

# mØxr β

## hello

**mØxr** is an ***Audio Signal to MIDI converter*** which is calibrated to respond to Reason/Record Control Voltage signals passed as audio signals. MIDI note on events are processed through a Thor Polysonic Synthesizer and routed to the Hardware Interface as a pair of signals. One path carries the Key Note data and the other path routes velocity/gate data in a manner that mimics CV/Gate pairing within Reason and Record.

The CV audio streams are decoded in MaxMSP and generate a series of monophonic MIDI note events which can be directed to other applications or hardware MIDI devices. This allows any sequential note information generated in Reason/Record to drive outside synthesizer sources.

Before opening any file, please follow the documentation on setting up your system. Use of the files without first understanding the configuration will result in some audible unpleasantness!

NOTE: This documentation is preliminary information and is subject to change if i don't get bored with this project.

**mØxr** β

# βeta

**mØxr** is an experimental concept for establishing a Control Voltage to MIDI interface between Propellerhead Reason/Record and MIDI hardware devices.

Outside of this documentation, there is no support for the Max Patch or other software required to operate the script.

The audio to MIDI converter is subject to unknown stability problems depending on your particular workstation configuration. This may result in hardware crashes related to audio driver problems. Use of this system comes with the risk possible data loss.

smiley face!

# mØxr β

## Requirements

### Mac OS

Reason 5 or Record+Reason Duo

MaxMSP 5 or MaxMSP 5 Runtime

Soundflower or some other inter-application audio routing system.

Max5 and Soundflower are available from:

<http://cycling74.com>

### Windows

Reason 5 or Record+Reason Duo

MaxMSP 5 or MaxMSP 5 Runtime

ASIO4ALL? or some other inter-application audio routing system.

—Need someone to Test

Max5 and Soundflower are available from:

<http://cycling74.com>

# mØxr β

## Initial Setup

Several steps are required to make this system function properly, and this is where things can get squirrelly. The following checklist describes the recommended process for the initial configuration.

1. Preparing to use **mØxr** requires that you first install an inter-application audio routing system such as Soundflower.
2. Confirm that the inter-application audio routing is properly working by testing it with other applications
3. Device drivers for the inter-application audio router and the hardware audio interface must be aggregated. This is easy to configure in MacOS using the Audio MIDI Setup Utility.
4. Record must be configured to use the Aggregate Device Driver so that audio can be sent to both the hardware interface and the **mØxr** - MaxMSP application.
5. Once the Aggregate Driver is functioning with Record, you can open the included CV to MIDI Record Session file. This file has CV directed to audio outputs of the Record Hardware Interface, and should not be opened until you have configured your audio outputs as described in this documentation
6. Launch the **mØxr** application and set the audio input and MIDI outputs to receive audio from Record and send MIDI to a hardware port.
7. Troubleshoot the inter-application audio settings or if you're lucky, start playing a MIDI synth from Record. Typically you must reset the audio source to Soundflower 16ch and the MIDI Out driver on the **mØxr** menus

# mØxr β

## Audio MIDI Setup - Aggregate Device

An *aggregate* device driver is a virtual system component composed of several other audio drivers available to the Mac OS system. In order to fully appreciate the functionality of mØxr, an Aggregate Device must be created combining your main audio hardware driver with Soundflower. To get started with this, launch Audio MIDI Setup in the Utilities folder of your application folder.

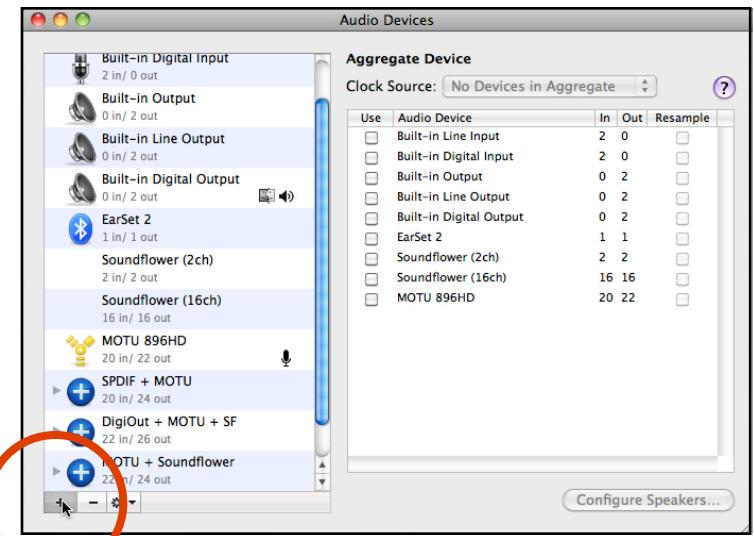


**Audio MIDI Setup**

# mØxr β

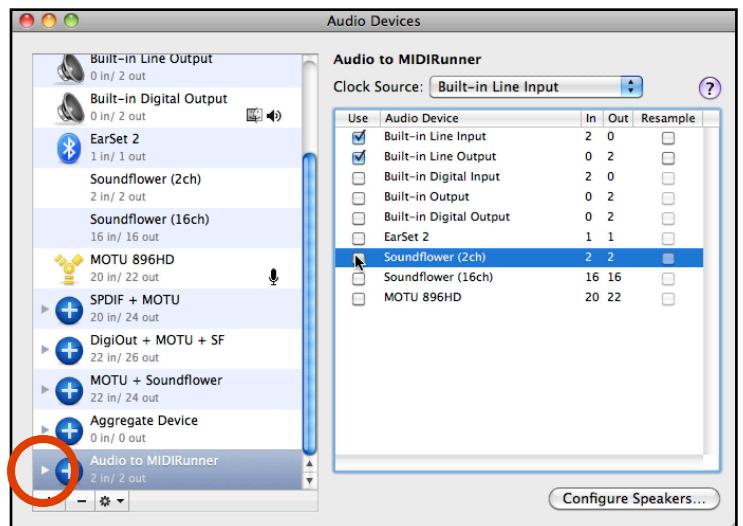
## Audio MIDI Setup

The Audio MIDI Setup displays a list of installed audio drivers. In the lower left corner click on the “+” symbol to add a new aggregate device.



In your Aggregate Device settings, first select your main hardware device drivers. This sets their priority as actual audio inputs and outputs. Then Select Soundflower as the last item.

When recording at sample rates other than 44.1kHz, set these in the Aggregate Device, then close Audio MIDI Setup



# mØxr β

## Propellerhead Reason+Record Duo

**mØxr** will function with Reason 4, Reason 5, or Reason+Record Duo. For full recording functionality from hardware MIDI sources, the choice should be obvious.

**mØxr** does not act as a Rewire Host application, making it possible to track audio signals from hardware MIDI synthesizers being triggered by Record MIDI sequences.



**Reason+Record**

# mØxr β

## Record Audio Preferences

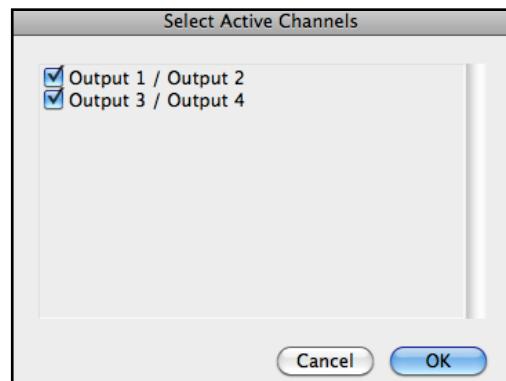
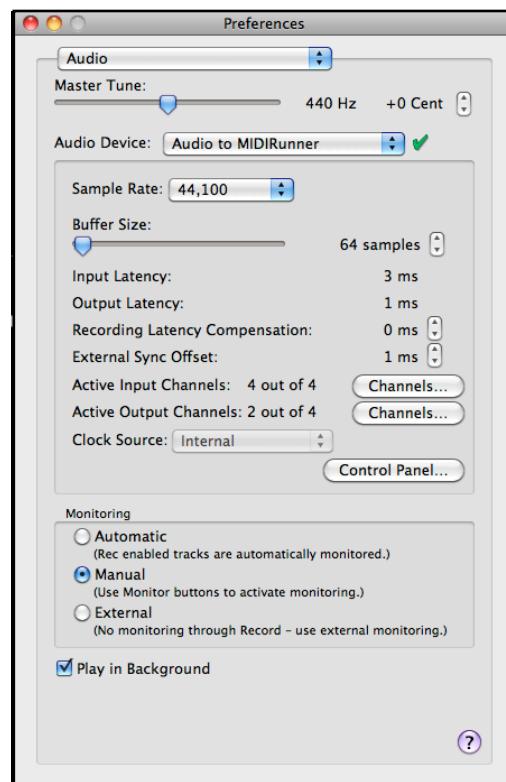
Select the Aggregate Device Driver in the Reason / Record Audio preferences.

To reduce latency, set the buffer size to minimum setting or increase the Sample Rate. If changes are made to the sample rate, you should first close the application and make the changes in Audio MIDI Setup, and then relaunch Reason/Record.

In the early tests, most problems seem to occur at this stage, sometimes requiring full system reboots.

Click on the “Channels...” button next to Active Output Channels.

Make sure the last pair of audio outputs are enabled. Because it was at the bottom of the aggregate device list, the last output pair listed corresponds to soundflower. For the example Record Session file included, only four outputs should be enabled. The first pair is the main audio outputs, and the last pair is directed to mØxr.



# mØxr β

## mØxr

**mØxr** relies on two audio signals from Reason. One carries Note and the other with velocity messages generated in a Thor patch.

The **mØxr+** application receives four audio signals. Two for the note data, one for mod wheel data, and one for pitch bend information.

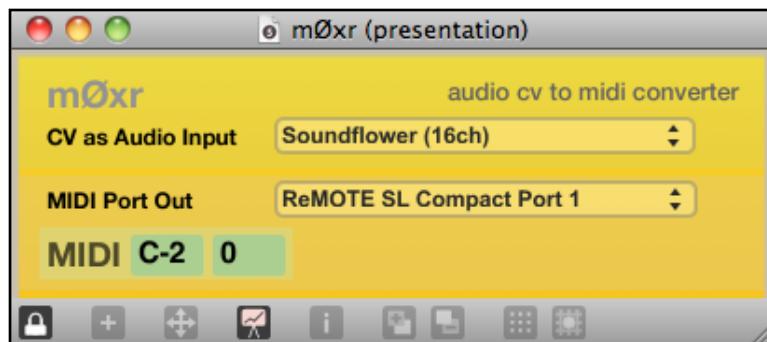


[mØxr.app](https://moxr.app)

# mØxr β

## mØxr Settings

The **mØxr** patch opens in a small window with the main settings visible for CV as Audio Input, and MIDI out.



For the **CV as Audio Input** Setting, click on the pull down menu and select Soundflower. This action should be performed even if soundflower is already selected. This insures that the driver is initialized

Select the MIDI Output Device. Next to **MIDI Port Out**, Click on the pull down menu and select a MIDI destination, even if one is already selected, this should be applied to reinitialize the setting.

# mØxr β

## Optional Settings

**mØxr** includes optional settings to select input sources if you are using the 16channel version of Soundflower. **mØxr+** includes modwheel and pitchbend scaling routings - this requires at least four audio channels streams from Record. To access these settings, expand the window size.

Four additional selection menus are included for optimizing performance of the control voltage conversion process. Overdrive, Scheduler in Audio Interrupt Mode, I/O Vector Size, and Signal Vector Size.

Overdrive and Sch. Audio Int. should be ON. The vector size settings may vary depending on the resources available of your system. 16 to 64 are the optimal values.



# mØxr β

## MIDI to CV

Open the file called “MIDI to CV.record” included in the file archive. This file can be used as a template to operate with mØxr.

The session contains a Thor patch that converts Voice Key Velocity and Voice Key Note value to Audio signals. The output signals are routed to outputs 3 and 4. This is programmed in this manner to prevent any accidental audio passing into a mix channel. The Thor settings are specific and changing the polyphony or mode parameters will change the MIDI output results.

The outputs from Thor are connected to the Hardware Interface Outputs 3 & 4 and the Master Section is connected to Outputs 1 & 2. The main stereo mix should be directed to your monitoring setup.

**Outputs 3/4 and 5/6 should be routed to Soundflower 16ch connected to mØxr+.**



MIDI to CV.record

# mØxr β

## Troubleshooting MIDI OUT

There might be a bit of troubleshooting at hand when you first launch this configuration. To first test MIDI out, select either the Thor Device or RPG-8 arpeggiator in the device list and press a few notes on your keyboard controller.

Observe the Møxr window and check to see if there's activity on the MIDI note and velocity fields. If you're lucky, then this should work straight away.

Typically the problems seem to lie with dropped audio driver settings. Monitor Record Hardware Interface VU meters and look to see signals appearing on output channels 3 and 4. If there is no signal, then you may need to reset the audio driver output channels.

On the mØxr settings, do not select your Aggregate Device as the driver. This will reset the audio preferences in Record. Make sure to only use Soundflower (2ch). If you change the audio driver in mØxr, you may need to quit and relaunch Record.

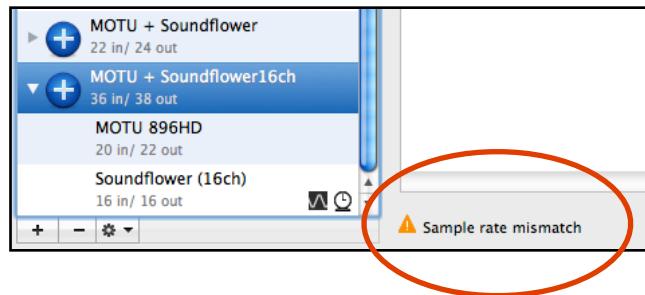
If errors persist with the Audio MIDI Setup utility and crash reports occur, you should reboot your system.

# mØxr β

## Troubleshooting Audio MIDI Setup

The Reason Audio Driver may cause issues at various buffer settings and changing the sample rate settings will most likely cause problems with the Aggregate Driver settings. This will result in audio drop outs and interrupted audio as CV between Reason and mØxr.

1. Quit Everything
2. Open your hardware interface control panel and set the sample rate.
3. Open Audio MIDI Setup and edit the aggregate device. Expand the aggregate driver to see the included devices.

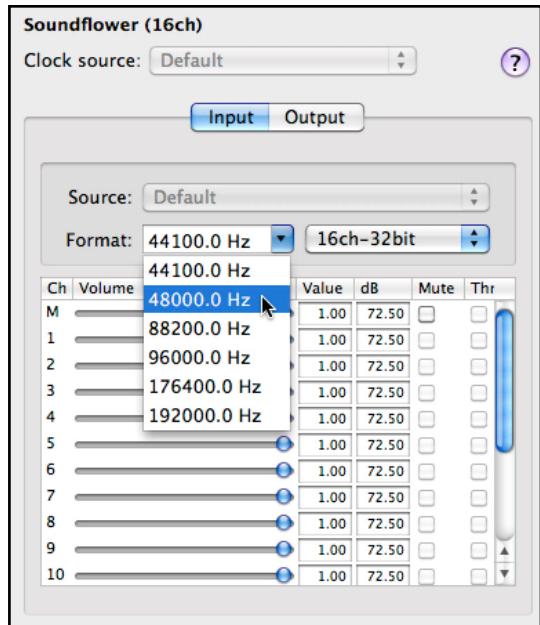
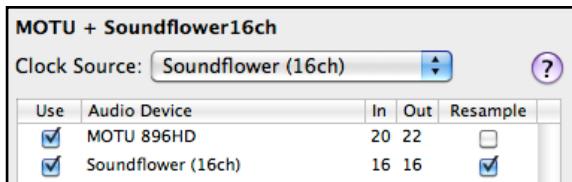


Audio MIDI Setup will display an error message when there's a sample rate mismatch.

# mØxr β

## Troubleshooting Audio MIDI Setup

4. Go into each of the aggregated device settings and set the input and output sample rates to match the hardware settings. Repeat this for each hardware device and soundflower.
5. Select the aggregate driver item, and make sure that Soundflower is set to resample.



6. Launch Record and mØxr.
7. In the mØxr edit menu, select Overdrive.
8. Open the Audio Preferences and adjust the buffer settings.

Record Audio Driver buffer settings vary depending on your hardware interface. The lowest setting of 64 seems to work well in most situations, however this will limit the amount of processing achievable for normal operations.

# mØxr β

## MIDI to CV Thor Patch (mØxr+)

The MIDI to CV Thor patch is the gateway to mØxr. Monophonic sequences of MIDI Notes, CV/Gate events, and Thor built in Sequencer patterns are converted to audio signals recognized by mØxr.

Key Velocity appears on Thor Audio Output 4

Key Note Value appears on Thor Audio Output 3

Mod Wheel values appear on Thor Audio Output 1

Pitch Bend data appears on Thor Audio Output 2

Monophonic MIDI Events can be directed from a Keyboard Controller or MIDI Sequence.

Matrix Pattern Sequencer or RPG-8 Arpeggiator should be connected to the Note and Gate Inputs of the Thor in their standard manner. However MIDI sequences of legato or adjacent notes are recognized as a continuous gate message until a gap between note events is received in mØxr.



# mØxr β

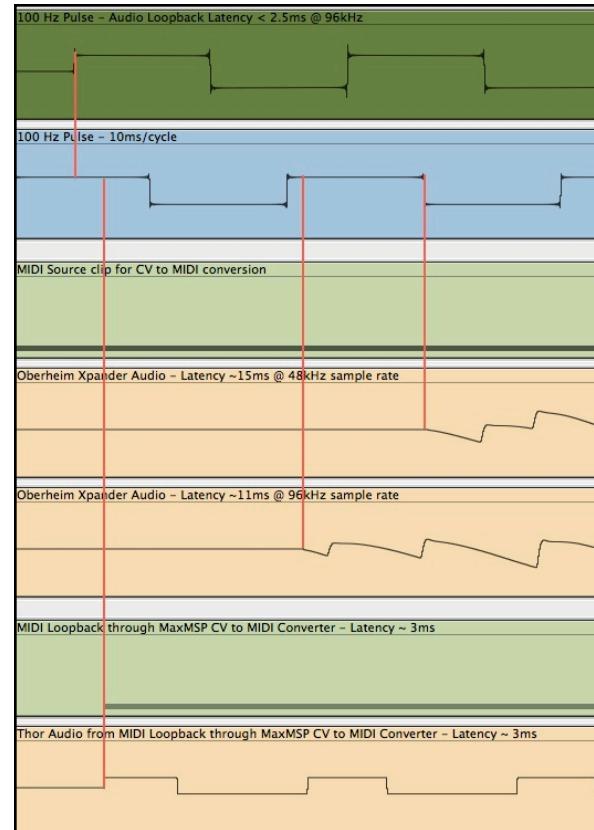
## Latency

Tests of **mØxr** routing reveal that latency is typically around 11ms to 19ms. This is calculated from the instance of a MIDI Note On event in a sequencer track relative to the rising envelope of a recorded audio signal from an external hardware synthesizer. When recording audio from hardware synthesizers, this should be manually adjusted in the clip editor.

MIDI loopback from MIDI Runner to a Record MIDI track is consistently around 3ms.

Audio Loopback latency varies depending on the buffer setting. The lowest latency achieved during testing is less than 2.5ms. This was achieved with a buffer setting of 64 running at 96kHz sample rate. **mØxr** total latency at this setting was about 11ms.

The unaccounted latency for MIDI interface and hardware latency is about 7ms.



# mØxr β

## mØxr+ development

mØxr+ will include mod wheel and pitch bend routings from Thor to audio channels. The additional functionality requires more processing resources in both Record and MaxMSP, however performance is not decreased in any significant manner.



MIDI to CV for mØxr+.record



mØxr+.app



# mØxr β

## ADAT Lightpipe Loopback

As an alternative to using soundflower, it may also be possible to create a hardware digital audio loopback configuration to route CV as audio signals from Record to mØxr. This approach requires a hardware interface with ADAT Lightpipe optical ports and a TOSLINK fiber connected between the ADAT Output and ADAT Input.

Both Record and mØxr would rely on data from a single hardware device driver, so aggregating drivers is not required. Simply set both mØxr and Record to the same hardware interface. Enable ADAT Output channels in Record, and enable ADAT Input channels in mØxr.

In Record, the CV as audio signals are directed to the ADAT Outputs, and on mØxr, the CV as audio signals are received from the ADAT Inputs.



# mØxr β

## License & stuff

**Reason & Record** are the property of Propellerhead Software. Stockholm, Sweden

**MaxMSP and Soundflower** - Cycling 74. San Francisco, CA.

**mØxr** © 2011 Kurt Kurasaki  
<http://peff.com>

Thanks to UC Berkeley Center for New Music and Audio Technologies (CNMAT)

This project was codenamed *MIDIRunner* in honor of Propellerhead User Forum members who have requested a MIDI out feature for Propellerhead Reason<sup>1</sup>.