

PROMIX 01 MANAGER



Version 1.5.1

MANUAL

Copyright © 2020 by Mark van den Berg

MOUNTAIN UTILITIES



<https://mountainutilities.eu/>

CONTENTS

1. Overview	3
2. Version history	4
3. Computer requirements	6
4. Installation of ProMix 01 Manager	7
5. MIDI setup	8
6. The main window	12
7. The ProMix 01 window	18
8. The MIDI input messages window	20
9. The MIDI controllers window	25
10. Known problems	27

1. Overview

ProMix 01 Manager is a utility by Mountain Utilities for working with the Yamaha ProMix 01 digital mixer.

It provides full instant control of the ProMix 01's channels, effects, compressors and oscillator, and you can save/open all these settings to/from files.

Note that ProMix 01 Manager does not support the ProMix 01's 'scenes'.

ProMix 01 Manager is only available for Windows. (A future Linux edition is unlikely, but not impossible. There are no plans for a macOS edition.)

ProMix 01 Manager is free, although donations are very welcome.

2. Version history

Version 1.5.1 (2020-11-17)

- The application has become more user-friendly concerning data errors in ‘.stp’ and ‘.mru’ configuration files:
 - On startup, when the application encounters a data error in a configuration file, the application now mentions the exact name of the offending file, and no longer refuses to start. Hence it is no longer necessary to delete corrupt configuration files manually.
 - On exit, the application reports any errors while saving configuration files.
- The update check mechanism works again on Windows versions supporting TLS 1.2 but not TLS 1.3, i.e. Windows 7, Windows 8 and pre-TLS 1.3 Windows 10. (Windows XP and Vista (which don’t even support TLS 1.2) were not hit by this problem, and still work too.)

Version 1.5.0 (2020-09-14)

- This manual is no longer a stub but fully featured.
- The update check mechanism should now work again on any operating system capable of running the application, including Windows XP.
- The installer allows you to deselect non-essential files: the manual and the skin definitions.
- Increased the application’s running speed by disabling some debugging safeguards. (From now on, this will apply to all Release Candidate and Release versions.)
- Skins: 39 alternative user interfaces, mainly color schemes.
- Full support for the ‘Per Monitor v2’ DPI awareness protocol introduced in Windows 10 Creators Update 1703: at high DPI settings all text characters look razor-sharp instead of blurry, and the resolution of each window automatically adapts to the monitor on which it is (mainly) being shown.
- Fixed the incorrect scaling of certain windows and dialog boxes if the operating system’s DPI setting isn’t its normal value of 96 DPI. (This problem was introduced in ProMix 01 Manager 1.4.0 Alpha 1.)
- To always show the *full* application title on the main window’s title bar, the width of the main window now takes the minimize, maximize and close buttons into account. This is especially relevant to Windows 10: its huge buttons previously caused the application title to be truncated.

Version 1.4.0 Alpha 1 (2018-06-14)

- The installer automatically installs the edition of ProMix 01 Manager matching 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.)
- The ProMix 01 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.

Version 1.3.0 (2016-03-26)

- The positions of the ProMix 01 window and its four parameter windows are restored upon program restart.
- You can switch between different setups for the program (including MIDI I/O device states). (This requires restarting the program.)
- You can switch between different window layouts (‘desktops’). (This does not require restarting the program.)
- In the ProMix 01 window, the ‘Default parameters’ operation (accessible from the Window pull-down menu) first asks whether you really want to do this.

- Hiding the ProMix 01 window no longer leads to an offer to save any changed parameters. Instead, this is now only done upon termination of the program.
- In the MIDI controllers window, the control bars respond to incoming Control Change messages.
- ProMix 01 Manager no longer refuses to start if system.ini doesn't exist in the Windows system folder.
- ProMix 01 Manager uses the new Mountain Utilities web site 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)

- A small memory leak was fixed.
- Several improvements to the user interface.

Version 1.2.0 (2011-01-11)

Version 1.0.2 (2009-11-02)

Version 1.0.1 (2009-08-06)

Version 1.0.0 (2009-07-11)

First published version.

3. Computer requirements

To run ProMix 01 Manager, your computer must comply with the following requirements:

- Processor: Any Intel 80486- or Pentium-compatible CPU. Processor speed is relatively unimportant. The number of cores is irrelevant, since ProMix 01 Manager only uses one core.
- Operating system: Windows 2000 or later (XP, Vista, 7, 8, 10 etc.).
- An SVGA-compatible graphical card and monitor:
 - The screen size should be at least 800×600 pixels. (However, some of the bigger windows and dialog boxes may be cut off if the screen is too narrow, so 1024×768 is the *practical* minimum.)
 - 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 10 MiB.
- RAM: when running, ProMix 01 Manager normally occupies roughly 10 MiB.
- ProMix 01 Manager runs perfectly without a MIDI link to an actual ProMix 01, but if you want to establish a real-time connection between ProMix 01 Manager and a ProMix 01, the ProMix 01 must be linked to the computer via MIDI input and output ports, nowadays usually on a MIDI-to-USB device.

4. Installation of ProMix 01 Manager

Three editions of ProMix 01 Manager are available: an installer, a 32-bit portable edition and a 64-bit portable edition:

Installer:

1. Download pm01man-a.b.c-install.exe (where *a.b.c* stands for the actual version number) to your computer from the ProMix 01 page at the Mountain Utilities web site (<https://mountainutilities.eu/promix01>).
2. Run pm01man-a.b.c-install.exe and follow its instructions. The installer automatically installs the edition of the actual application (PM01Man.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 → ProMix 01 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 ProMix 01 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 pm01man-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 ProMix 01 page at the Mountain Utilities web site (<https://mountainutilities.eu/promix01>). The ‘x64’ (64-bit) edition only runs on 64-bit Windows, the ‘x86’ (32-bit) edition on 32- and 64-bit Windows.
2. Unzip pm01man-a.b.c-xnn-portable.zip completely (maintaining the zip file’s tree structure) to any folder to which ProMix 01 Manager itself (PM01Man.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 ProMix 01 Manager would *not* have write-access.

Running ProMix 01 Manager itself

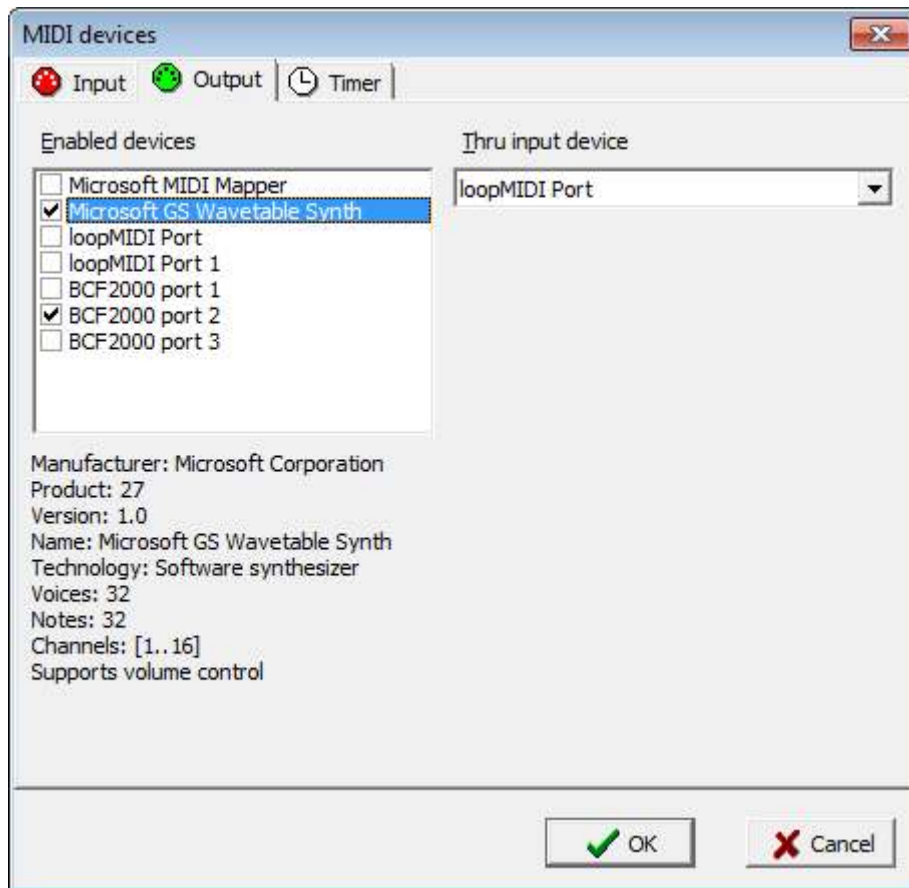
After installation, you can start ProMix 01 Manager (PM01Man.exe) itself, e.g. via the Windows start menu.

If you have never run ProMix 01 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 (PM01Man.ini) 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

You must tell ProMix 01 Manager manually where to find your ProMix 01. This involves the following steps:

1. You must ensure that the pertinent MIDI I/O ports are *enabled* in ProMix 01 Manager:
From the main window's Options pull-down menu, 'MIDI devices' opens a dialog box in which you can select the MIDI I/O devices to which ProMix 01 Manager connects:



Enabling the proper I/O devices is relevant in several ways:

- If you want ProMix 01 Manager to communicate with a ProMix 01, the MIDI input and output ports connected to this ProMix 01 need to be 'enabled' here.
- More in general, *all* ProMix 01 Manager's MIDI utilities (including those not directly related to ProMix 01s) can only exchange data via MIDI devices that are 'enabled'.
- 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 ProMix 01 Manager, in order to avoid access conflicts.

Tip: In the MIDI devices 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.

2. You must indicate to which MIDI output device the ProMix 01 is connected and what the value of its basic channel is:
From the ProMix 01 window, open the 'ProMix 01 options' dialog box via the MIDI pull-down menu, select the MIDI output device to which the ProMix 01 is connected, and match the basic channel to the value used by the ProMix 01 itself. See §7 for further discussion.

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** (<https://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 at best, and may even lead to communication errors.

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 the program's communication with the MIDI device; this can result in many types of error.

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. ProMix 01 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 ProMix 01 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 ProMix 01 Manager's MIDI devices dialog box, then any MIDI data sent to 'Out To MIDI Yoke: 1' not only comes back to ProMix 01 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 1.75 (but *not* with MIDI Yoke *for Windows 95/98/Me*): closing any MIDI Yoke NT *input* port causes a delay of 1 second. (Certain earlier versions even 3 seconds.)

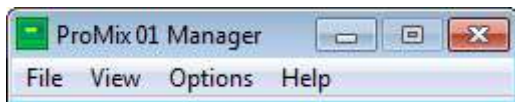
Concerning ProMix 01 Manager this is mainly relevant during program exit. In principle ProMix 01 Manager terminates almost instantly upon exit, but when all the input ports of MIDI Yoke NT 1.75 are enabled, termination of ProMix 01 Manager takes some 16 seconds longer than normal! Therefore you should disable as many MIDI Yoke NT input ports as possible in ProMix 01 Manager's MIDI devices dialog box, i.e. any MIDI Yoke NT input ports that ProMix 01 Manager itself doesn't use. (Note that you can still use any MIDI ports disabled in ProMix 01 Manager in *other* programs!)

To help you avoid some of the serious problems discussed above, ProMix 01 Manager takes the following steps:

- On *first* startup, if ProMix 01 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 ProMix 01 Manager's exit procedure), unless some other program (e.g. MIDI-OX) is routing a ProMix 01 through a MIDI pipe.
- On *every* startup, ProMix 01 Manager optionally warns you if any MIDI Yoke NT input ports are enabled and thereby cause extra delays during termination of ProMix 01 Manager. You can enable/disable this warning on the Input tab of the MIDI devices dialog box.

6. The main window

ProMix 01 Manager's main window only consists of a pull-down menu:



This menu provides the following operations:

File → Restart:

Terminates ProMix 01 Manager and starts a new instance of it. Restart behaves exactly like Exit (q.v.) in that you will be asked whether you want to save any unsaved parameter changes before termination.

File → 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 PM01Man.stp is deleted before the restart, so that all ProMix 01 Manager's setup values (such as MIDI I/O device settings and window positions/sizes) are restored to their defaults. Consequently, ProMix 01 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 PM01Man.stp but in PM01Man.mru. If you want to clear these file lists, simply use their 'Clear list' operations, or (for a total clearance) remove PM01Man.mru while the application isn't running.

File → 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 ProMix 01 Manager to use a specific setup file via the command line, as follows:

/s setupfile

File → Exit:

Terminates ProMix 01 Manager. If you have made any unsaved changes to ProMix 01 parameters, the application will prompt you to save all parameters to a ProMix 01 parameter file (with extension '.plp').

Note that 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 ProMix 01 Manager by clicking on

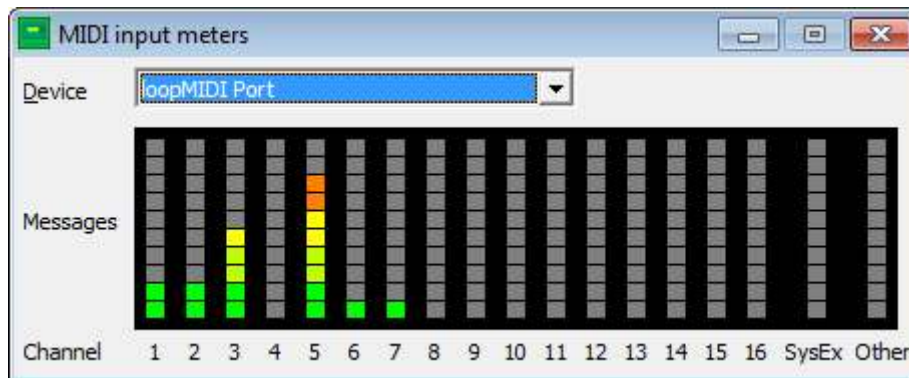
the **X** icon on the main window's title bar. Pressing **Alt+F4** also works, but (unlike **Alt+X**) only from the main window.

View → ProMix 01:

Opens the 'ProMix 01' window. This is the gateway to the main functionality of ProMix 01 Manager. See §7 for more information.

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.

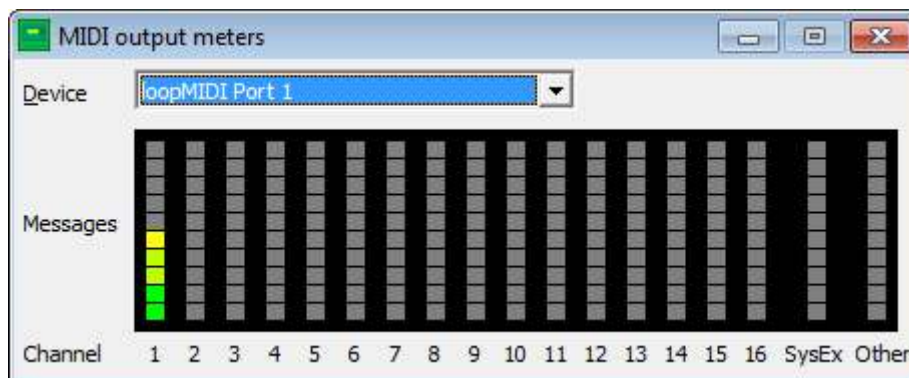
'DISABLED' in front of the selected MIDI input device indicates that the device is disabled, so no MIDI messages can currently be received from that device. (You can enable devices in the MIDI devices dialog box.)

View → MIDI → Input messages:

Opens the MIDI input messages window. Here you can record and view messages from the MIDI input devices. See §8 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. See §9 for more information.

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 File → 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, ProMix 01 Manager’s main window stays on top of any other windows belonging to ProMix 01 Manager. The main window also stays on top of other applications (except of course those that have the stay-on-top property too).

View → Skin:

Opens a window in which you can select the ‘skin’ of the application; apart from the default skin (called ‘Windows’), there are 39 alternative skins. Most prominently, a skin defines the colors of all the visual elements of the application’s windows, but a skin may also change the font type and/or size, or the shape or size of buttons etc.

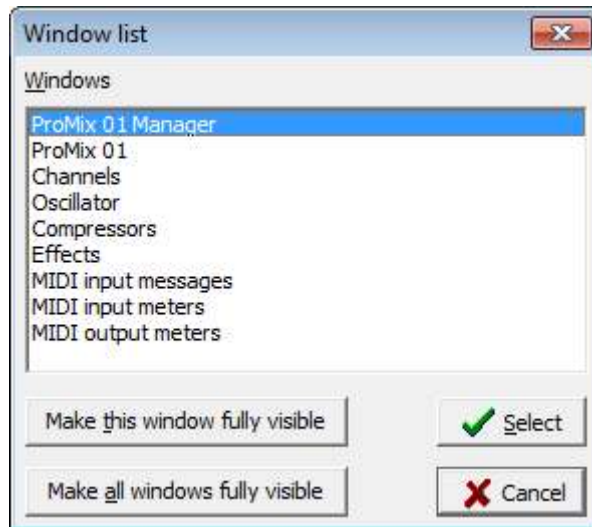
Some tips:

- The default ‘Windows’ skin isn’t really a skin: it’s simply the current interface of the Windows version you are running. So for instance if your computer is using an Aero theme on Windows 7, that’s what the ‘Windows’ skin will give you; but on Windows 10 it will amount to the standard Windows 10 look.
- The six ‘Windows 10 ...’ skins are available on any Windows version, so you can make your Windows XP or 7 look like Windows 10! (However, there are subtle differences between these skins and the *actual* Windows 10 interface.)

Disclaimer: The 39 alternative skins were not developed by Mountain Utilities, but are part of the Delphi programming environment on which this Mountain Utilities application is built. So these skins are offered on a take-it-or-leave-it basis. Some of these skins may cause undesired visual effects, such as certain screen elements disappearing behind others. However, due to the sheer number of these skins it would be a huge task to fine-tune every window of every Mountain Utilities application for every skin. So be prepared for some unpleasant and perhaps confusing surprises. If you encounter a ‘really bad’ problem, feel free to report it in the application’s forum at the Mountain Utilities web site.

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.

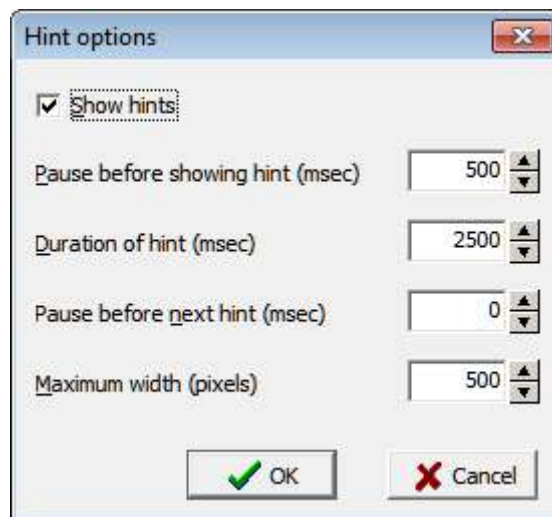
If the highlighted window is partially or completely outside the current monitor(s), you can move it into full view by pressing 'Make this window fully visible'. 'Make all windows fully visible' performs this operation on *all* windows in the list.

Options → MIDI devices:

Opens a dialog box in which you can configure the MIDI devices that ProMix 01 Manager monitors. See §5 for more information.

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.:

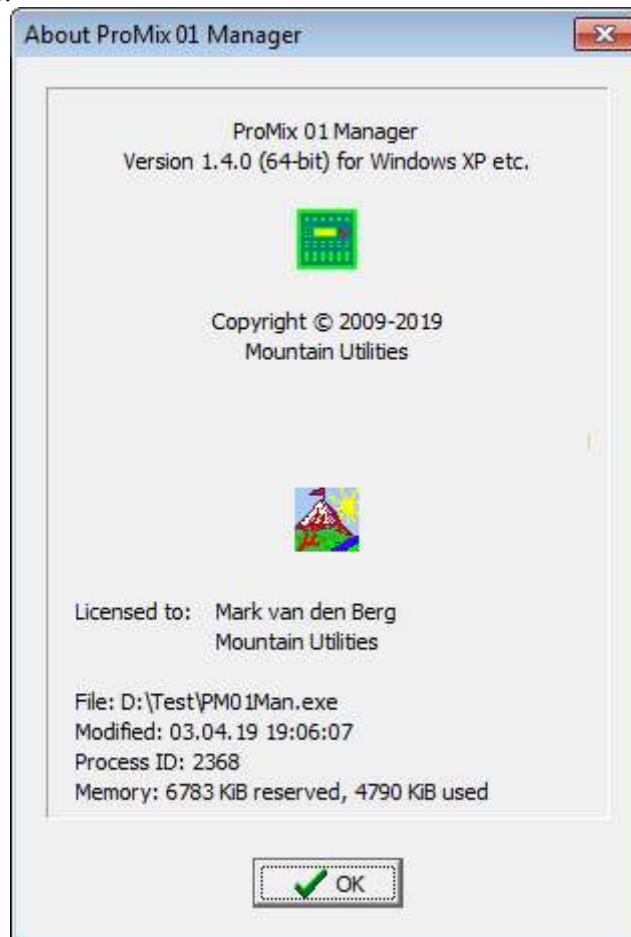


Help → Manual:

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

Help → About ProMix 01 Manager:

Opens a dialog box containing information on ProMix 01 Manager, such as its version number and memory usage:

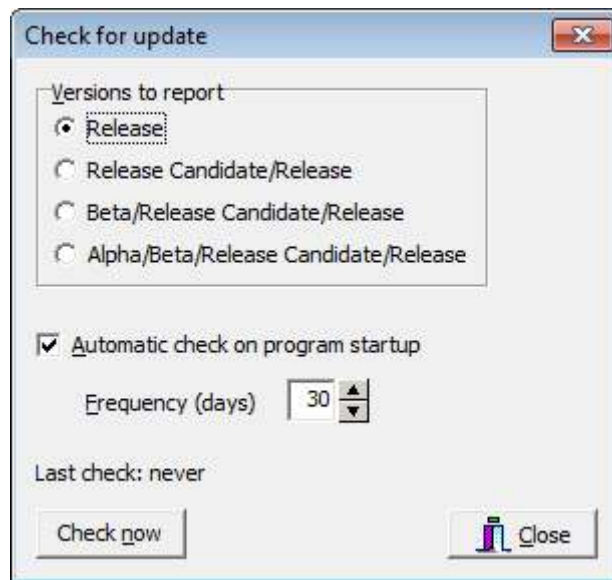


Help → Mountain Utilities web site:

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 ProMix 01 Manager and other matters related to the ProMix 01.

Help → Check for update:

Opens a dialog box in which you can set the frequency at which ProMix 01 Manager automatically searches for updates at the Mountain Utilities web site, and 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 pertinent 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 ProMix 01 Manager and support its further development by making a donation.

7. The ProMix 01 window

The ProMix 01 window only consists of a pull-down menu and a statusbar:



The statusbar displays the file that was most recently opened or saved via File → Open/Save/Save As.

The menu provides the following operations:

File → Open:

Opens a dialog box from which you can open a file (with extension ‘.p1p’) containing all ProMix 01 parameters.

File → Save:

Saves all ProMix 01 parameters to the file (with extension ‘.p1p’) shown on the statusbar. If you haven’t opened or saved a parameter file before, this defers to the Save As operation (see below).

File → Save As:

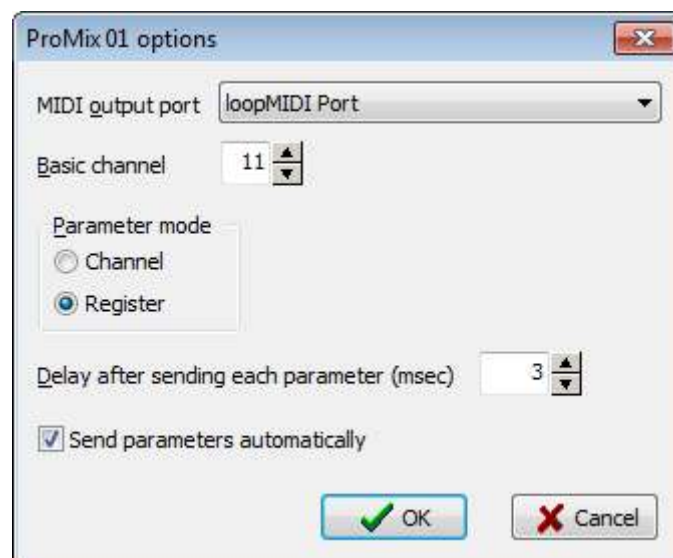
Opens a dialog box from which you can save all ProMix 01 parameters to a file (with extension ‘.p1p’).

MIDI → Send all parameters now:

Sends all ProMix 01 parameters to the MIDI output port defined in the ‘ProMix 01 options’ dialog box (see below), provided that this port is enabled in the ‘MIDI devices’ dialog box.

MIDI → Options:

Opens the ‘ProMix 01 options’ dialog box:



- MIDI output port:
This must be set to the output port to which the ProMix 01 is connected.
- Basic channel:
This must match the basic channel defined on the ProMix 01 itself.
- Parameter mode:
This determines which type of message ProMix 01 Manager uses to send parameters to the ProMix 01.
 - ‘Channel’ sends non-standard Control Change messages via many MIDI channels.
 - ‘Register’ sends NRPN messages via the Basic channel.
 Channel is more economical than Register. However, Channel can have more unwanted side-effects, which is why Register is the default.
- Delay after sending each parameter (msec):
This aims to give the ProMix 01 time to process each parameter change.
- Send parameters automatically:
If this is setting is on, ProMix 01 Manager automatically sends any parameter change you make to the ProMix 01 immediately.

Window → Channels/Compressors/Effects/Oscillator:

Opens the corresponding window.

Each of these windows contains a particular ProMix 01 parameter set.

To select a parameter, you can mouse-click on its field, or you can navigate to its field by Tab (=forward) or Shift+Tab (backward).

To change the value of the selected parameter, use the Up or Down arrow key on your computer keyboard.

(A future version of ProMix 01 might include knobs and faders, but that would necessitate redesigning the layouts of these windows, in particular the Channels window. The nice thing about the current layout of the Channels window is that it shows *all* channel parameters simultaneously, which would be difficult with knobs and faders.)

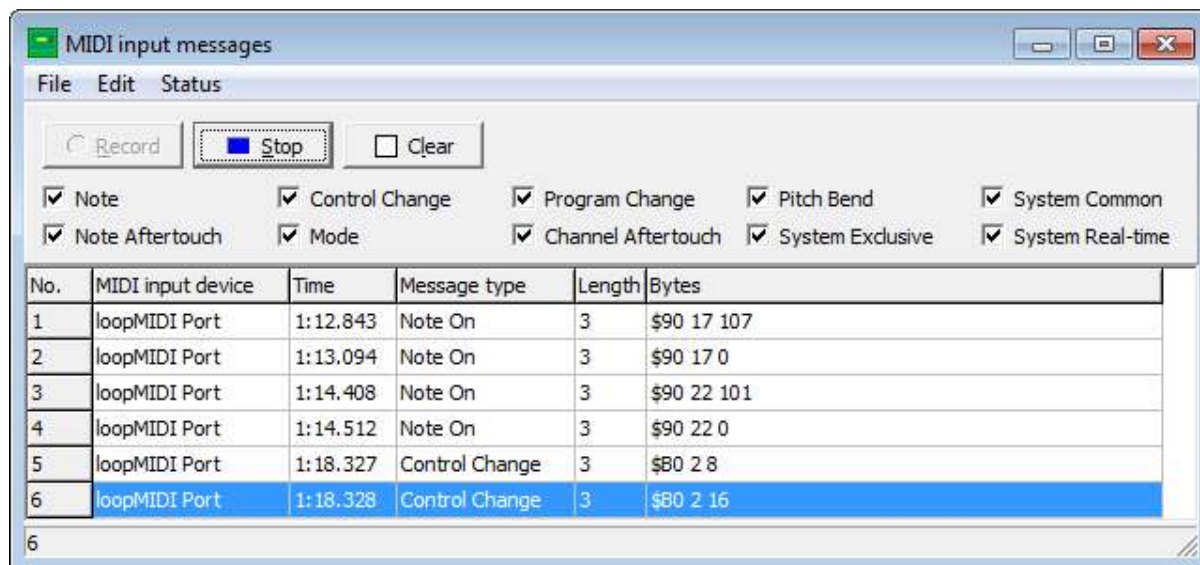
Window → Default parameters:

Resets all ProMix 01 parameters in the Channels, Compressors, Effects and Oscillator windows to their default values, and sends all these values to the connected ProMix 01.

Since this is a ‘momentous’ decision, a dialog box asks you whether you really want to do this.

8. The MIDI input messages window

The MIDI input messages window is accessible from the main window in two ways: its toolbutton (third from left) and the View pull-down menu (→ MIDI → Input messages):



The MIDI input messages window allows you to record and view MIDI messages sent to ProMix 01 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 of a ProMix 01's output messages. You can also copy recorded messages to the 'MIDI message clipboard' for inclusion in 'Custom' ⇒ 'Bytes' definitions of ProMix 01 elements, and you can 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 ProMix 01 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 ('ProMix 01 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 ProMix 01 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:

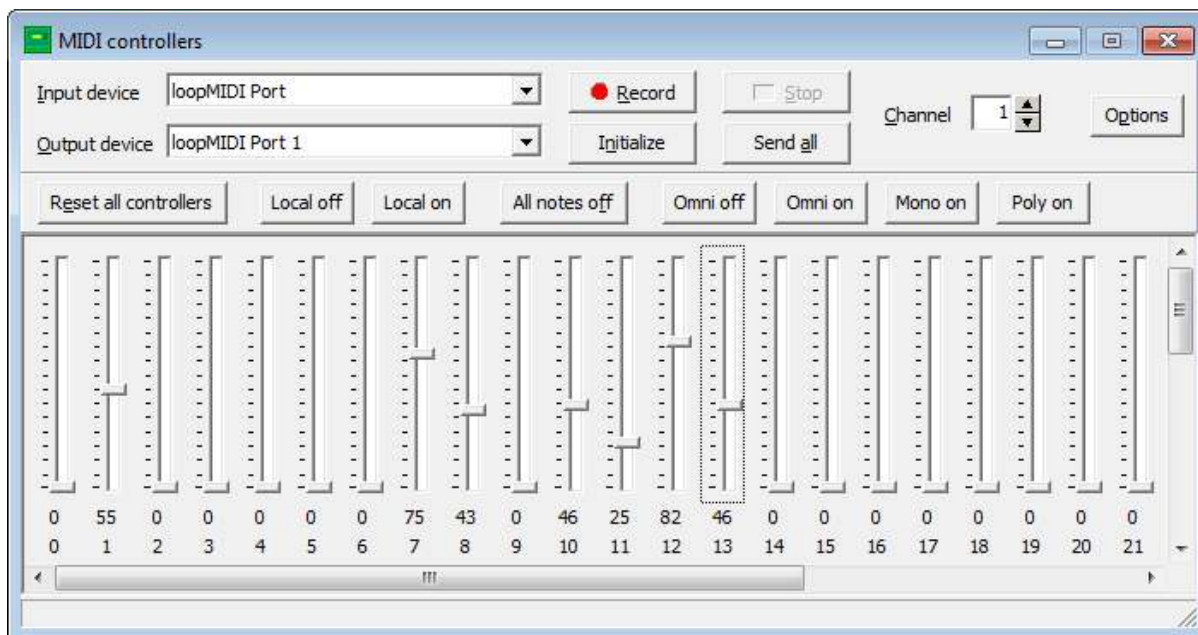
These buttons duplicate the corresponding menu items.

Note/Note aftertouch/etc.:

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

9. The MIDI controllers window

The MIDI controllers window is accessible from the main window via the View pull-down menu → MIDI → Controllers.



This window allows you to send MIDI Control Change messages and to monitor incoming MIDI Control Change messages.

Each of the 128 MIDI controllers (0-127) has its own fader:

- Moving a fader (via mouse or computer keyboard) changes the value of the controller and outputs the corresponding MIDI Control Change message on the selected MIDI channel to the selected MIDI output device.

Moving a fader via the computer keyboard works as follows:

- Arrow Up: +1
- Arrow Down: -1
- Page Up: +8
- Page Down: -8
- Home: set 127 (maximum)
- End: set 0 (minimum)
- If the Record button has been pressed, the faders react to incoming messages on the selected MIDI channel of the selected MIDI input device. To stop recording, press the Stop button.

The functions of the other buttons:

- The Initialize button sets all visible controllers (as defined in the Options dialog box, described below) to 0.
- The 'Send all' button sends the values of all visible controllers (as defined in the Options dialog box).
- The buttons labelled 'Reset all controllers', 'Local off' etc. send particular mode messages (using controllers 121-127). For instance, pressing 'Reset all controllers' outputs a Control Change message using controller 121 with a value of 0. Note that the corresponding fader is updated accordingly.

- The Options button opens a dialog box called ‘MIDI controllers: options’, in which you can customize the following things:
 - Whether the mode message buttons (‘Reset all controllers’, ‘Local off’ etc.) are shown.
 - How big the faders of the controllers are.
 - How many controllers are visible per row.
 - Which controllers are visible. (The buttons on the right of the dialog box are ‘macros’, selecting particular ranges.)

10. Known problems

MIDI Thru:

ProMix 01 Manager's MIDI Thru feature only passes on *short* MIDI messages, i.e. any message *except* SysEx (System Exclusive).

This is because ProMix 01 Manager achieves its MIDI Thru feature by simply calling the `midiConnect` function in Windows' `MMSYSTEM` library: basically Windows handles all Thru traffic behind ProMix 01 Manager's back, but unfortunately `midiConnect` doesn't pass on SysEx messages.

I may try to find a work-around for this in a future version of ProMix 01 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...)

MIDI input:

If an incoming SysEx message is followed immediately by a channel message, the channel message may be processed first. This may be fixed in a future version of the application.

USB MIDI ports:

You should not connect or disconnect a MIDI-to-USB device via its USB cable while ProMix 01 Manager is running, since this may lead to nasty error messages; instead, you must exit and restart ProMix 01 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 ProMix 01 Manager.