* from 'In From MIDI Yoke: 1' to 'Out To MIDI Yoke: 1' via the MIDI Thru feature, in principle ad infinitum, although MIDI Yoke does perform some checks that spot and kill the infinite loop — but still...

Close delay:

This problem only occurs with MIDI Yoke NT (but *not* with MIDI Yoke for Windows 95/98/Me): closing any MIDI Yoke NT 1.75 *input* port causes a delay of 1 second. (Certain earlier versions even 3 seconds.)

Concerning FCB1010 Manager this is mainly relevant during program exit. In principle FCB1010 Manager terminates almost instantly upon exit, but when all the input ports of MIDI Yoke NT 1.75 are enabled, termination of FCB1010 Manager takes some 16 seconds longer than normal! Therefore you should disable as many MIDI Yoke NT input ports as possible in FCB1010 Manager's MIDI devices dialog box, i.e. any MIDI Yoke NT input ports that FCB1010 Manager itself doesn't use. (Note that you can still use any MIDI ports disabled in FCB1010 Manager in *other* programs!)

To help you avoid some of the serious problems discussed above, FCB1010 Manager takes the following steps:

- On *first* startup, if FCB1010 Manager detects any of the MIDI pipes listed above [excluding CopperLan, since by default its virtual ports aren't interconnected], it asks you if you want to enable the I/O devices of these pipes. It's best to answer *No* (to avoid feedback loops, and to avoid MIDI Yoke NT's close delays during FCB1010 Manager's exit procedure), unless some other program (e.g. MIDI-OX) is routing one of your MIDI devices through a MIDI pipe.
- On *every* startup, FCB1010 Manager optionally warns you if any MIDI Yoke NT input ports are enabled and thereby cause extra delays during termination of FCB1010 Manager. You can enable/disable this warning on the Input tab of the MIDI devices dialog box.

6. The main window

FCB1010 Manager's main window only consists of a pull-down menu, a toolbar and a statusbar:



The toolbar merely contains a number of buttons duplicating some of the most useful menu items.

The statusbar contains two panels:

- The panel on the left says 'Modified' if there are any unsaved edits.
- The panel on the right displays the name of the last-opened/saved MIDI SysEx file.

The menu provides the following operations:

File → FCB1010:

Opens a submenu from which you can open and save an FCB1010's setup.

Note that you only have to specify the FCB1010 SysEx ('.syx') data file: the bank and preset descriptions (which are metadata only maintained by FCB1010 Manager) are automatically opened from and saved to an accompanying '.fcb' file. **Make sure that you always copy/move the fcb file along with the syx file!**

File → Export bank/preset names to text file:

Allows you to export the bank and preset names to a text ('.txt') file. This file may be useful for further processing, such as opening by other FCB1010 editors.

FCB1010 Manager (macOS)/File (Windows) → Restart:

Terminates FCB1010 Manager and starts a new instance of it. This is particularly useful after the MIDI device configuration has changed while FCB1010 Manager has been running, either because a MIDI device (often a USB-based one) has become invalid or because a new one has become available: restarting FCB1010 Manager updates FCB1010 Manager's device list to the new configuration.

FCB1010 Manager (macOS)/File (Windows) → Restart with default setup:

Functions like Restart (see above), with two differences:

- Since this operation is somewhat 'momentous', a dialog box requires you to confirm that you indeed want to do this.
- The default setup file FCBMan.stp is deleted before the restart, so that all setup values (such as MIDI I/O device settings and window positions/sizes) are restored to their defaults. Consequently, FCB1010 Manager's new instance behaves as if you have never run the application before.

This operation may be useful when some setup problem has developed that you find yourself unable to fix quickly otherwise. However, the restoration of the default setup also has its drawbacks: for instance, you must configure all MIDI devices again.

Note: 'Restart with default setup' does not destroy the file lists in the application's menus (such as the Setup submenu — see below): these file lists are not stored in FCBMan.stp but in

FCBMan.mru. If you want to clear these file lists, simply use their ‘Clear list’ operations, or (for a total clearance) remove FCBMan.mru while the application isn’t running.

FCB1010 Manager (macOS)/File (Windows) → Setup:

A submenu from which you can open and save setup (‘.stp’) files.

A setup file includes nearly all the application’s customizable settings, including MIDI I/O device settings and window positions/sizes.

By opening a (previously saved) setup file you can quickly switch from one setup to another. However, opening a setup file involves restarting the application; thus you will lose transient data like unsaved recorded MIDI input messages. So if you only wish to change the window positions/sizes, it’s simpler to use the desktop (‘.dsk’) file facility (see View → Desktop), because this doesn’t involve restarting the application.

Tip: you can force FCB1010 Manager to use a specific setup file via the command line, as follows:

macOS: -s *setupfile*

Windows: /s *setupfile*

FCB1010 Manager → Quit (macOS)/File → Exit (Windows):

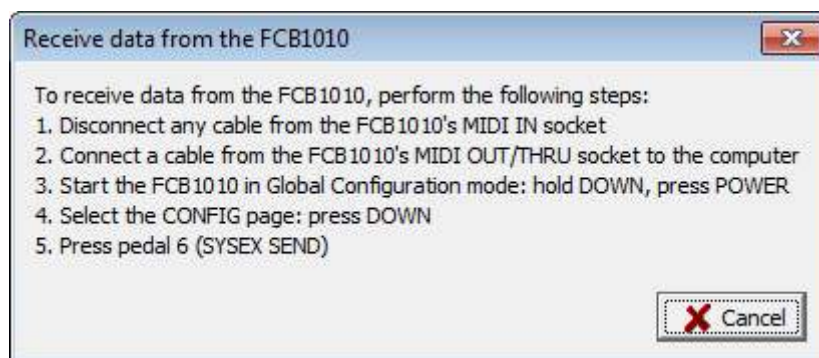
Terminates FCB1010 Manager.

Note that on Windows the associated hotkey (Alt+X) works from almost *any* location in the program, not just from the main window. Of course you can also terminate FCB1010 Manager by clicking on the X icon on the main window’s title bar: the same questions are asked. Pressing Alt+F4 also works, but (unlike Alt+X) only from the main window.

MIDI → Receive:

Allows you to send an FCB1010’s setup to FCB1010 Manager. The MIDI OUT/THRU socket of the FCB1010 must be connected to the ‘MIDI input port’ defined in the FCB1010 options dialog box. (And of course this MIDI input port must have been enabled in the MIDI devices dialog box.)

The following dialog box pops up [at least in the Windows 2.1.0 version; in the macOS 2.0.0 version the instructions are more primitive]:



Please refer to the official FCB1010 manual for further details on the SYSEX SEND procedure.

A few pitfalls:

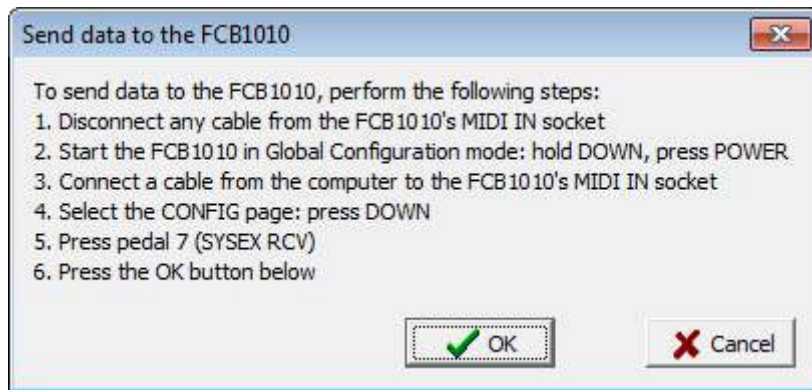
- Entering Global Configuration mode may fail if MIDI messages are being sent to the FCB1010’s MIDI IN socket at the moment you switch on the FCB1010. This even applies to the Timing Clock and Active Sensing messages that many MIDI devices (incl. interfaces) output continuously. So it’s usually best to temporarily disconnect any cable from the MIDI IN socket before entering Global Configuration mode. (And you don’t need to reconnect it

- for this procedure.)
- You cannot transmit an FCB1010 setup via the USB cable of a Behringer BCF2000 or BCR2000 MIDI Controller. This is because the FCB1010's SysEx message is 2352 bytes long and the BCF/R2000's USB MIDI facility maims any SysEx message longer than 1019 bytes.

MIDI → Send:

Allows you to send the FCB1010 setup currently defined in FCB1010 Manager to an FCB1010. The MIDI IN socket of the FCB1010 must be connected to the 'MIDI output port' defined in the FCB1010 options dialog box. (And of course this MIDI output port must have been enabled in the MIDI devices dialog box.)

The following dialog box pops up [at least in the Windows 2.1.0 version; in the macOS 2.0.0 version the instructions are more primitive]:



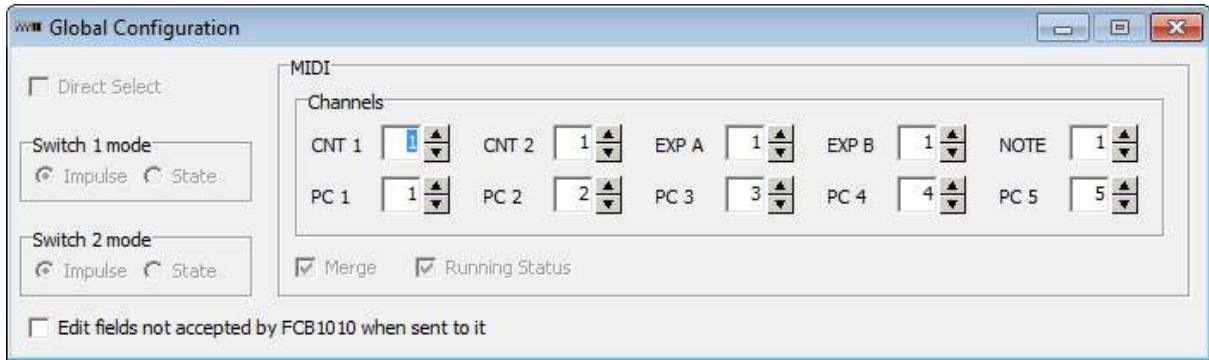
Please refer to the official FCB1010 manual for further details on the SYSEX RCV procedure.

A few pitfalls:

- Entering Global Configuration mode may fail if MIDI messages are being sent to the FCB1010's MIDI IN socket at the moment you switch on the FCB1010. This even applies to the Timing Clock and Active Sensing messages that many MIDI devices (incl. interfaces) output continuously. So it's usually best to temporarily disconnect any cable from the MIDI IN socket before entering Global Configuration mode. But of course you must reconnect the cable before the actual transfer, i.e. before pressing the OK button in the 'Send data' dialog box.
- You cannot transmit an FCB1010 setup via the USB cable of a Behringer BCF2000 or BCR2000 MIDI Controller. This is because the FCB1010's SysEx message is 2352 bytes long and the BCF/R2000's USB MIDI facility maims any SysEx message longer than 1019 bytes.

View → Global Configuration:

Opens a window in which you can set the FCB1010's global configuration:



All settings in this window are stored in the FCB1010's SysEx file (together with the 100 individual preset definitions).

However, as you can see, many of these settings are grayed-out. This is because these settings aren't actually *accepted* by a real FCB1010 when it receives them: you will have to edit these settings manually on a real FCB1010. (See the official FCB1010 manual for the proper procedures.)

Nevertheless, these settings can be un-grayed by checking 'Edit fields not accepted by FCB1010 when sent to it', and then you can edit them — at least in FCB1010 Manager! (It's good practice to uncheck this again immediately after you're done editing.) There are two reasons why this can be useful:

1. These settings affect the behavior of FCB1010 Manager's 'Soft FCB1010' window.
2. Matching the settings on your real FCB1010 may help you remember them.

View → Banks:

Opens a window in which you can edit the descriptions of the ten banks:

| Bank | Description |
|------|---------------------------------|
| 0 | GS-10: ChSel + 2 2-pedal switch |
| 1 | GS-10: 4 2-pedal switches + SW |
| 2 | GS-10: 8 1-pedal switches + SW |
| 3 | GS-10: 5 2-pedal switches |
| 4 | GS-10: 4 MS + 2 2-pedal switch |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| 9 | |

These descriptions are stored in the '.fcb' file that is saved/opened together with the main SysEx data file, and they are displayed in the Soft FCB1010 window.

View → Presets:

Opens a window containing a table of all 100 presets:

| Preset | SW 1 | SW 2 | PC 1 | PC 2 | PC 3 | PC 4 | CNT 1 | Val 1 | CNT 2 | Val 2 | PC 5 | NOTE | EXP A | Min A | Max A | EXP B | Min B | Max B | Description |
|--------|------|------|------|------|------|------|-------|-------|-------|-------|------|------|-------|-------|-------|-------|-------|-------|------------------|
| 0:1 | | | | | | | 6 | 0 | | | | | 11 | 0 | 127 | 12 | 0 | 127 | Channel Select A |
| 0:2 | | | | | | | 6 | 1 | | | | | 11 | 0 | 127 | 12 | 0 | 127 | Channel Select B |
| 0:3 | | | | | | | 3 | 0 | | | | | 11 | 0 | 127 | 12 | 0 | 127 | CC 3 Off |
| 0:4 | | | | | | | 4 | 0 | | | | | 11 | 0 | 127 | 12 | 0 | 127 | CC 4 Off |
| 0:5 | Imp | | | | | | | | | | | | 11 | 0 | 127 | 12 | 0 | 127 | Switch 1 |
| 0:6 | | | | | | | 6 | 2 | | | | | 11 | 0 | 127 | 12 | 0 | 127 | Channel Select C |
| 0:7 | | | | | | | 7 | 127 | 7 | 0 | | | 11 | 0 | 127 | 12 | 0 | 127 | CC 7 On-Off |
| 0:8 | | | | | | | 3 | 127 | | | | | 11 | 0 | 127 | 12 | 0 | 127 | CC 3 On |
| 0:9 | | | | | | | 4 | 127 | | | | | 11 | 0 | 127 | 12 | 0 | 127 | CC 4 On |
| 0:10 | Imp | | | | | | | | | | | | 11 | 0 | 127 | 12 | 0 | 127 | Switch 2 |

Note that the columns from PC 1 to NOTE are in the order in which the FCB1010 outputs a preset's MIDI messages when the corresponding pedal is pressed. It's for this reason that PC 5 is displayed to the right of CNT 2; see under View → MIDI → Program changer for further discussion.

Via the menu and the toolbar you can edit the presets in various ways. You can select any range of presets: any edit applies to all selected presets simultaneously.

Note that the this table applies the same color rules as the Preset window (see below).

View → Selected preset:

Opens the Preset window, in which you can edit the definition of the selected preset:

'Unused functions' in the Options → FCB1010 dialog box determines whether the Controller, Value and Program fields of any disabled CNT, EXP, NOTE and PC are grayed out or invisible. (Invisible is the default.)

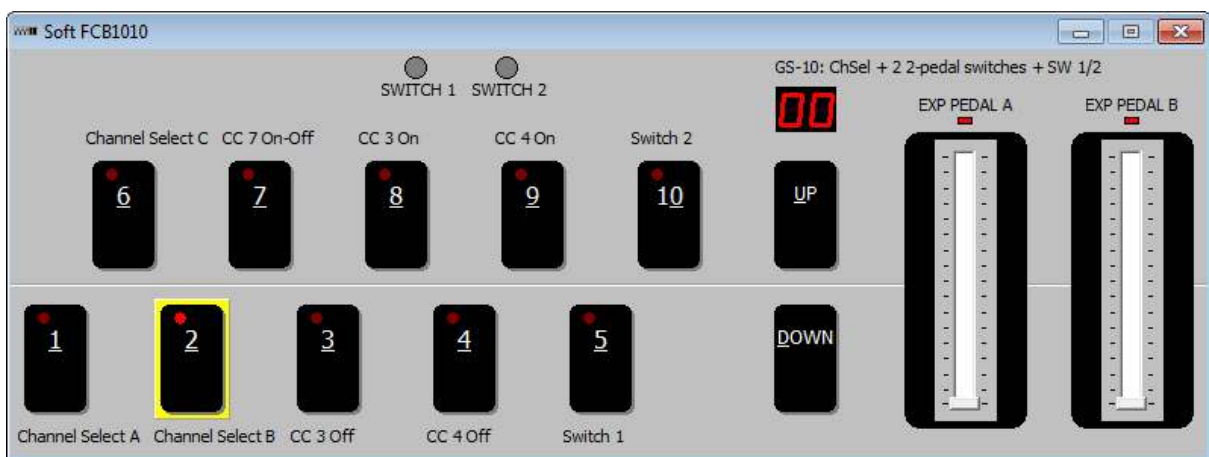
Note that the Description fields of all presets are stored in the '.fcb' file that is saved/opened together with the main SysEx data file. These descriptions are shown in the Presets window and in the Soft FCB1010 window.

The following rules govern the background colors of CNT 1, CNT 2 and NOTE:

- Normally the background of CNT 1 is light pink and that of CNT 2 medium pink. However, when CNT 1 and CNT 2 are both enabled on the same MIDI channel and have the same controller number, both backgrounds are dark pink. This indicates that the ‘controller toggle function’ is active: pressing the pedal repeatedly toggles between the values defined in CNT 1 and CNT 2. (See section 2.5 of the official FCB1010 manual.)
- If the FCB1010’s firmware (as set in the FCB1010 options dialog box) is version 2.5 and Direct Select (as set in the Global Configuration window) is disabled, the backgrounds of CNT 1 and NOTE are bright green when NOTE is *enabled* but CNT 1 is *disabled*. This indicates that the ‘tap-tempo function’ is active. (See section 2.6 of the official FCB1010 manual.)

View → Soft FCB1010:

Opens the Soft FCB1010 window:



This window offers a virtual FCB1010:

- The description of the selected bank is shown in the upper right corner of the window (above the two red digits).
- The descriptions of presets 1-10 are shown below/above the corresponding pedals.

This window behaves almost identically to a real FCB1010:

- Selecting a preset works the same way as on a real FCB1010:
 - If Direct Select (in the Global Configuration window) is off, you switch between banks (0-9) by clicking the UP and DOWN pedals, and you select a preset (1-10) by clicking the corresponding pedal.
 - If Direct Select is on, you select a preset by clicking its two corresponding pedals, and clicking the UP and DOWN pedals toggle SWITCH 1 and 2 respectively.

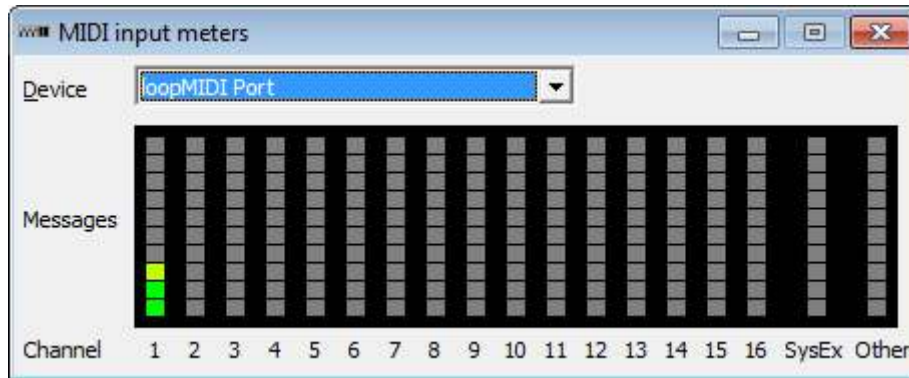
Tips:

- You can also press a pedal by pressing the underlined letter/number on your computer keyboard.
- You can (re-)press the selected pedal (the one with a yellow border) by pressing the spacebar on your computer keyboard.
- You can move from one pedal to the next/previous via Tab/Shift+Tab.
- If you select a preset or move one of the EXP PEDAL sliders, FCB1010 Manager outputs MIDI data to the ‘MIDI output device for Soft FCB1010’ selected in the FCB1010 options window (accessed via Options → FCB1010), exactly as a real FCB1010 would. However, SWITCH 1 and 2 on a real FCB1010 don’t produce MIDI output, so the Soft

FCB1010 window only toggles their LEDs in the window.

View → MIDI → Input meters:

Opens a window showing the messages received recently from the MIDI input devices, via (logarithmical) LEDs per MIDI channel:



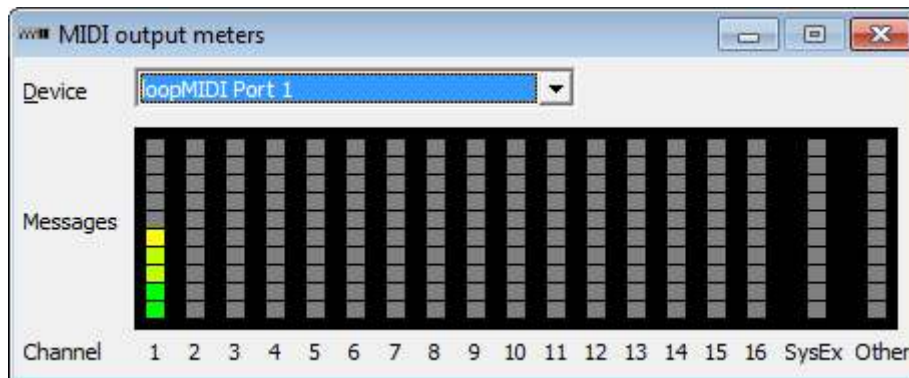
This window can be useful for troubleshooting your MIDI connections. 'DISABLED' in front of the selected MIDI input device indicates that the device is disabled, so no MIDI messages can currently be received from that device. (You can enable devices in the MIDI devices dialog box.)

View → MIDI → Input messages:

Opens the MIDI input messages window. Here you can record and view messages from the MIDI input devices. See §7 for more information.

View → MIDI → Output meters:

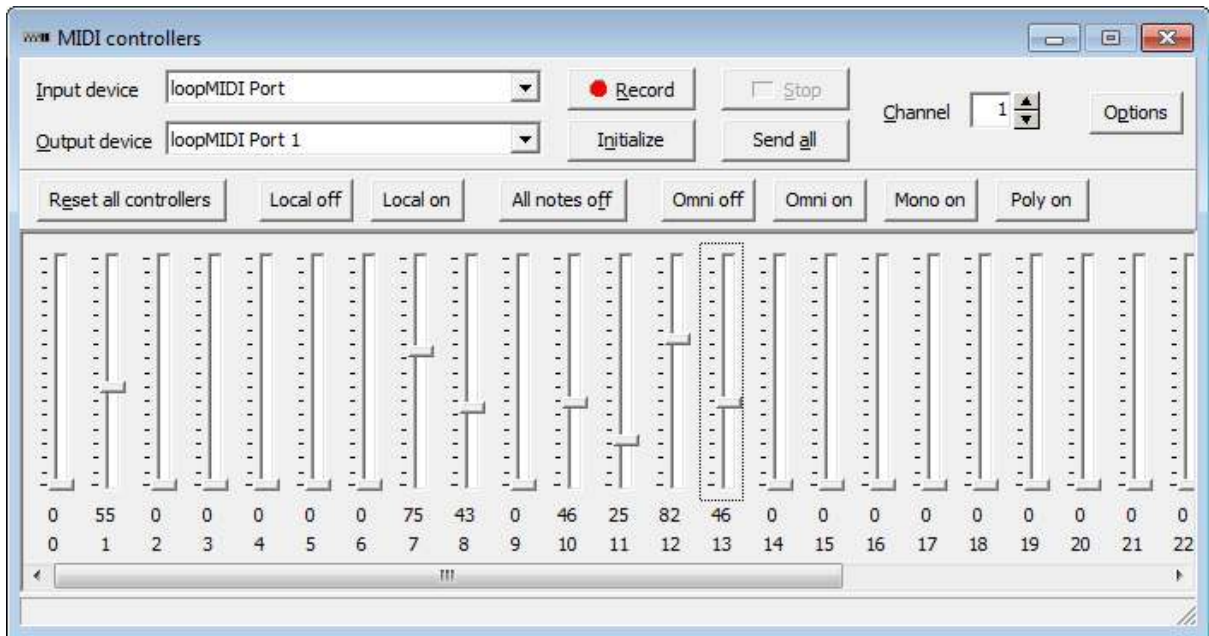
Opens a window showing the messages sent recently to the MIDI output devices, via (logarithmical) LEDs per MIDI channel:



This window can be useful for troubleshooting your MIDI connections. 'DISABLED' in front of the selected MIDI output device indicates that the device is disabled, so no MIDI messages can currently be sent to that device. (You can enable devices in the MIDI devices dialog box.)

View → MIDI → Controllers:

Opens a window in which you can receive and send MIDI Control Change messages:

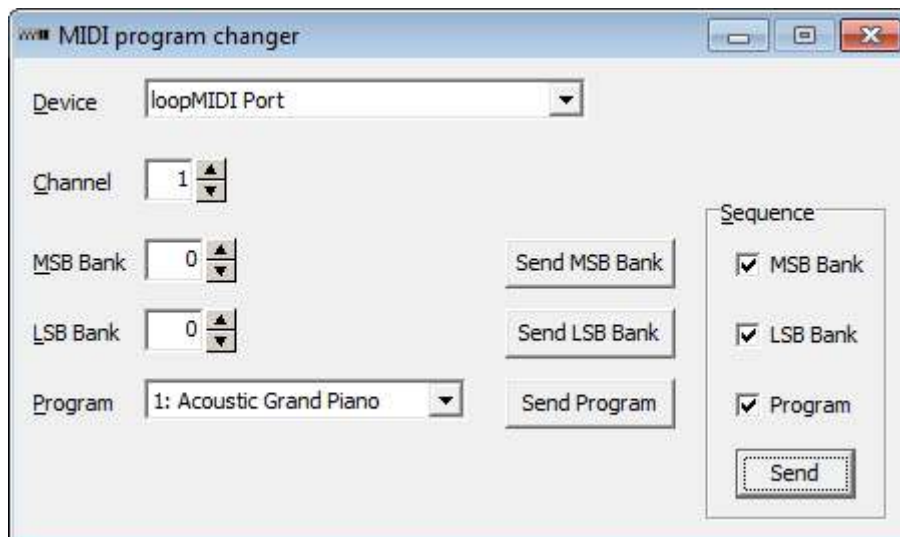


Possible uses of this window:

- From this window you can send test messages to the MIDI device that you intend to control via your FCB1010. (Note: you can also send Control Change messages from the Soft FCB1010 window, but that's not as convenient for this purpose.)
- After you have set up your FCB1010, you can test whether it actually outputs the intended Control Change messages.

View → MIDI → Program changer:

Opens a window from which you can send MIDI Program Change messages prefixed by Bank Select MSB/LSB messages, in any combination:



This window is very useful for testing devices that seem unwilling to change programs.

Tip: if you want to put a Program Change message prefixed by an MSB and/or LSB Bank message in an FCB1010 preset definition, you must put the Bank message(s) in CNT 1 and/or 2 and the Program Change message in **PC 5**, because the Program Change message must be sent *after* the Bank message(s). PC 1-4 are unsuitable in this case, since the Program Change message

would be sent *before* the Bank message(s).

View → Desktop:

A submenu from which you can open and save desktop (‘.dsk’) files.

A desktop file contains the positions, sizes and states (hidden/visible/minimized) of all the windows in the application. Thus, a desktop file contains a subset of the data in a setup file: see FCB1010 Manager (macOS)/File (Windows) → Setup.

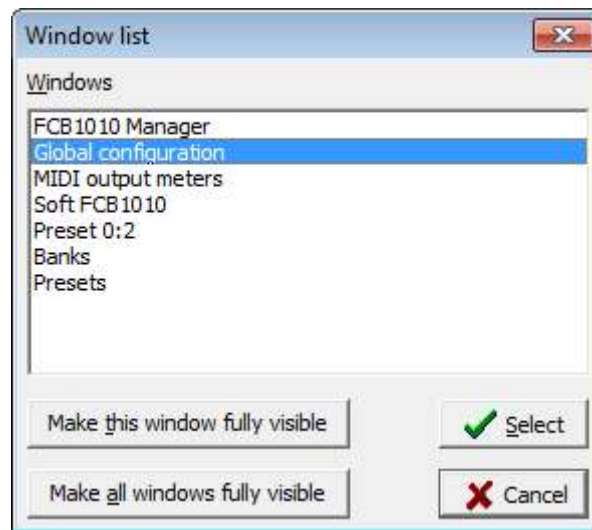
By opening a (previously saved) desktop file you can quickly switch from one desktop (i.e. layout) to another. Unlike opening a setup file, this does not involve restarting the application. By default the ‘Keep numbers’ option is off, so that opening or saving a file promotes the file name to position 1 in the list; when ‘Keep numbers’ is on, the list stays as it is, which can be useful when you’re continually switching between particular desktops.

View → Stay on top:

When this menu item is checked, FCB1010 Manager’s main window stays on top of any other windows belonging to FCB1010 Manager. On Windows, the main window also stays on top of other applications (except of course those that have the stay-on-top property too).

View → Window list:

Opens a dialog box that allows you to quickly navigate to any open window:



Note that the hotkey (Alt+0) for opening this dialog box works from almost *any* location in the program, not just the main window.

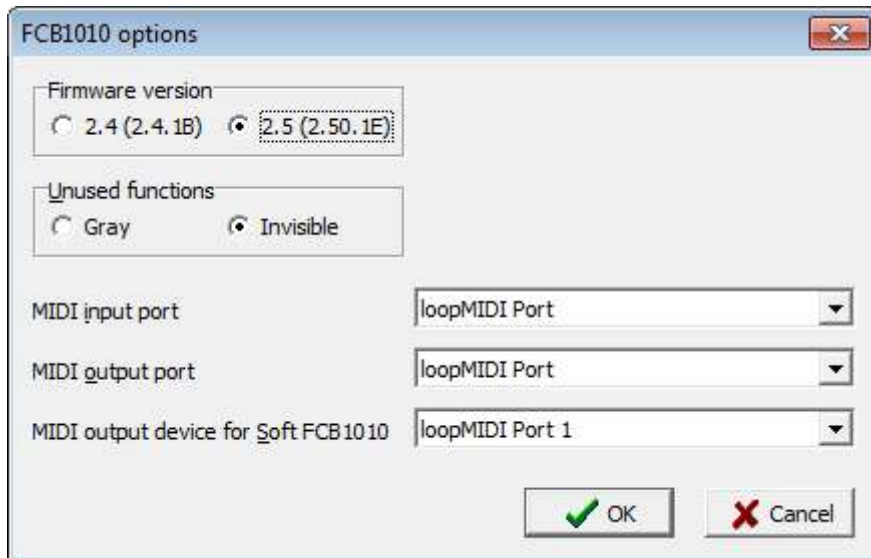
If the highlighted window is partially or completely outside the current monitor(s), you can move it into full view by pressing ‘Make this window fully visible’. ‘Make all windows fully visible’ performs this operation on *all* windows in the list.

Options → MIDI devices:

Opens a dialog box in which you can configure the MIDI devices that FCB1010 Manager monitors. See §5 for more information.

Options → FCB1010:

Opens a dialog box in which you can set several FCB1010-related options:



- **Firmware version:**
If an FCB1010's ROM chip contains firmware version 2.5, the FCB1010 is capable of the 'tap-tempo function'. (See section 2.6 of the official FCB1010 manual.)
If you select '2.5' in this dialog box, FCB1010 Manager assumes the tap-tempo function is indeed possible, and behaves accordingly: the Presets window and the Preset window apply different color rules concerning CNT 1 and NOTE, and the Soft FCB1010 window's MIDI output is different.
- **Unused functions:**
Determines whether the Controller, Value and Program fields of disabled CNT, EXP, NOTE and PC are grayed out or invisible in the Preset window.
- **MIDI input/output port:**
These ports are used for receiving and sending setup data to a connected FCB1010.
- **MIDI output device for Soft FCB1010:**
When you press/move the pedals in the Soft FCB1010 window, any generated MIDI data are sent to the device selected here.
Typically this should be the MIDI device (sound processor etc.) that you will later control by your actual FCB1010. It should *not* be the MIDI port to which your actual FCB1010 is connected!

Options → Hints:

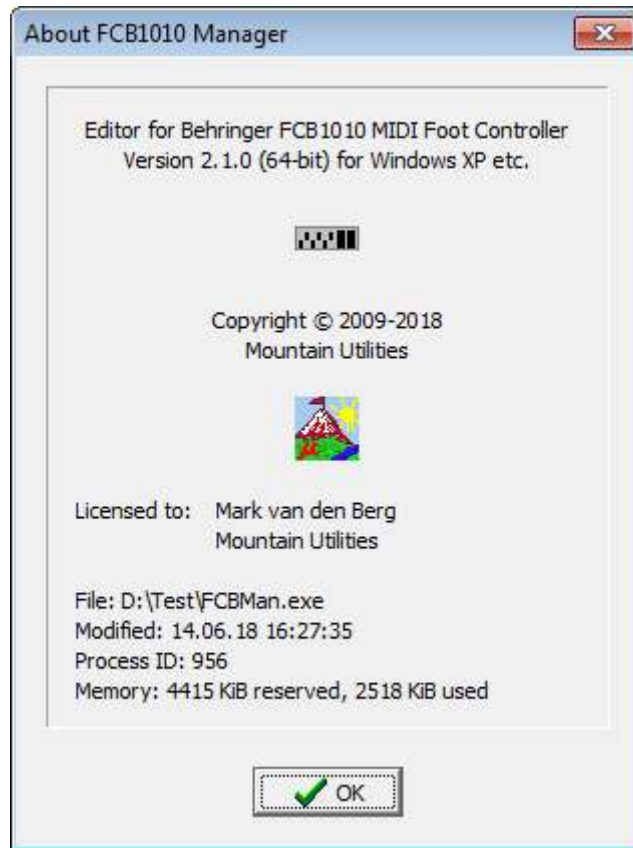
Opens a dialog box in which you can set options affecting the hints that are displayed when you move the mouse cursor over buttons etc.:

Help → Manual:

Opens this manual in the external application associated with the file extension 'pdf'.

FCB1010 Manager (macOS)/Help (Windows) → About FCB1010 Manager:

Opens a dialog box containing information on FCB1010 Manager, such as its version number and memory usage:

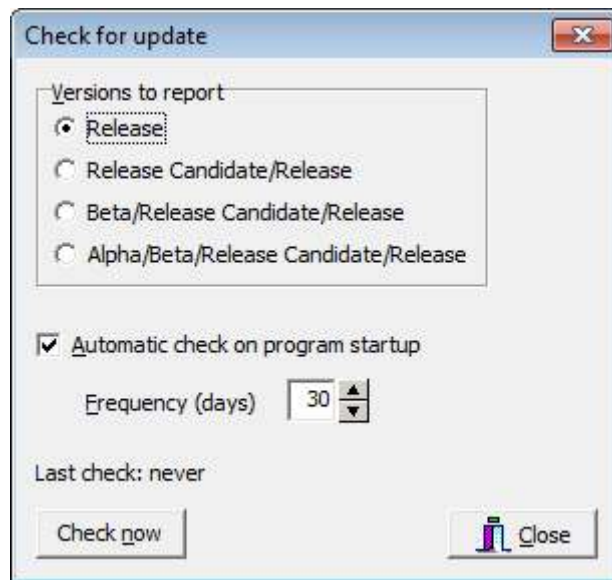


Help → Mountain Utilities web site:

Provided you're connected to the internet, your web browser opens the Mountain Utilities web site, where you can find up-to-date information about FCB1010 Manager and other Mountain Utilities applications and documents.

Help → Check for update:

Opens a dialog box in which you can set the frequency at which FCB1010 Manager automatically searches for updates at the Mountain Utilities web site, and [as of version 2.1.0, so this is not available in the macOS 2.0.0 edition] which types of updates you are interested in:



If you set 'Versions to report' to 'Release', you will only be notified about Release versions. 'Release Candidate/Release' will also notify you about Release Candidate versions, etcetera.

When an update is available, the program asks you whether you wish to open the program's web page at the Mountain Utilities site. You can also check for updates manually, by pressing the 'Check now' button.

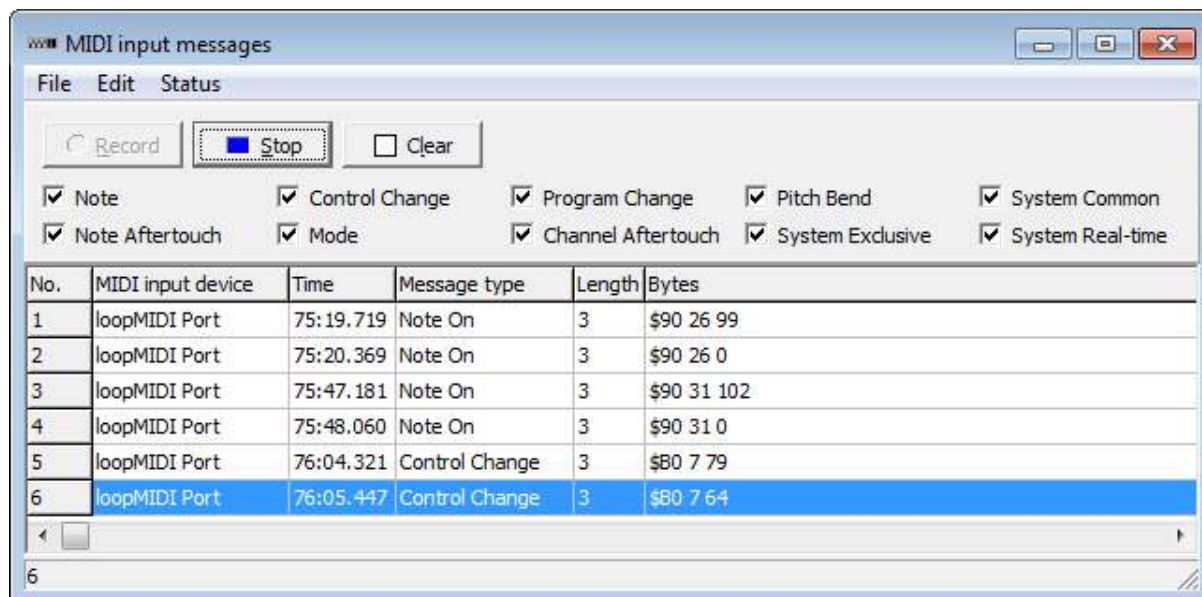
Note: If your firewall catches the program's connection attempt and asks you whether you want to allow this, you can safely say yes: no information identifying you or your computer will be sent to the Mountain Utilities web site.

Help → Donate:

Makes your web browser open the Donate page at the Mountain Utilities web site (<https://mountainutilities.eu/donate>), at which you can express your appreciation of FCB1010 Manager and support its further development by making a donation.

7. The MIDI input messages window

The MIDI input messages window is accessible from the main window's View pull-down menu (→ MIDI → Input messages).



The MIDI input messages window allows you to record and view MIDI messages sent to FCB1010 Manager from any of the currently *enabled* MIDI input devices, as defined in the MIDI devices dialog box (opened via the main window's Options pull-down menu). Thus, this window is very useful for advanced troubleshooting. You can also save recorded messages to files (in various formats).

All recorded MIDI messages are displayed in the table at the bottom of the window, one per row. The following columns exist:

- No.:
The sequential number (index) of the message in the table.
This number is for reference only. It has no further meaning: when you remove a message, the numbers of all subsequent messages simply decrease by one.
- MIDI input device:
The MIDI input device from which the message was received.
- Time:
The time at which the message was received, counted from the moment FCB1010 Manager was started. You can set this column's format in the Edit → Options dialog box (see below).
- Message type:
The type of the message: Control Change, System Exclusive, etc.
- Length:
The number of bytes in the message.

- Bytes:
The bytes of the message. The formatting (hexadecimal, decimal etc.) can be set via the options dialog box, accessed via the Edit pull-down menu.

The menu provides the following operations:

File → Save MIDI file:

Saves the selected (highlighted) MIDI messages to a standard MIDI file ('SMF'). You can load this file in a sequencer program etc.

Two versions of this operation are available via a submenu:

1. 'Times relative to first-saved message':
The original recording times are maintained, but for convenience a displacement is applied: all messages are saved with their times 'normalized' to the *first* message saved; so the first message saved itself always appears at time 0.
2. 'All times zero':
All messages are saved with their times set to zero.

Technical notes:

- The MIDI file is in 'format 0', i.e. a single track.
- For convenience, the name of the program ('FCB1010 Manager') plus its version number is included as the track name. (It completely depends on the receiving program whether you can see this in any way.)
- The file includes a tempo specification of 120 BPM.
Beware: It seems that when you import a MIDI file into an *existing* Sonar 7 project, Sonar ignores this file tempo of 120 BPM and wrongly interprets the message times according to the existing project's tempo. In the case of a file saved via 'Times relative to first-saved message', this may lead to unwanted stretching, so it's best to only import such a file into a Sonar project having a tempo of 120 BPM. (I haven't tested later Sonar versions yet.)
- MIDI 'running status' is automatically applied, i.e. where possible the status bytes of channel messages are removed.

File → Save binary file:

Saves the bytes of the selected (highlighted) MIDI messages to a binary file. Note that the recording times are *not* saved: you should save to a standard MIDI file for that (see 'Save MIDI file' above).

You can select 'bin', 'syx' or any other extension for the output file, but your choice does not affect the *content* of the output file in any way.

Beware: a *syx* output file is only valid (i.e. usable in a standard way by other programs) if it *only* contains *System Exclusive* MIDI messages. And since FCB1010 Manager specifically allows you to create a *syx* file containing only the recorded System Exclusive messages (see 'Save System Exclusive messages' below), the 'save binary file' operation is primarily intended to facilitate further processing by some specialistic computer program expecting a 'flat' sequence of MIDI messages. (Typically this is a program you write yourself!) Note that you can

also save MIDI message bytes to a *text* file (see below), which may or may not be easier for further processing.

File → Save text:

Saves the selected (highlighted) MIDI messages to a text file: the bytes of each message are output on a separate line. The bytes are written in the formats defined in the options dialog box (cf. Edit → Options), so exactly as they are currently being displayed in the Bytes column of the window's table.

You could process the output file in an external text editor, then convert them to a binary file: see 'Convert text file(s) to binary file(s)' below.

File → Save System Exclusive messages:

Saves any selected (highlighted) MIDI *System Exclusive* messages to a syx file.

File → Convert text file(s) to binary file(s):

Converts a text file containing lines of hexadecimal bytes (*without* '\$' prefixes) to a binary file. You can select 'bin', 'syx' or any other extension for the output file, but your choice does not affect the *content* of the output file in any way.

This is a somewhat obscure utility that could be applied to a text file created by a 'Save text' operation (see above), possibly edited afterwards via a normal text editor (such as Notepad). Specifically, you can thus convert a text file containing only MIDI System Exclusive messages to a legal syx file.

Edit → Copy to MIDI message clipboard:

Copies the selected (highlighted) MIDI messages to the MIDI messages clipboard. Note that MIDI 'running status' is automatically applied, i.e. where possible the status bytes of channel messages are removed.

Edit → Delete:

Removes the selected (highlighted) recorded MIDI messages.

Edit → Clear:

Removes all recorded MIDI messages.

Edit → Select all:

Selects all recorded MIDI messages.

Edit → Options:

Opens a dialog box in which you can set various options related to the MIDI input messages window:

- **Buffer size:**
Sets the number of MIDI messages that can be recorded. The default is 65536; this is also the maximum. Note that lowering this setting removes any existing recorded messages beyond the new buffer size.
- **Buffer overflow protocol:**
Determines what happens if the buffer is full (as determined by the 'buffer size' setting) when a MIDI message comes in:
 - **Clear:**

The whole table is cleared, and the incoming message is entered at number 1. This is the default setting.

- **Shift:**
The existing message at number 1 is removed from the table, all other messages shift back one position, and the incoming message is added at the bottom.
Beware: this setting can be very time-consuming.
- **Freeze:**
The incoming message is ignored. However, the recording process itself isn't stopped automatically, so when you manually remove one or more recorded messages (e.g. via the Clear button), new messages will be recorded again.
- **Stop:**
Recording stops automatically.
- **Scroll to new message:**
Determines whether the message table automatically scrolls to any incoming MIDI message.
'On' is the default, but may result in 'frantic' scrolling when MIDI input is heavy, which may also starve other parts of the program. For instance, the MIDI input and output meter windows may become unable to update their gauges at the required frequency, so that not all incoming messages are displayed. So if you want a quieter display, switch scrolling off.
Note that the number of recorded MIDI messages is always shown on the status bar at the bottom of the window: this allows you to establish that messages are being recorded even when you have disabled scrolling.
- **Time format:**
Determines the time format used in the Time column. Five formats are available: ms, sec.ms, min:sec.ms, hrs:min:sec.ms and days:hrs:min:sec.ms. So e.g. in the sec.msec format you could get '61.000', which would be '1:01.000' in the min:sec.msec format.
- **Byte formats:**
Determines the ways in which MIDI message bytes are formatted: this affects both the window's Bytes column and the 'Save text' operation.
Separate settings are available for 'status' and 'data' bytes in both System Exclusive and non-System Exclusive messages. A byte in a MIDI message is a status byte if it is in the range of \$80-\$FF (128-255), and a data byte if it is in the range of \$00-\$7F (0-127).

Status → Record:

Starts the recording process.

Status → Stop:

Stops the recording process.

The panel below the menu contains the following items:

Record/Stop/Clear buttons:

These buttons duplicate the corresponding menu items.

Note/Note aftertouch/etc.:

These checkboxes determine which incoming MIDI messages are recorded. Checked means 'yes'.

8. Using the computer keyboard

FCB1010 Manager's user interface uses mostly standard widgets (buttons, checkboxes, pull-down boxes etc.). This means that it may sometimes be easier to use the keyboard instead of the mouse for particular operations.

For Windows, the following standard keystrokes are worth mentioning:

| Control | Key(s) | Action |
|---------------|-------------------|-------------------------------|
| <i>Any</i> | Tab | Select the next control |
| | Shift+Tab | Select the previous control |
| Checkbox | Space | Toggle the setting on/off |
| Pull-down box | Left/Up arrow | Select the previous item |
| | Right/Down arrow | Select the next item |
| | Home | Select the first item |
| | End | Select the last item |
| | Alt+Up/Down arrow | Open/close the pull-down list |

9. Known problems

MIDI Thru (Windows only):

FCB1010 Manager's MIDI Thru feature only passes on *short* MIDI messages, i.e. any message *except* SysEx (System Exclusive).

This is because FCB1010 Manager achieves its MIDI Thru feature by simply calling the `midiConnect` function in Windows' `MMSYSTEM` library: basically Windows handles all Thru traffic behind FCB1010 Manager's back, but unfortunately `midiConnect` doesn't pass on SysEx messages. (Incidentally, the Huskervu utility's MIDI Thru feature doesn't pass on SysEx messages either, so it probably uses `midiConnect` as well!)

I may try to find a work-around for this in a future version of FCB1010 Manager. In the meantime you should use MIDI-OX if you need to pass on SysEx messages via a MIDI Thru connection. (Apparently MIDI-OX doesn't use `midiConnect`, but handles all MIDI Thru traffic manually, which might actually be marginally slower than `midiConnect`, for non-SysEx messages that is...)

USB MIDI ports:

While FCB1010 Manager is running, connecting or disconnecting a MIDI device via its USB cable must be avoided, since it may lead to nasty error messages; instead, you must exit and restart FCB1010 Manager manually. I've been working on a fix, but I don't know if and when this will be made available.

Window widths:

If the screen dimensions are too small, big windows of *fixed* size can get cut off. Normally you're safe with a screen of 1024×768 pixels, but you can run into problems when you decrease the screen size of a virtual machine running FCB1010 Manager.

Alternative DPI settings:

Nearly all screen elements scale correctly under alternative DPI settings. However, the bitmaps used in the pull-down menus *don't* scale, which leads to rather cramped-looking pull-down menus at high enlargement DPI settings, because the heights of the menu items follow these bitmaps instead of the menu items' *names* (which *do* scale).

macOS only:

Several shortcomings of the program's GUI (graphical user interface), such as:

- Scrollbars around grids don't go away once they are there: once you've made such a scrollbar appear by making the window smaller, it remains, even when you re-enlarge the window.
- Numbers in editboxes don't align to the right but to the left. (Even worse, a long number may initially be partly hidden behind the left edge of the editbox.)

All these problems are caused by bugs in the GUI library the program uses on macOS, so at the moment there's not much I can do.