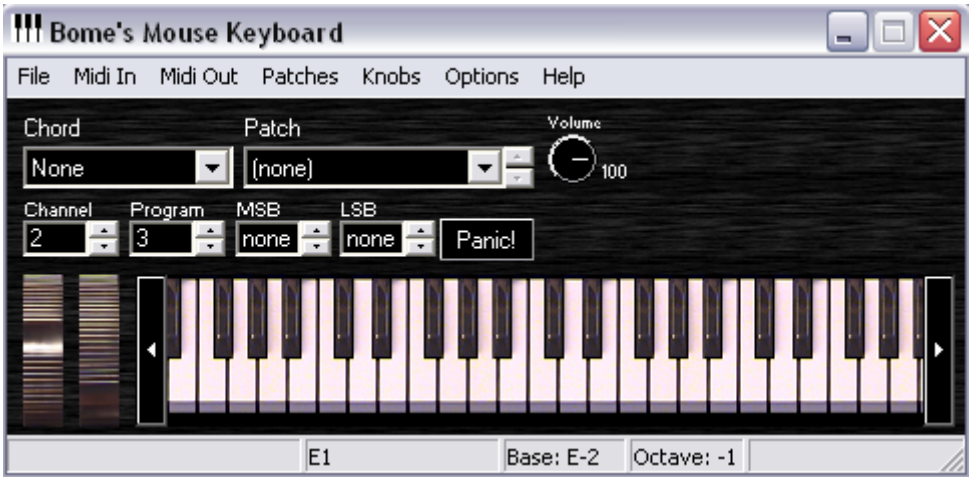
MIDI

Mouse Keyboard
MIDI Yoke /MIDI-OX

Bome's Mouse Keyboard



V2.00 beta 6 OVERVIEW This program primarily simulates a MIDI keyboard which you can play with the mouse or the computer keyboard. You can control your MIDI devices without having a real keyboard - for example you can play the internal synth of your sound card. With a virtual MIDI cable (for example Hubis Loopback Device or MIDI Yoke) you can play any sequences into your sequencer or route your external synthesizer through the chord-maker of Bome's Mouse Keyboard. Additionally, you can define as many knobs as you want and control with them any MIDI controller. SYSTEM REQUIREMENTS - Windows 95/98/ME/NT4/2000/XP - Some kind of MIDI connection/synthesizer/soundcard Optionally supported: - Mouse with wheel - Joystick

QUICK START

1. Select a MIDI out device in the menu. Everything that you play will be sent to that MIDI device.
2. Now you can already play: Use the left mouse button or the keyboard keys. A right click will play in Pitching Mode: While the right mouse button is pressed, you can alter the pitch/modulation by moving the mouse horizontally/vertically.
3. Adjust the size of the keyboard by
   - resizing the window

   - changing the base note (Alt+Mouse drag)
   - adjust the key width (in the Settings Dialog)
4. Change the chord and play: Ahhh !
5. To find a good patch/program, enter the Hold Mode by playing a note with CTRL+right mouse button. The note will keep on being played when you release the mouse button. Now, change the program number or patch - it will replay your note for each new program so it's made easy browsing through the programs. The same applies to the bank numbers or chords. Which bank numbers to use is detailed in the documentation of your MIDI device.
6. Change the master volume with the volume knob. You can add more knobs in the menu "Knobs|Set Up...".

TUTORIALS For more tutorials, see http://www.bome.com/midi/keyboard/help/

ADVANCED FUNCTIONALITY

*** not all beta functions included yet in this description ***

When you have a real keyboard connected to your computers MIDI port, select this as MIDI in and check the menu Options|Midi thru. Like this, the keys you play on the real keyboard are visualized on the mouse keyboard and the note and any chords you selected are routed to the MIDI out device.

To use several mouse keyboards at once (with their own settings and knobs) create a copy of MouseKeyboard.exe under a new name, e.g. SynthKeyb.exe. SynthKeyb will then use its own set of settings.

Mouse Operations

There are several modes to play with the mouse:

- Left mouse button: "Normal Mode": the note/chord that you press down will be played. Moving the mouse while having the mouse button pressed will always play the new note/chord and release the old one.
- Right mouse button: "Pitching" mode: only the initially pressed note/chord will be played. With moving the mouse while the right mouse button is still pressed you can alter 2 controllers - one is changed by moving up/down, the other one by moving left/right. When you release the right mouse button, the note/chord will stop playing. You can set these controllers in the Settings dialog. By default, you change pitch and modulation wheel.
- CTRL+mouse button: "Hold Mode": When you release the mouse button while holding down CTRL, the note/chord will keep on playing. The note will not be released before you play another note with the mouse. While a note is on hold, you can change patch/program, chords, etc. to browse through them: each time you change something, the note will be replayed with the new patch/program/chord.
- SHIFT: Pressing SHIFT while playing will lower the velocity of the played notes/chords. The velocity is the dynamic volume of the key. It differs from the general volume in that it does not change the volume of anything else played on that synthesizer - just the notes

played with the mouse are affected. You can set the amount of lowering in the Settings dialog.

Other mouse operations:

- **ALT+left mouse button on the keyboard: Drag the mouse keyboard** left/right to change the view of keys.
- **Wheels** (at left of keyboard): You change the value by dragging the mouse up and down. Holding SHIFT while dragging is the "Fine Mode" which will give fine control. Double clicking the wheel will revert it back to its default position. For the pitch wheel, holding CTRL while releasing the mouse button will keep the wheel at its position without rolling back into the middle position. You can change the controller of the right wheel in the Settings dialog.
- **Mouse Wheel:** (the little wheel on the mouse itself - where present): You can change a controller with the mouse wheel. Holding down shift while rolling is the Fine Mode as with operating the wheels. The controller which is changed with the mouse wheel can be chosen in the Settings Dialog.
- **Knobs:** They work exactly the same way as the wheels (including Fine Mode with SHIFT and double click). However, the knobs will be selected (blue on selection) and grab the mouse wheel while selected. So rolling the mouse wheel while a knob is selected, will always change the knob controller. Once a knob is selected, you can also change its value with the cursor up/down and page up/down keys.

COMPUTER KEYBOARD OPERATIONS

The computer keyboard has 2 manuals mapped to the notes: The first manual occupies the first 2 rows: the 1st row ("23567..." on English keyboard layout) are black keys, while the second row ("qwerty..." on English layout) are the white keys. The second manual is used analogously with the third and fourth rows of the computer keyboard. It is one octave lower than the first manual. By default, the first white key is mapped to note "C", i.e. keys Q and Z on the English keyboard layout. You can change this in the Settings Dialog.

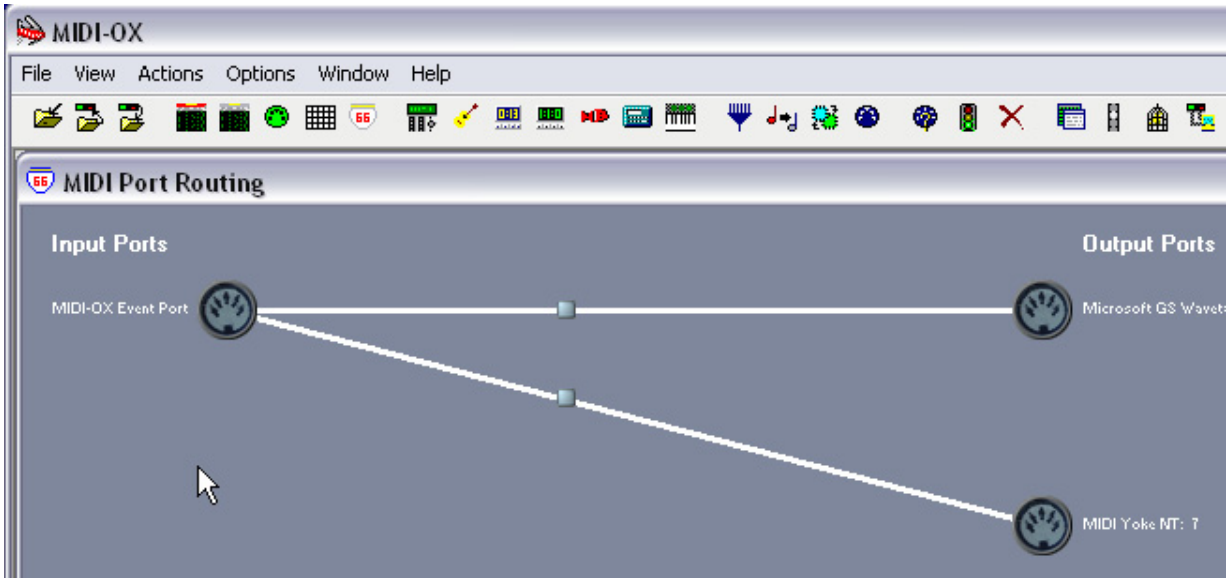
Playing with SHIFT pressed down will lower the velocity just as it does when playing with the mouse.
There are a number of key shortcuts that you can use: (SUBJECT TO CHANGE)

chord +/-	Pos1/End
patch +/- :	PageUp/PageDown
program +/- :	Shift+PageUp/Shift+PageDown
Panic :	Ctrl+Space
Transpose :	Ctrl+T
Settings :	Ctrl+S
Knob Editor :	Ctrl+K
Chord Editor :	Ctrl+E
Choose Patch set :	Ctrl+P
Octave +/- :	F3/F4
Apply Snapshot :	F5...F12
Store Snapshot :	Ctrl+F5....Ctrl+F12

MIDI yoke

Bedienung über MIDI-OX

MIDI Yoke MIDI OX http://www.midiox.com/



Description

MIDI YOKE is a MIDI Patch Cable driver.

MIDI YOKE is used to connect any Windows MIDI Application outputs to any other Applications inputs. The MIDI data stream is passed directly from output to input -- Example:

[Sequencer *Out* **]==>[** *Out* **MIDI Yoke** *In* **]==>[** *In* **MIDI-OX** 32 **]**

Another:
[MIDI Keyboard *In***]==>[***In* **MIDI-OX** *Out***]==>[***Out* **MIDI Yoke** *In***]==>[***In* **Sequencer]**

This allows you to connect the MIDI output from one program to the MIDI input of a different program. MIDI Yoke can be configured to provide a varying number of MIDI Ports (from 1 to 16). In addition, each port allows multiple opens of both input and outputs: up to 4 openings per port. This flexibility provides for almost any configuration imaginable. **Note:** The NT driver is fixed at 3 ports and 3 openings for the moment.

There is additional information in the **FAQ**.

MIDI Feedback

The powerful nature of MIDI Yoke requires that a bit of care be exercised in its use: do not connect the outputs of one port to the same number inputs within a single application. If you do, it will cause MIDI feedback -- this phenomena will bring a computer to its knees, and likely crash the system.

Version 1.50+ of MIDI Yoke attempts to detect MIDI feedback by any of 3 configurable methods. The **first** method was supported in previous versions: it simply sends an undefined controller number out the input port every so often (once every 50 messages). It looks at the output port to see if the controller shows up: if it does it assumes MIDI feedback and disables the driver. The **second** (new) method analyzes the rate of messages traveling through the driver. If they exceed a certain threshold (2048 messages per second by default), the driver assumes MIDI feedback and disables the driver. The **third** (new) method combines the previous two: it analyzes the data rate. If the rate exceeds the threshold, the driver sends an undefined controller out the input. If the controller shows up at the output, MIDI feedback is detected, and the driver is disabled.

To recover from any of these conditions, simply close all connections attached to both ends of the particular MIDI Yoke port. After that the port may be reopened (but common sense would dictate that you should configure the routing differently).

Using MIDI Yoke

MIDI Yoke is ***not*** a normal driver! It is a like a MIDI Patch Cable: it takes any data that shows up at its MIDI Yoke Output port and spits it out of its MIDI Yoke Input port! This seems to be really counterintuitive to people, but it's the only way that it can work as a patch cable. MIDI Yoke is used to attach different MIDI applications together -- it links them via MIDI.

This also means that you should **never**, **ever**, open the same MIDI Yoke port number as both input and output within a single application -- it is suicidal to do so. When this happens, any MIDI message sent to the output port is immediately routed back to the input port, which is then sent out the output port and immediately routed back to the input port... I think you get the idea: **MIDI Feedback**! This all happens very, very fast (at interrupt time), and if it is not stopped, your machine will appear to be hung. MIDI Yoke attempts to detect this condition and will break the connection when it is detected.

After a MIDI feedback report, you need to close and reopen the driver in each application that has opened it. You can usually do this by opening the applications MIDI devices dialog (MIDI-OX and Cakewalk each have these), unselecting the MIDI Yoke port, and Press OK. The driver is now closed. Now do the same in the other application (if there's more than one application). Next figure out how you want the driver to be routed (avoiding feedback scenarios). Open the MIDI devices dialog and reselect the MIDI Yoke driver. Press OK. Do the same in the second application. The driver should now be re-enabled and operational.

You can use MIDI Yoke to communicate between applications by having then share a single MIDI Yoke port. The first application (the Master) opens a MIDI Yoke port as an **Output** port, the second application (the Slave) opens the same numbered MIDI Yoke port as an **Input** port. With this configuration, MIDI data (and System Exclusive) is passed in one direction from Master to Slave. The Master might be Cakewalk or Cubase or MIDI-OX. The Slave might be Rebirth or Cakewalk or MIDI-OX: it depends on what you want to do.

MIDI Connections

MIDI Yoke is normally used to connect two applications together. The first application will open **MIDI Yoke 1** as an *output* port, and the 2nd application will open **MIDI Yoke 1** as an *input* port. MIDI data will travel from the 1st application to the 2nd.

Single Client to Multi-Client: You can use MIDI-OX in conjunction with MIDI Yoke to achieve a virtual multi-client input. Launch an instance of MIDI-OX. In MIDI Devices... dialog, select the single client hardware driver as *input* and select **MIDI Yoke 1** as *output.* In *each* application that you wish to share the driver, open **MIDI Yoke 1** as *input*. Although I mention MIDI Yoke **1**, it can be any MIDI Yoke port as long as they all use the same port number.

Multi-Client to Single Client output: The reverse scenario is a mirror image of the previous paragraph. You can use MIDI-OX in conjunction with MIDI Yoke to achieve a virtual multi-client output. Launch an instance of MIDI-OX. In MIDI Devices... dialog, select **MIDI Yoke 1** driver as *input* and select your single client hardware port as *output.* In *each* application that you wish to share the output driver, open **MIDI Yoke 1** as *output*. Although I mention MIDI Yoke **1**, it can be any MIDI Yoke port as long as they all use the same port number.