

Jamstix on Macintosh

Rev 1.2
5/26/2009

Jamstix is currently not available on the Macintosh platform, however its possible to run Jamstix on a Macintosh and even to synchronize with some popular DAW's such as Digital Performer, Logic, etc.

This guide will describe how to setup and use Jamstix together with Digital Performer and several other software technologies that will make it possible:

- Digital Performer
- JackOSX
- CrossoverGames 7.0.0
- JackWASIO (wineasio)
- Reaper for Windows

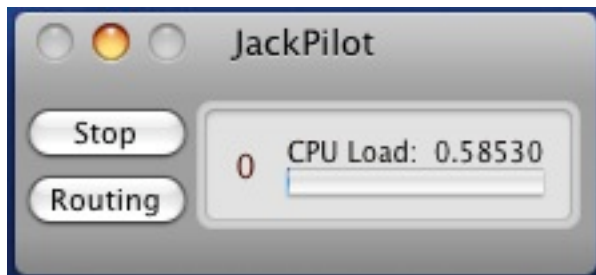
Installation

Installing JackOSX

Download and install the latest version of jackosx. You may have to reboot:

<http://www.jackosx.com/>

Start up JackPilot application. You will see the following window on screen for JackPilot:



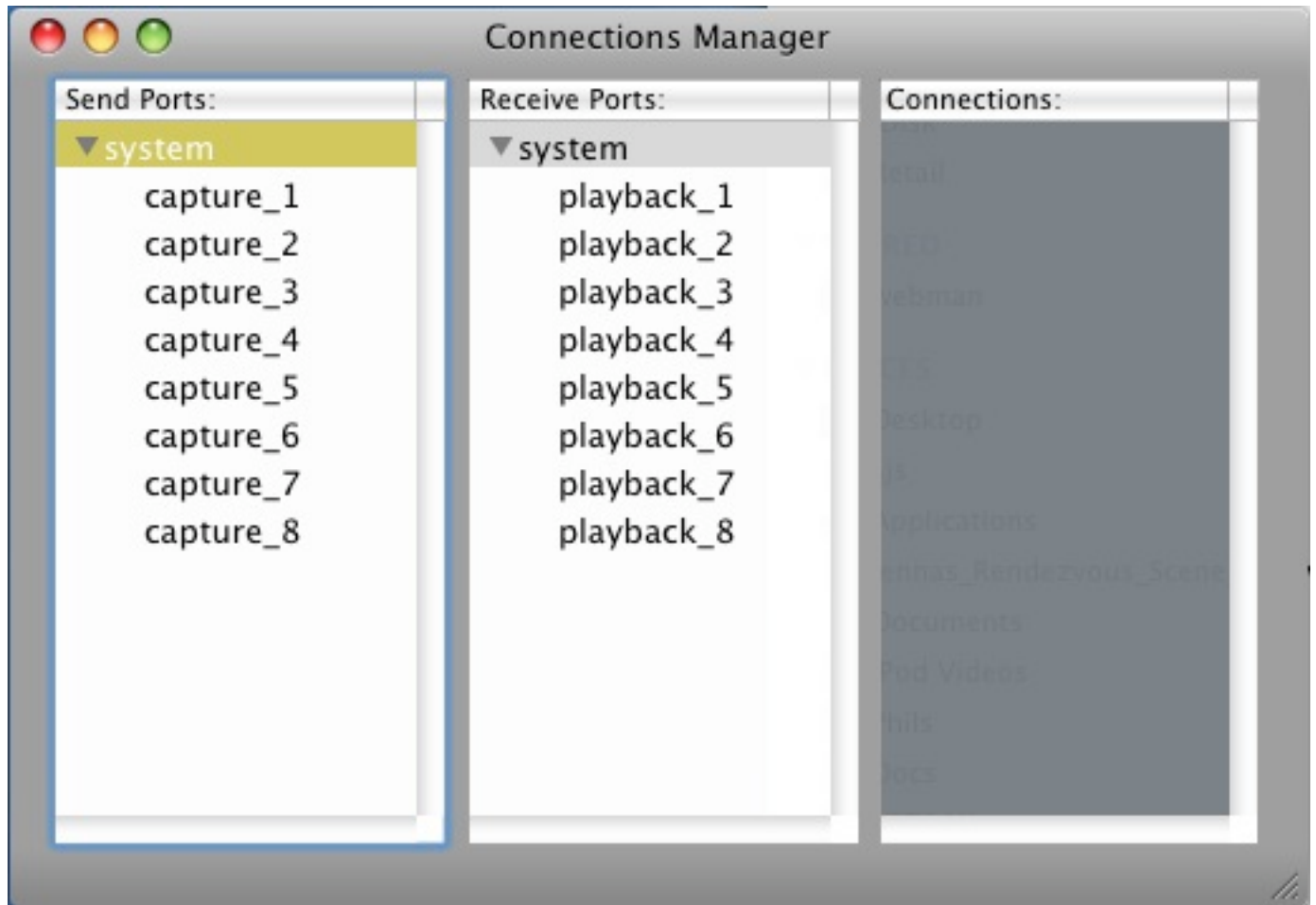
Go to the Preferences and configure JackPilot for your audio device. You can configure as many virtual audio ports as you desire for JackOSX. For this test you won't need more than 4. Make sure to set the buffer to a safe value for now like 256 or 512. Then click "Save". Later on you should consider lowering this value to something like 64 or 128, depending on your system. (See end of document for more comments on this)



The image shows a 'Preferences' dialog box for JackPilot. It is divided into two main sections: 'Jack Server' and 'JackRouter'. The 'Jack Server' section contains six settings, each with a dropdown menu: 'Driver' (coreaudio), 'Interface' (MOTU Traveler), 'Sample Rate' (44100), 'Buffer Size' (256), 'Interface Output Channels' (8), and 'Interface Input Channels' (8). The 'JackRouter' section contains four settings: 'Virtual Input Channels' (4), 'Virtual Output Channels' (4), 'Auto-Connect with physical ports' (checked), and 'Verbose logging for debug purposes' (checked). At the bottom of the dialog are 'Cancel' and 'Save' buttons.

Preferences	
Jack Server:	
Driver:	coreaudio
Interface:	MOTU Traveler
Sample Rate:	44100
Buffer Size:	256
Interface Output Channels:	8
Interface Input Channels:	8
JackRouter:	
Virtual Input Channels:	4
Virtual Output Channels:	4
Auto-Connect with physical ports:	<input checked="" type="checkbox"/>
Verbose logging for debug purposes:	<input checked="" type="checkbox"/>
<div>Cancel Save</div>	

After configuring preferences, click on the start button of JackPilot to startup the JackOSX engine. At this time you can click on the Routing button to see the current audio inputs and outputs of your hardware audio device. They are currently not patched to anything, but this is what it will look like for a hardware audio device with 8 ins and 8 outs:

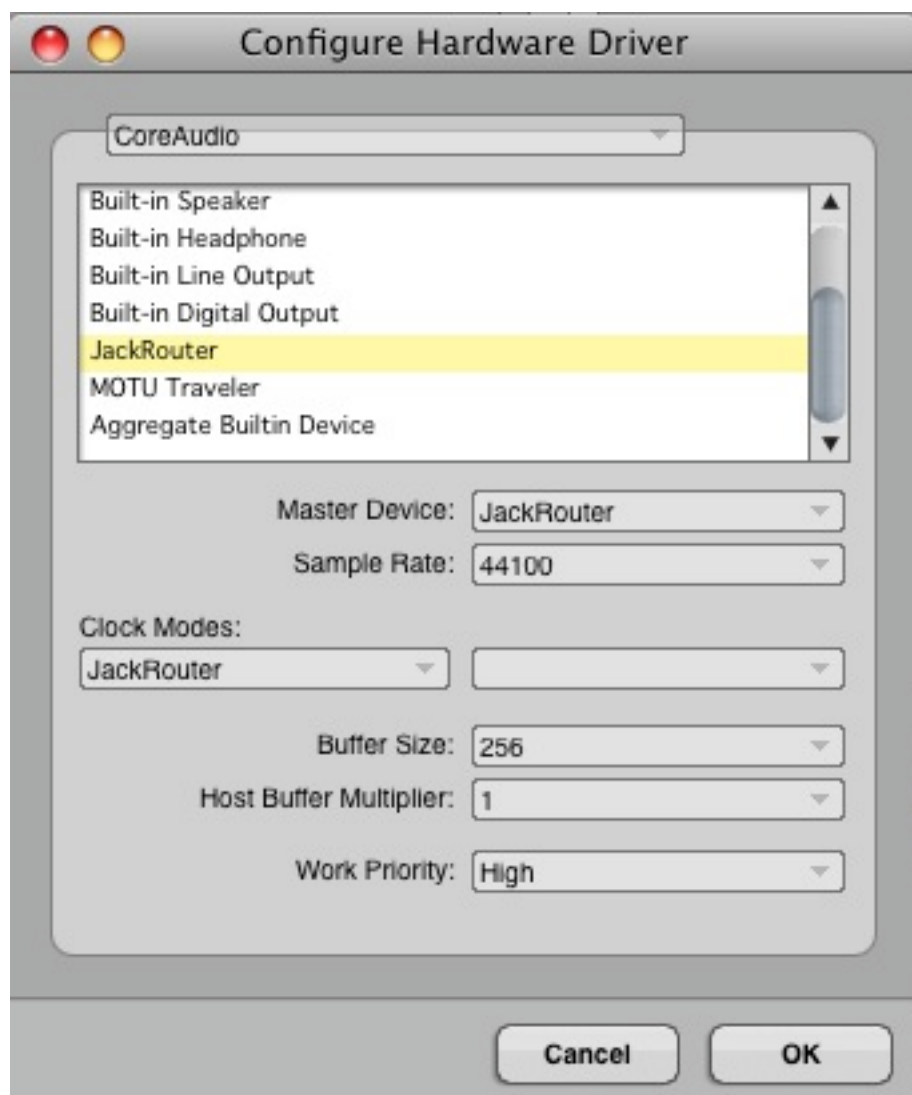


NOTE – You must always make sure to start JackPilot before launching any other audio applications you intend to use with it and you must Quit JackPilot last after closing the other audio applications.

Launch Digital Performer and configure it to use JackOSX by going to the following menu:

Setup->Configure Audio System->Configure Hardware Driver

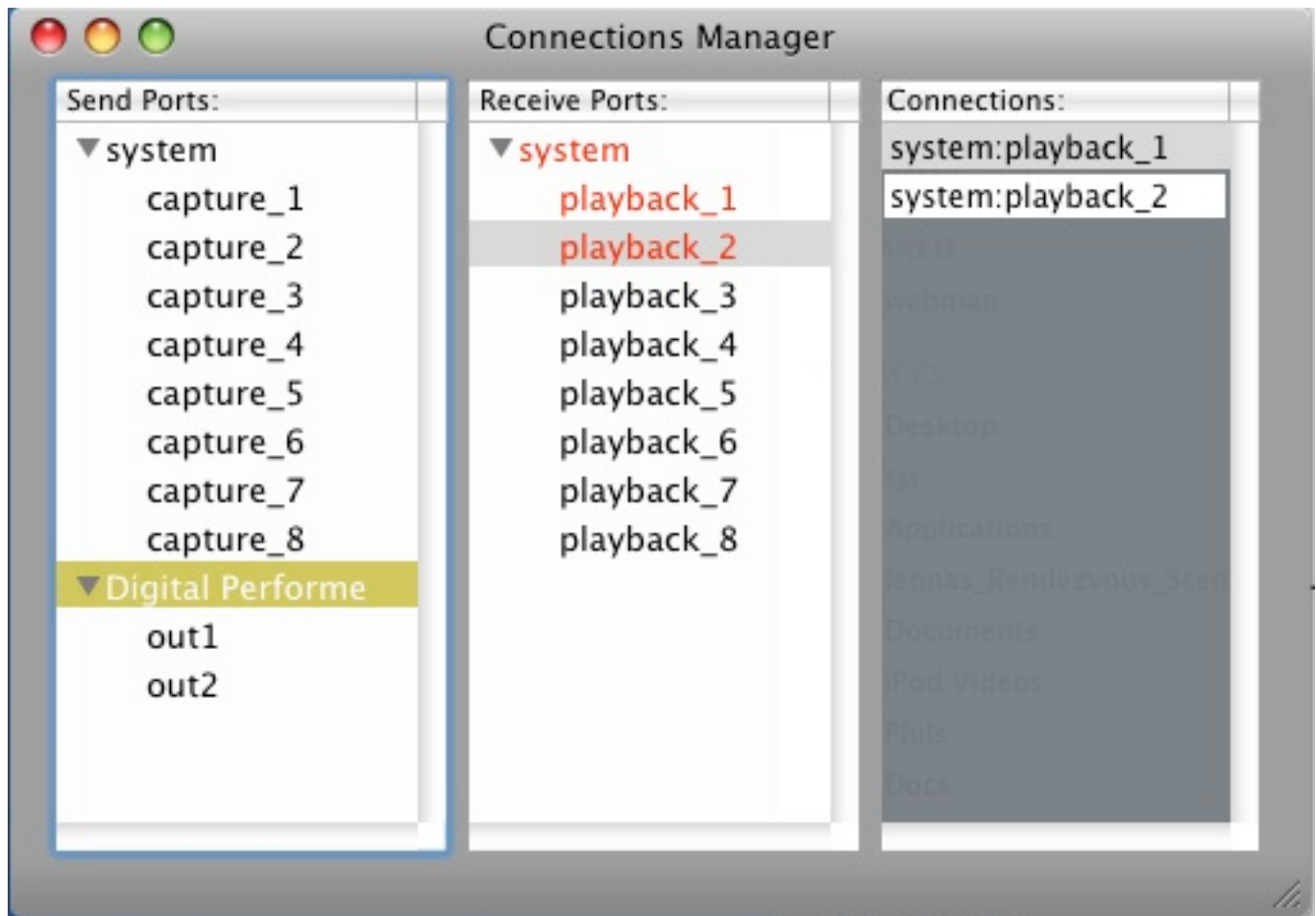
Select JackRouter as the CoreAudio device to be used by DP. You will not be able to change the buffer setting here when JackOSX is used, You set that earlier in JackPilot.



You should test out JackRouter now by making sure that when you have JackRouter configured as the audio device in DP, you actually hear audio. Load up a soft instrument or something you can use to generate sound. In DP you will see audio ports labeled **JackRouter 1-2**, which you should send your instrument to instead of an actual hardware port. Here is an example using Superior Drums, the audio is being sent to JackRouter 1-2.



Once you have enabled a track for playing in DP with JackRouter 1-2 as the audio out ports to use, you will see these show up in the JackPilot Router:



Try to play the instrument. If you do not hear sound, then it means you need to use the JackPilot router to connect the outputs from Digital Performer to your audio device. This is explained in more detail in the JackOSX manual.

But briefly, select “out1” under the “Digital Performer” tree item in the router. While its selected look in the Receive Ports column and look at system->playback_1 to see if its red. If it’s red it is wired up and you should be hearing audio. If not, then double click on it. That will create a connection from Digital Performer->out1 to system->playback_1 and you should start to hear audio. Do the same thing for out2.

Do not proceed unless you have this working and understand it. You should be able to play the instrument in DP and hear audio, using JackRouter as the audio interface in DP.

Installing CrossoverGames 7.0.0

Download and install the CrossoverGames product. This is a commercial product. This product can be found at:

<http://www.codeweavers.com>.

NOTE - that I have found some versions of this product work with jamstix/wineasio and others do not. As of this writing, CrossoverGames 7.0.0 for sure works, but newer updates have some problems with Jamstix. Its possible those problems can be resolved. For now I can verify that CrossoverGames 7.0.0 works. I had problems with CrossoverMac, even though it worked in 6.x versions. The status of this will undoubtedly change in the future. Just be advised that you may have to experiment until you find the version of Crossover(Mac or Games) that will work completely with Jamstix and several other key technologies. Codeweavers maintains all versions of the product available for download so try them until you find the right combo or just use the version I am telling you to use here.

There is a 30-day demo if you want to try this out first.

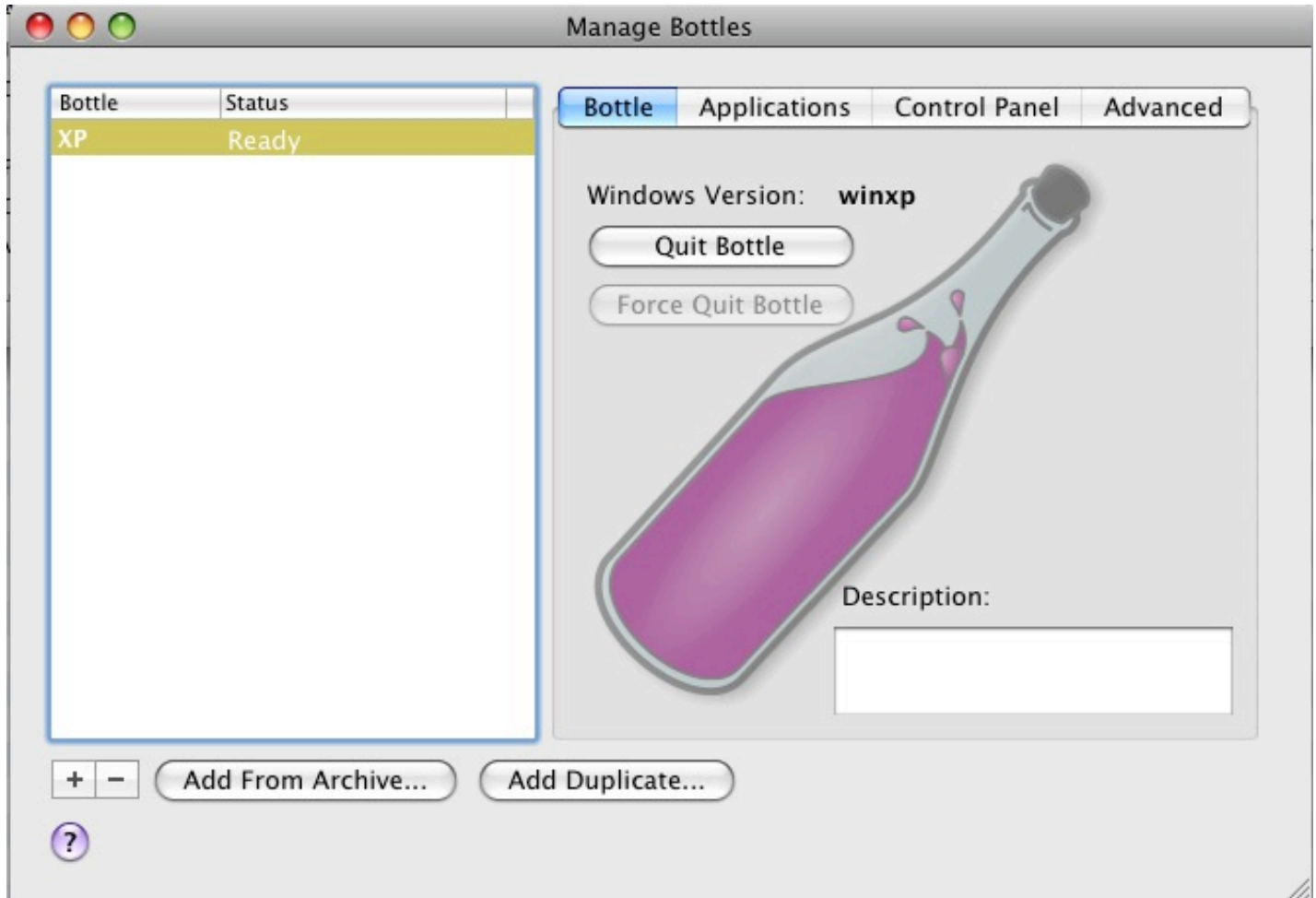
This is also based on the opensource Wine product, sometimes called **Darwine**. You may be able to get the freely available Darwine to work, but I had problems getting it to work and I feel that the Crossover commercial product is not very expensive and will save many hours of frustrating configuration with Darwine.

Installing CXGames or CXMac is very simple, just copy it to your Applications folder.

Launch CXGames the first time and configure a new bottle with the following menu item:

Configure->Manage Bottles

Click on the **+** sign to create a new bottle. Call it anything you want. I called mine "XP":



Once you have a bottle created, you can install Windows programs into it just as if it was a working windows machine. ALWAYS install new programs into CXGames using the CXGames Installer dialog. Do not ever directly launch a windows setup.exe type of program. CXGames keeps track of certain things so you need to launch the installers using this wizard.

The following pages will show how to install win32 Reaper onto your mac under CXGames. Read up a bit about how CXGames configures a virtual drive_c for your virtual environment, where its located and where it will place program icons for windows programs you will install. These are normally located in the following two locations and you may find yourself looking for them later:

```
~/Library/Application Support/Crossover Games/Bottles/<bottlename>/drive_c/
```

```
~/Applications/Crossover Games/
```


Installing Reaper

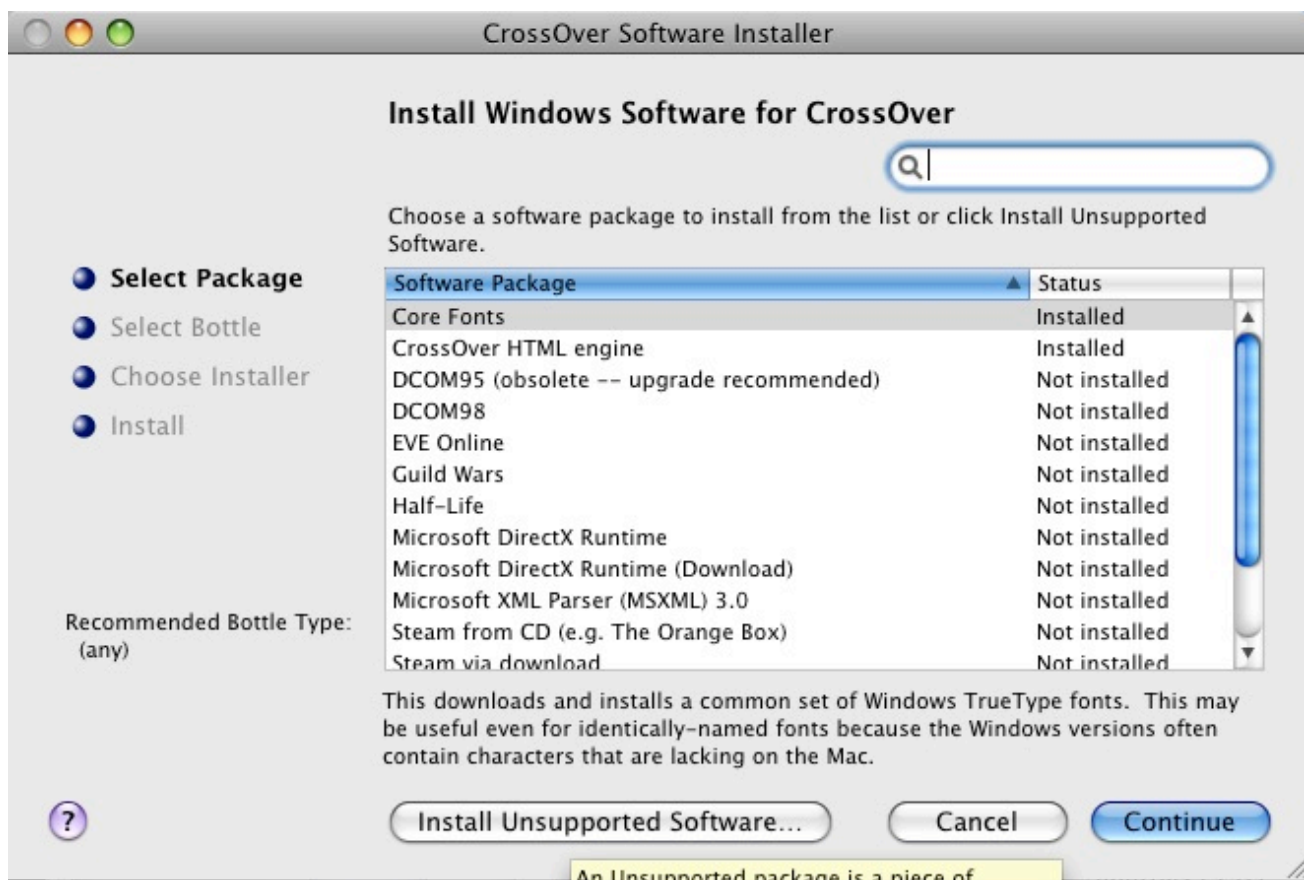
Note - Reaper3 is now available and appears to work quite well with virtual midi, as we are going to soon demonstrate. Some of the last versions of 2.5.x were broken in terms of external midi, FYI. I found that 2.5.5 worked, but later versions did not. However, Reaper 3.0 appears to work very well.

Download the Reaper installer BUT DO NOT INSTALL from:

<http://www.reaper.fm/>

In CXGames, go invoke the following menu item:

Configure->Install Software



Click the **"Install Unsupported Software"** button.

Click the **"Continue"** button on the next useless screen.

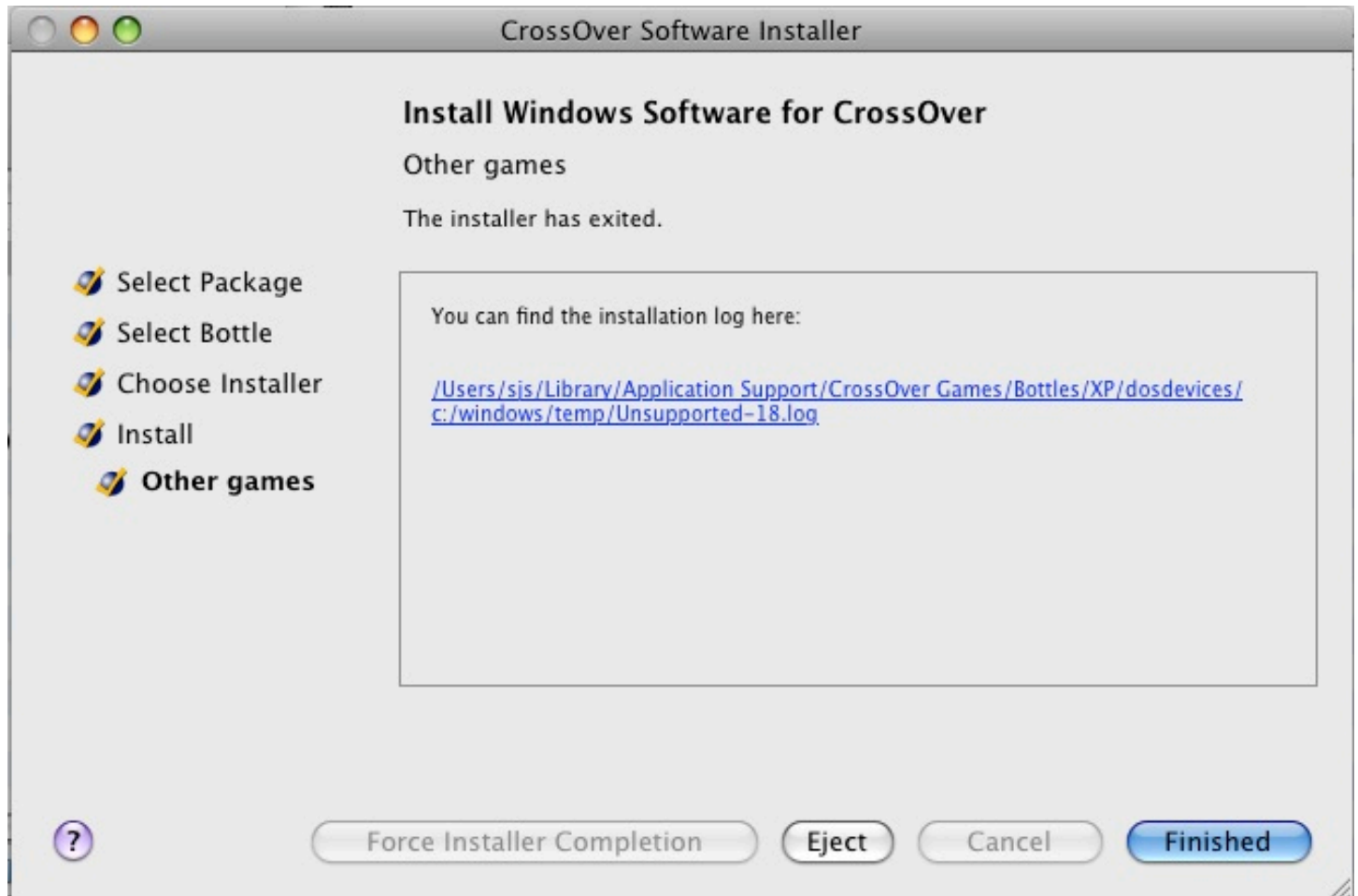
Make sure to Install into the bottle you created earlier and hit “Continue”:



Choose the icon for “Choose Installer File” and then click the “Install” button:



You will be prompted for an installer exe. Look for the Reaper setup exe that you downloaded earlier and open it. The Reaper installer will begin. Install Reaper but DO NOT elect to run Reaper at the end when asked. There is a final CXGames dialog that will come up telling you that the installation is complete and you should click the “Finished” button before proceeding:



Now to run Reaper, from the CXGames menu, select:

Programs->REAPER->REAPER

This will launch the win32 version of Reaper. Make sure that it is able to run ok, but do not try to do anything with it yet. Exit the application. Do not proceed until this is working properly.

NOTE – You may also notice that some program icons have been created for launching Reaper. These are found in ~/Library/Applications/Crossover Games/ and will launch things like Reaper directly. You can drag this icon to your dock bar and will be able to launch win32 reaper from the dock after that.

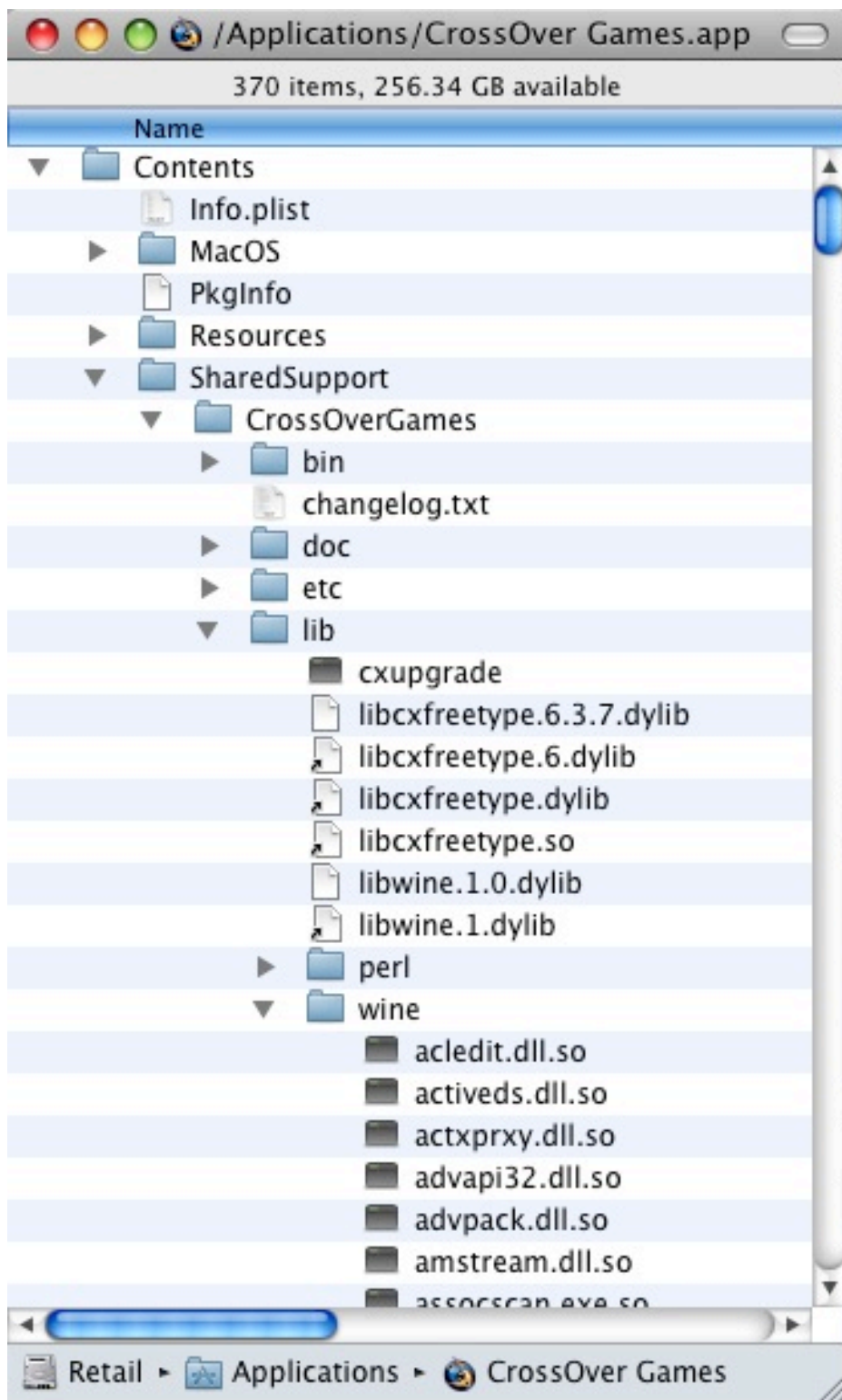
Installing JackWASIO

In order to use low latency audio from CXGames on your mac you will need to use a special little driver, which has been designed specifically to work with CXGames and JackOSX. It will provide ASIO access within the CXGames universe; directly to JackOSX virtual ports.

Currently this can be downloaded from the following source:

<http://www.meskalina.it/blog/index.php?entry=entry071002-222040>

There is a readme that explains how to install it, but to summarize: quit out of CXGames and then navigate to the Applications folder, right click on the CrossoverGames icon and select “**Show Package Contents**”. Navigate until you see the following “wine” folder:



Copy the file from the JackWASIO download called **"wineasio.dll.so"** into that "wine" folder inside the CrossoverGames package, then close the pkg windows.

Start up CXGames again and go to the following menu:

Programs->Run Command...

Type in a command of **“regsvr32 wineasio.dll”** and hit the “run” button:



When it is complete, close the Run command window.

Low latency ASIO audio from Reaper through JackRouter should now be ready.

Create VstPlugins Folder

Create a folder that you will use for VSTPlugins. I use the following:

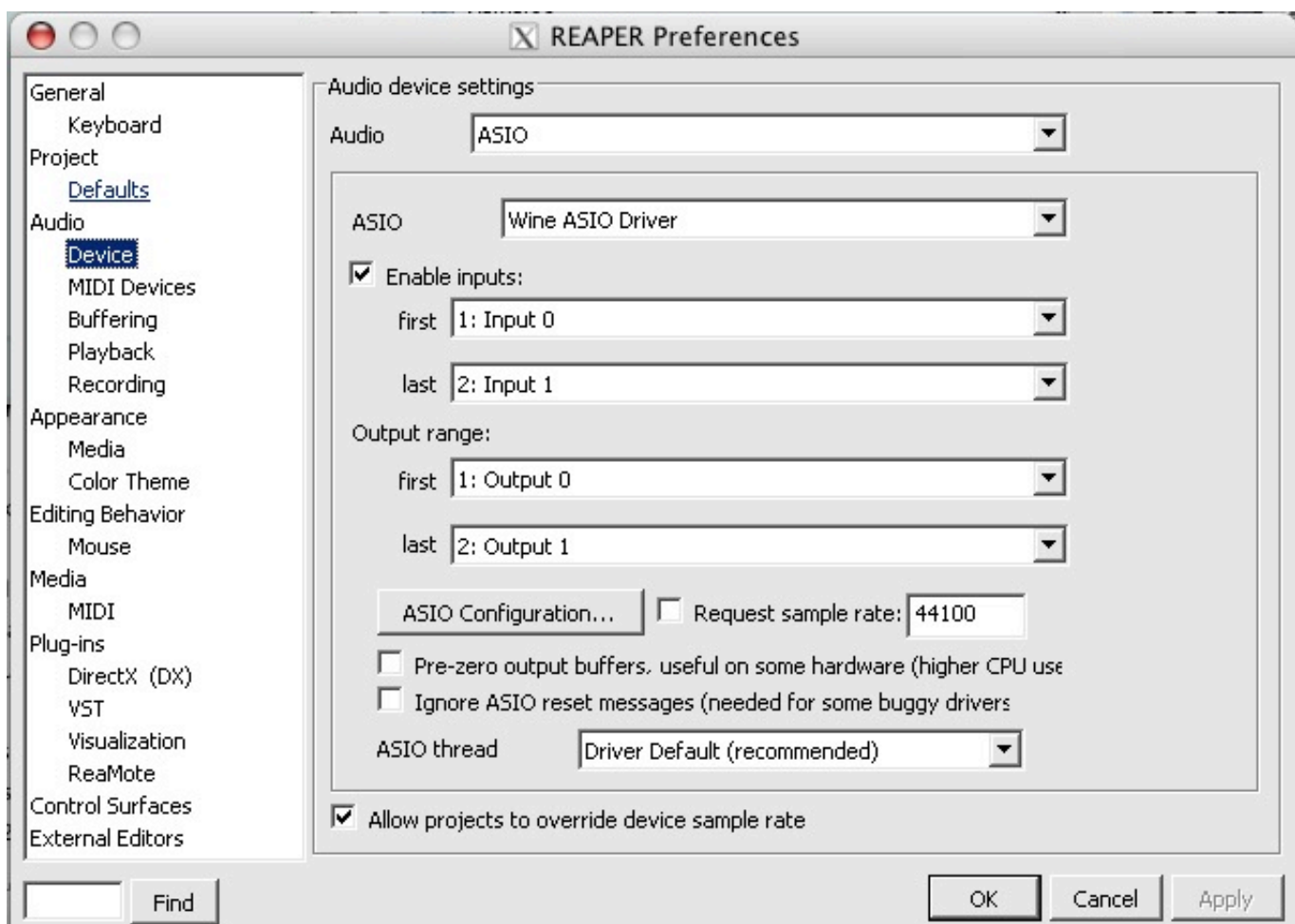
```
~/Library/Application Support/Crossover Games/Bottles/<bottlename>/drive_c/Program  
Files/VstPlugins
```

Test Reaper

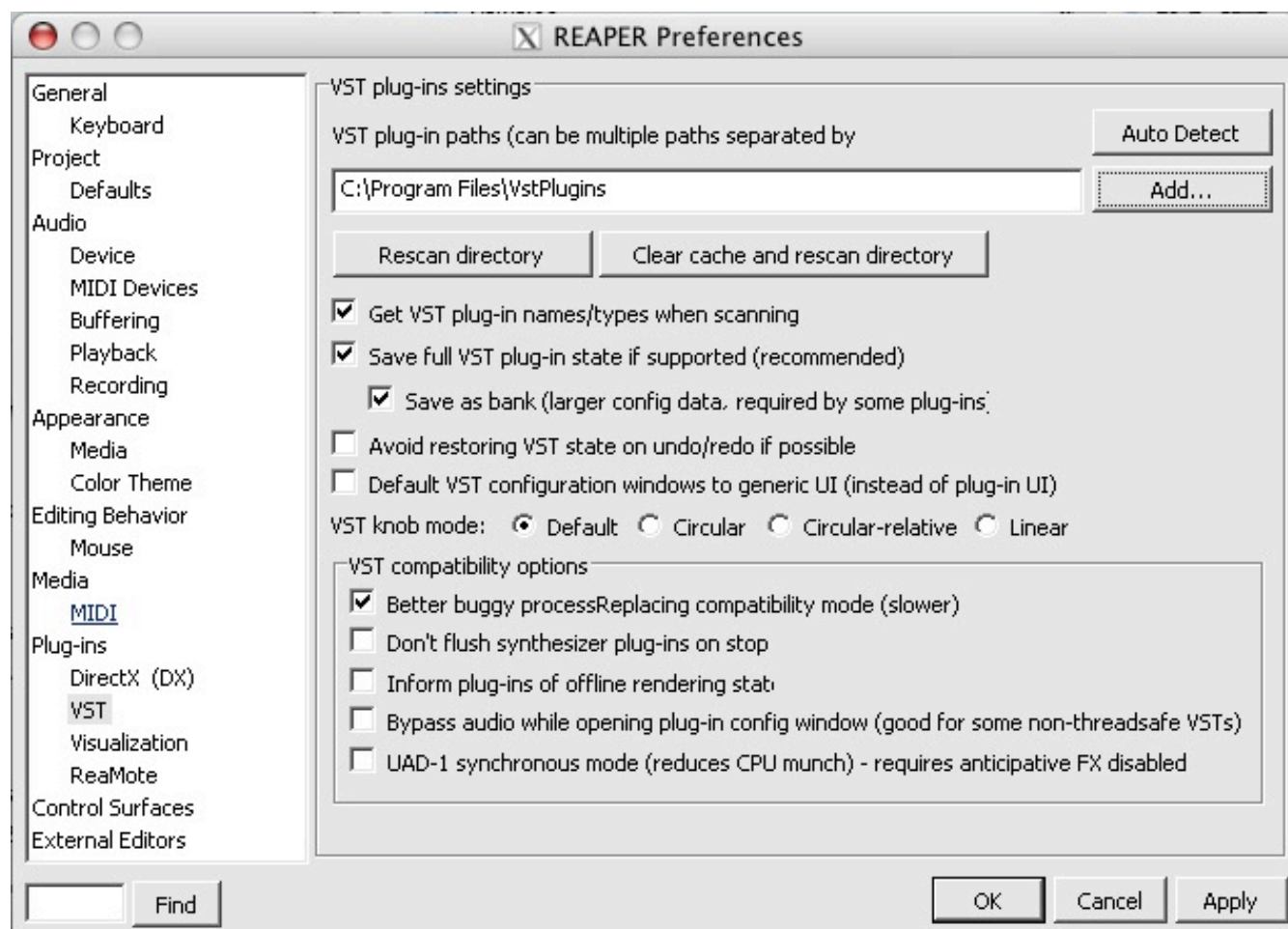
Reaper is now mostly ready. We just need a plugin to use within it, like Jamstix. If you have another simple VSTi you can use to test out Reaper, now is not a bad time to do that before proceeding to install Jamstix.

Make sure JackPilot is already running.

Launch Reaper. Go to the **Options->Preferences** dialog to configure several things. First configure the audio device to use ASIO:



Then configure the VST path, select the path you created earlier and make sure to click on the “**rescan**” button to rescan the vst plugins you have there already.

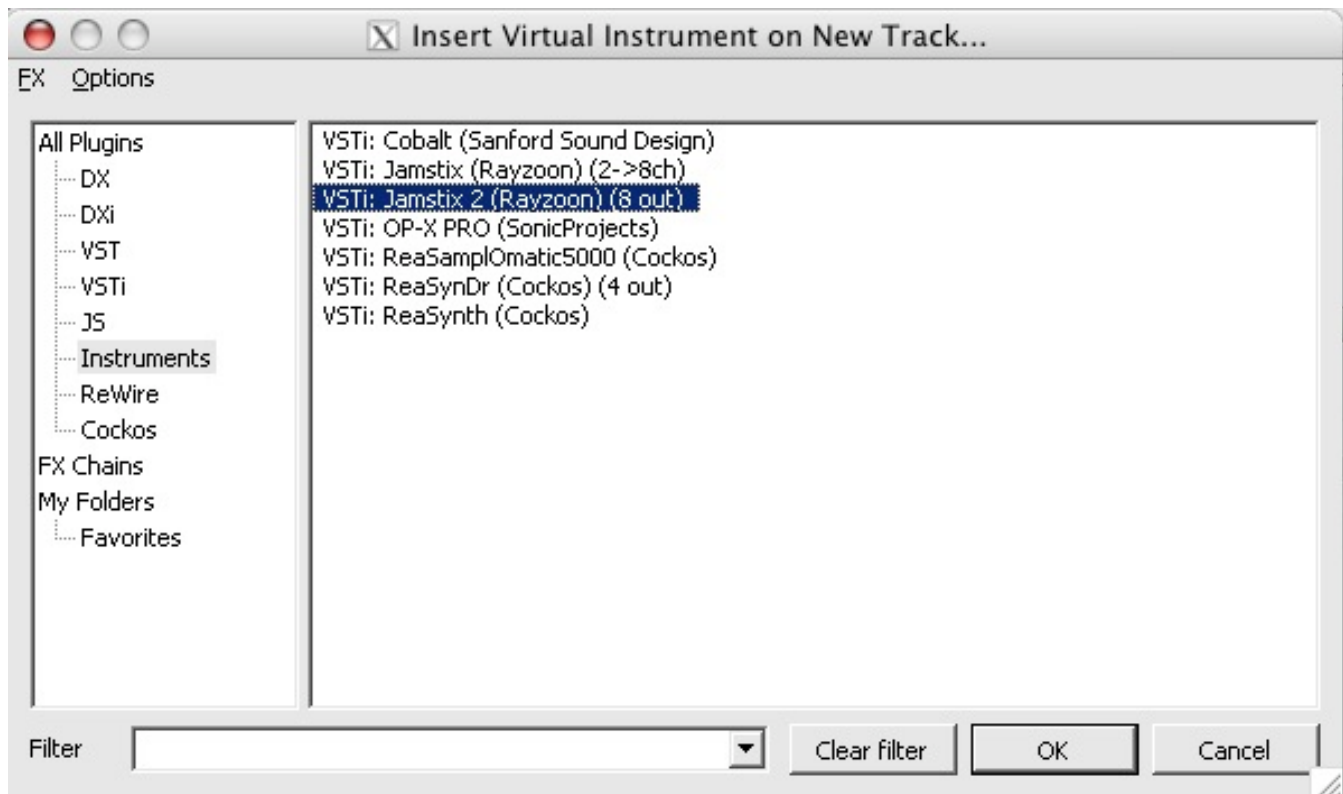


Close the preferences dialog.

From the Reaper menu select:

Insert->Virtual Instrument on new track

And select the VST of your choice to use as a test.



Go to the JackPilot application and look at the Router to see that Reaper has appeared on the routing table with x number of virtual ins/outs depending on the VST you used. If not done already, make sure the first two reaper outs are connected to the system->playback_1 and 2 outs like we did earlier for Digital Performer. That will route the output of Reaper to your audio card.

When the VST gui comes up, attempt to make some sound on the instrument. You should hear audio. If you don't, then figure out why not before proceeding. Don't forget to check the JackPilot router to make sure the outputs from Reaper are connected to the system outputs(ie, soundcard)

Installing Jamstix

Installing Jamstix is pretty straightforward. Similar to how we installed Reaper earlier, use the CXGames Install Software dialog to look for the Jamstix installer and execute it. For Jamstix1, make sure to run the data installer this way too as well as any program updaters.

Note that there is a small bug in Jamstix2 that is overcome from version **2.1.0** onwards. You will not be able to install any samplepaks into jamstix2 on CXGames until you first update to 2.1.0 or later. After updating to 2.1.0, you must edit the **jamstix.ini** file with a special value that will work around the bug.

Add the following entry to the [General] section of the jamstix.ini

```
BlockMouseMove=1
```

I recommend that you first install the updates up to 2.1.0 and then update the jamstix.ini. Then you will be able to launch Jamstix in Reaper in order to install samplepaks. After all the samplepaks are installed, go back and re-run all the updaters.

Finally time to test out Jamstix. First make sure JackPilot is running. Run Reaper. Insert a VST like we did earlier, but with jamstix this time. The Jamstix gui will pop up. You know what to do next. You should be able to hear drum sounds. If not, double check the JackPilot routing as before until you do. Do not proceed until this is all working so far and you understand it.

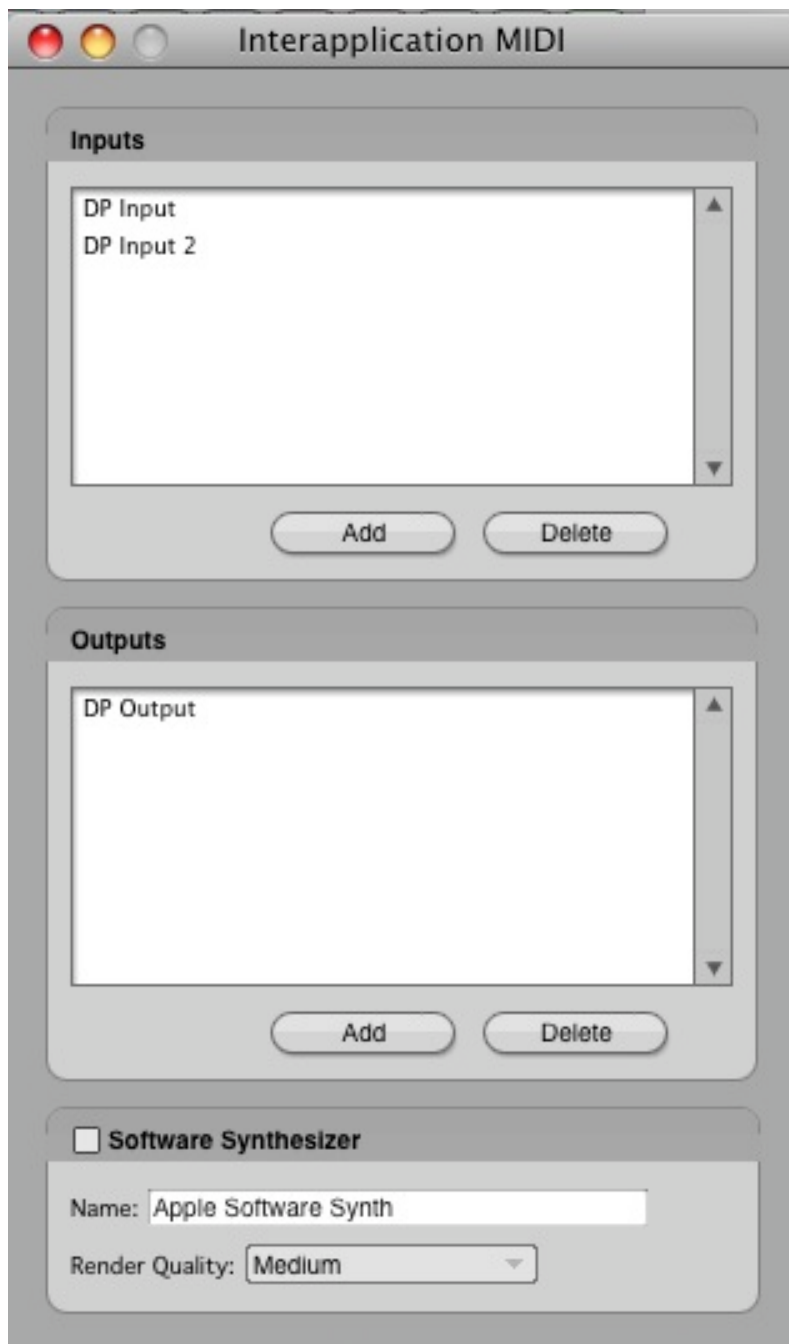
Jamstix/Reaper midi to DP

Now comes the fun part, tying it all together with Digital Performer. First we will deal with only sending midi from Jamstix over to Digital Performer.

First make sure JackPilot is already running.

Next you will need to create some virtual midi ports to connect these two applications. Digital Performer has the ability to create virtual midi ports. You access this screen from the following menu:

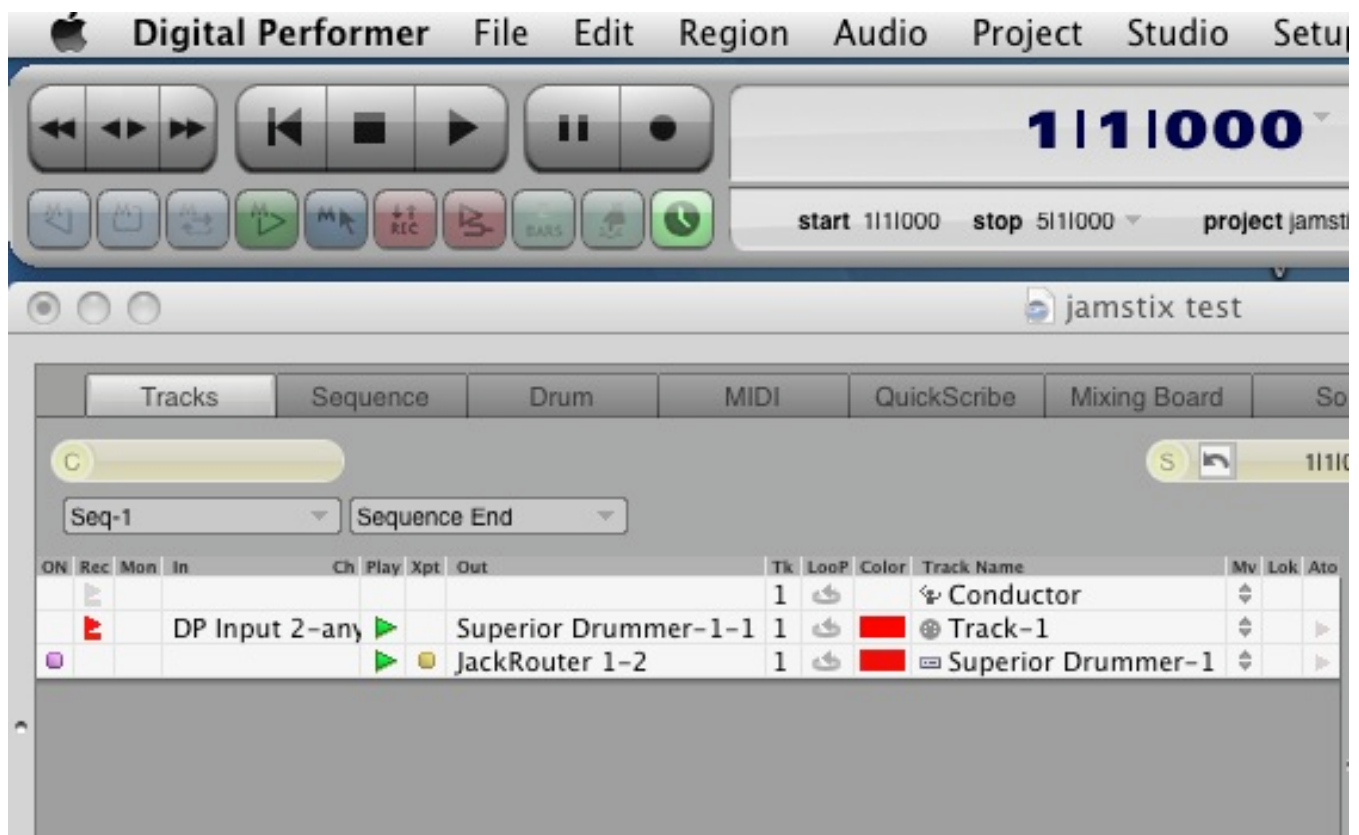
Setup->Interapplication Midi...



Add a couple of input midi ports. No output port is necessary at this time.

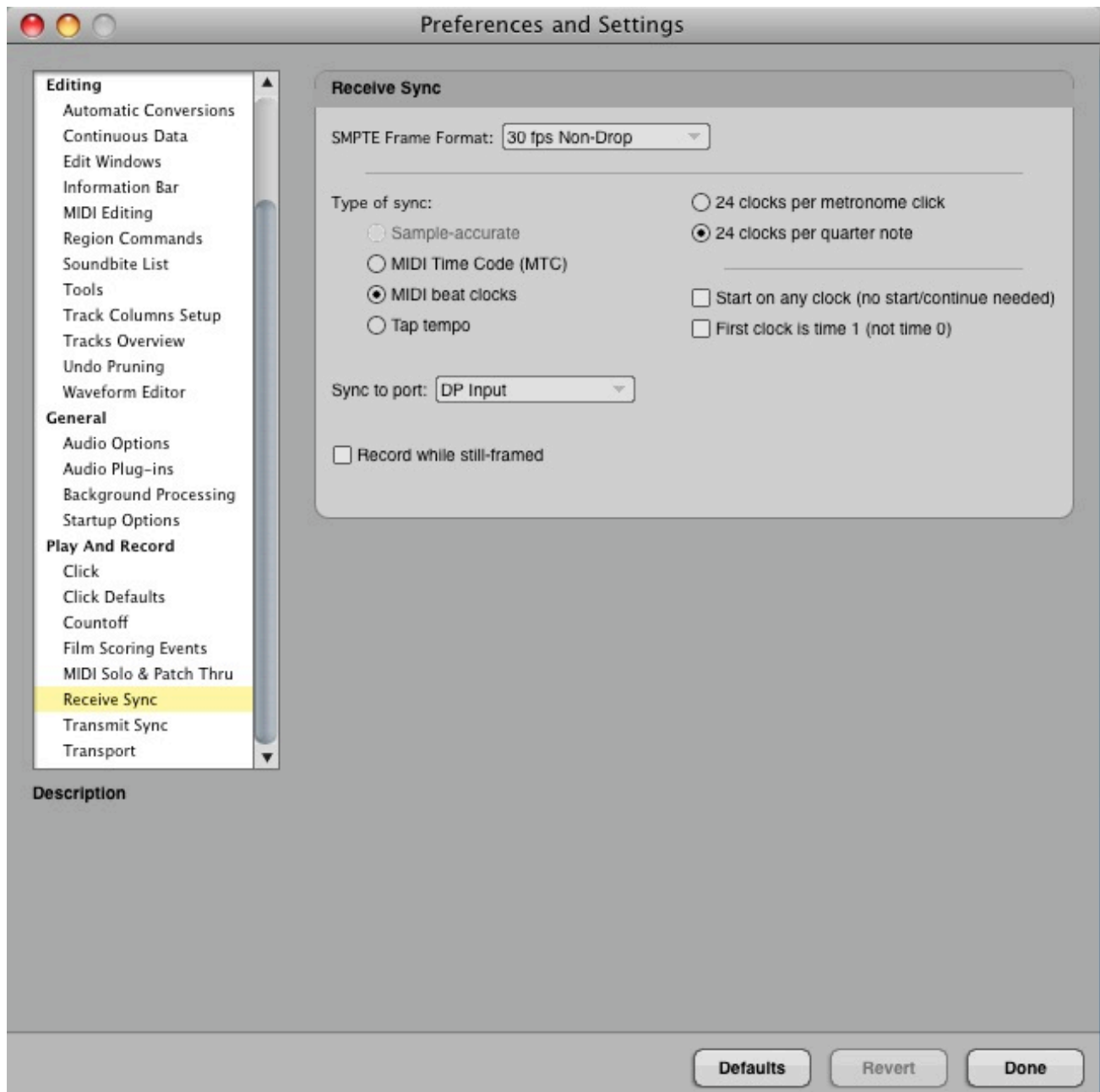
NOTE – if you are not using DP, then you will have to figure out another way to create a virtual midi port if you intend to use a different sound source then Jamstix for drumsounds. There are several alternatives including IAC, Midi Patchbay, MidiPipe, etc. Your DAW may also provide this capability similar to DP.

Create a new DP project file, and set it up with an instrument track for Superior2(or your drum plugin), and a midi track to receive midi from Reaper. Assign the input port for the midi track to “**DP Input 2-Any**” and route the output of that track to Superior2. Make sure the track is record enabled:



Set up external sync. DP will then be slave to Reaper. Go to the following menu:

Setup->Receive Sync...



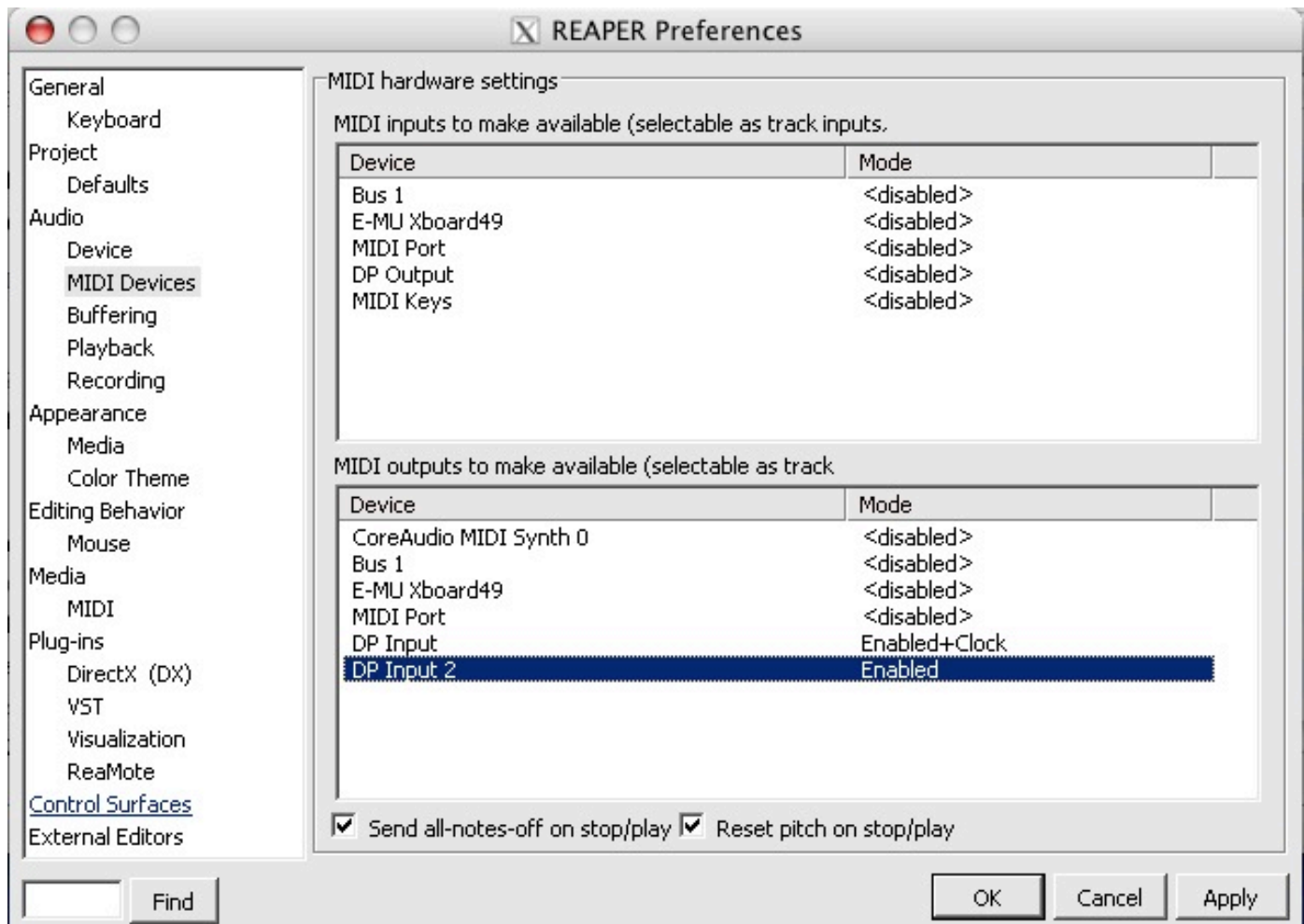
make sure to select **midi beat clock** as the method of sync and choose “**DP Input**” as the midi port to listen for sync. Click “Done”

From the Setup menu select “**Slave to External sync**”

When you hit the PLAY button in DP you will notice that the play button blinks green and does not play. This is because its waiting for Reaper to tell it to go. It is now in slave mode. For now, click STOP.

Make sure that when you play sounds on your drum plugin in DP, you hear the audio. It should be routed to JackRouter 1-2 and you may have to go into JackPilot to configure connections in order to hear audio. Do not proceed until you are able to hear your drum plugin audio. Now go to Reaper, if not launched already, then launch it and insert Jamstix into a track.

Go to the **Options->Preferences** menu and go to “Midi Devices”



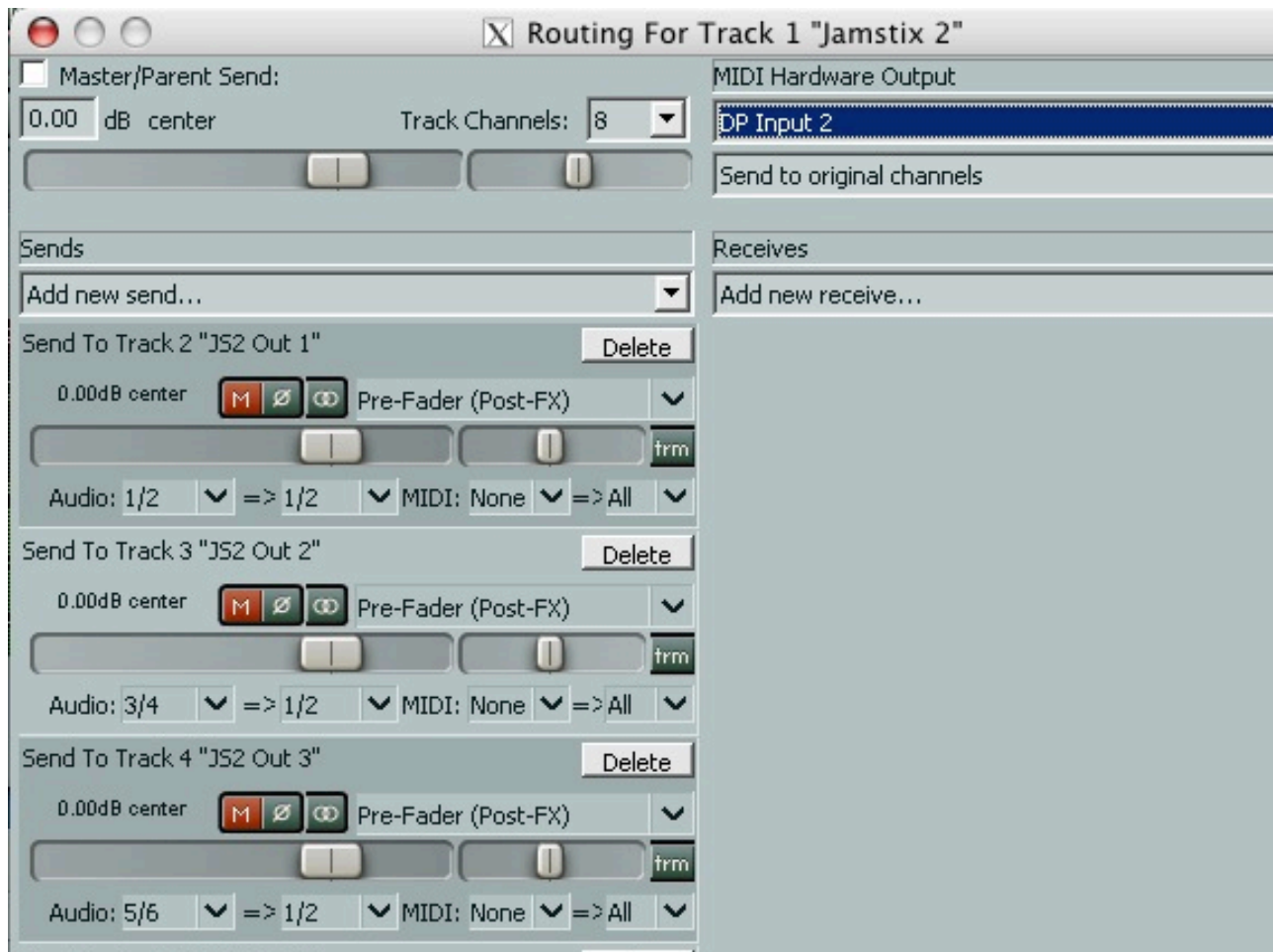
make sure that you see in the Output section “**DP Input**” and “**DP Input 2**” as available midi output ports. You want to configure **DP Input** to be enabled and also with clock output. **DP Input 2** should be enabled but not with clock. Right click on <disabled> to configure them. Click OK.

That should configure the **DP Input** port to carry midi clock information over to DP so that DP can slave to Reaper.

In Reaper on the track Jamstix is in, click the tiny **io** button.



Configure the **Midi Hardware Output** for this track to be “**DP Input 2**”. This should route the midi output from Jamstix over virtual midi pipe to DP where it will be received by the midi track you already configured there that is patched into your drumsound VST.



Theoretically you should be ready to go. Setup Jamstix with a kit and grooves and hit the Reaper PLAY button. Jamstix should start playing and you should be hearing drum sounds triggered in the drum vst you're hosting in DP. But at this time DP is not playing to the timeline itself. Just the midi events from Jamstix are being routed over "DP Input 2" pipe over to DP and triggering your drumsound VST.

NOTE – I have found some finicky problems sometimes this does not seem to work right away. I have found that by opening the midi device preferences in Reaper WHILE ITS PLAYING and disabling/enabling it a few times eventually got things working. Somehow some data needs to be seen on the pipe for Crossover to do the right thing. Play around with it until it works. I have found also that once I save the working project and open it again, it seems to work without this hassle.

Now to test the sync between Reaper and DP, go over to DP and hit the PLAY button. It will blink and not play yet. Go back to Reaper and rewind to the beginning and hit PLAY. DP should start playing or recording in sync with Reaper. In fact if you move the song position pointer in Reaper, the pointer in DP will also be moved to the same place. They will remain in sync, with Reaper as the master.

NOTE – like above, if this fails to work, try futzing with the midi devices dialog in Reaper, turning that port on and off until it starts to work. Its finicky for some reason, but it will work eventually.

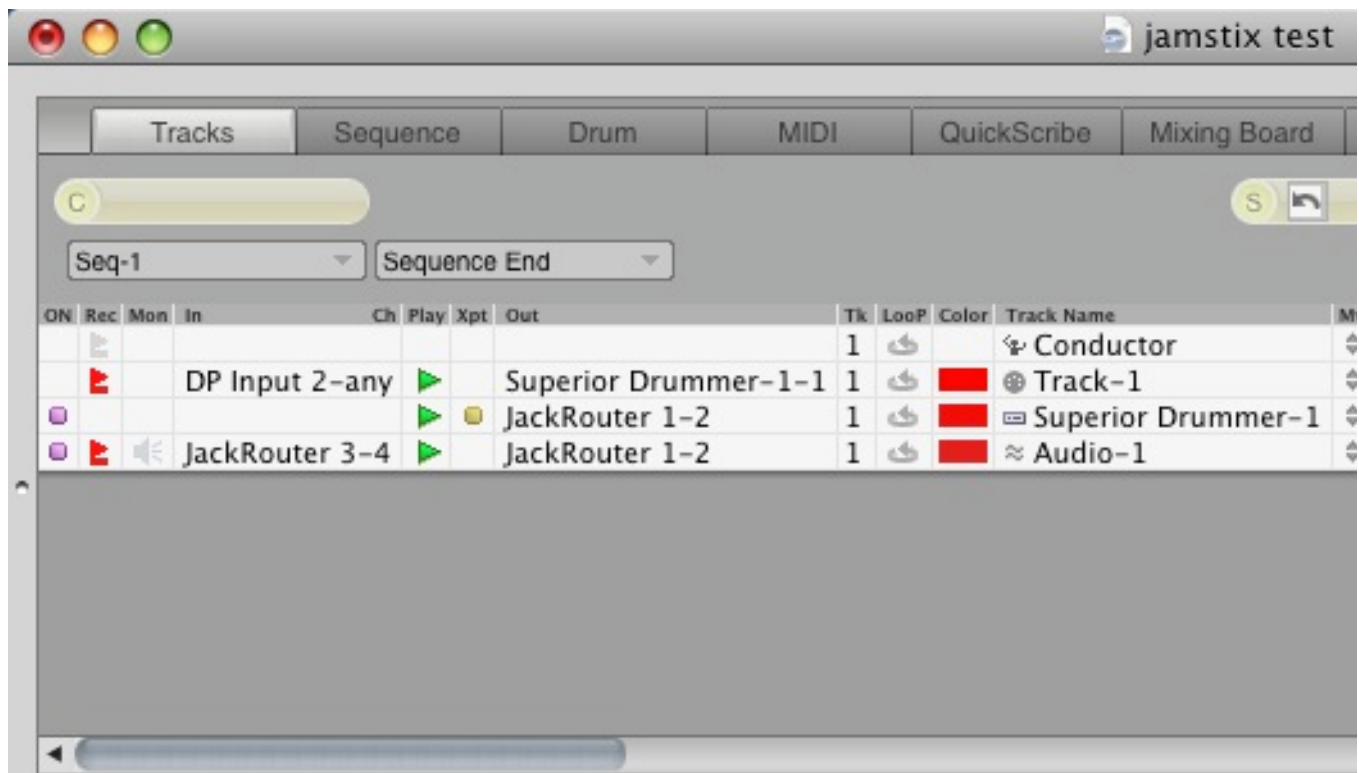
So at this point you should theoretically be hearing Jamstix audio played out of Reaper through the JackOSX router to your audio hardware; and the Jamstix midi will be piped over the DP and play back whatever drumsound VST you have loaded there, that audio playing through JackOSX routing to the soundcard also. And if sync is working you should see both Reaper, Jamstix and DP all staying in perfect sync on the song timeline.

Jamstix/Reaper midi+audio to DP

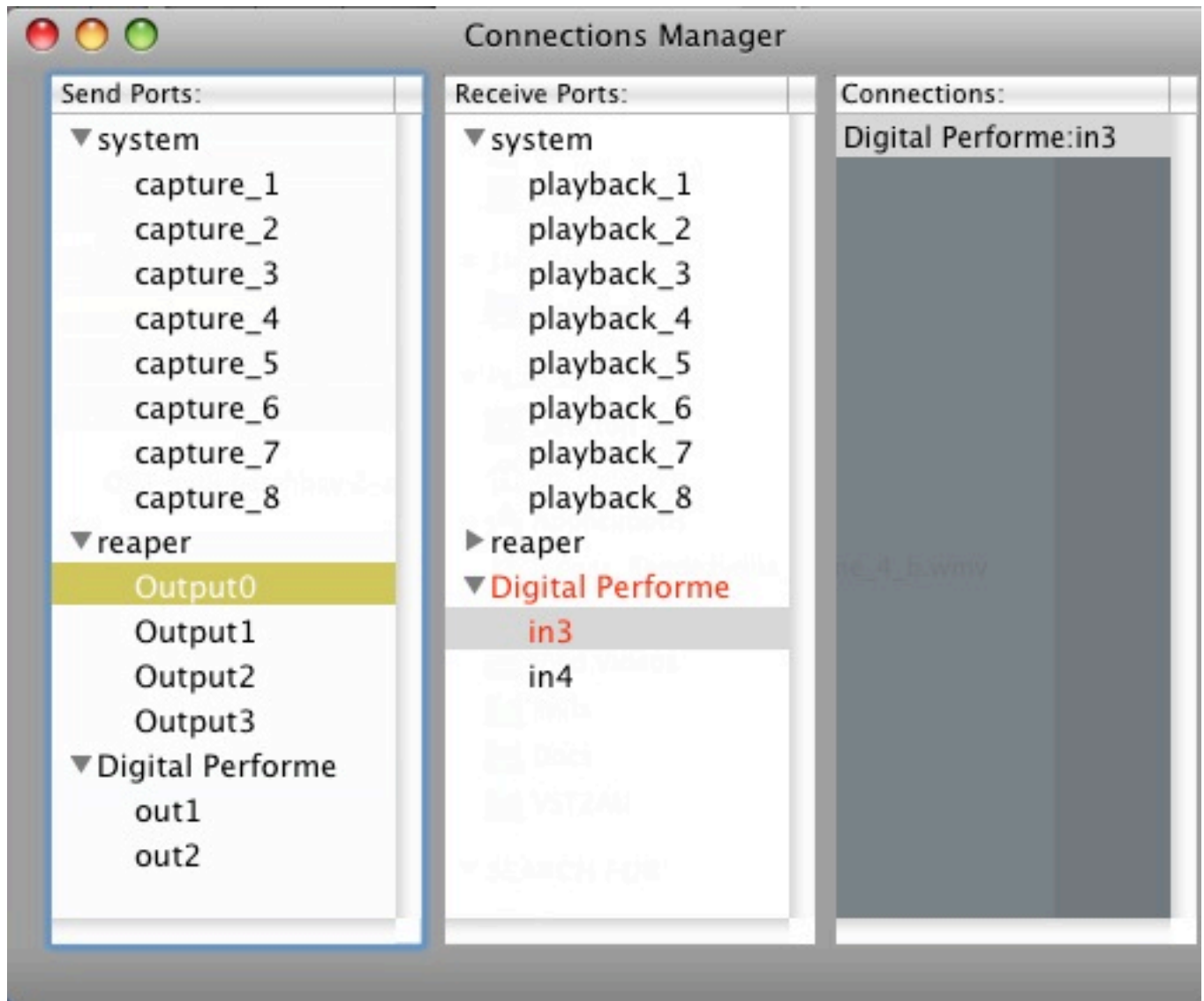
The final test is to also route the audio from Reaper over to the inputs of Digital Performer so that you can do the entire mixdown through DP. This is very easy and actually consumes less CPU.

First you need to create an audio track in DP that will receive the audio from Reaper/Jamstix. You could theoretically create up to 4 audio tracks since Jamstix supports that many.

Configure them to use JackRouter 3-4 as audio input ports. Make sure this audio track is record enabled and output sent to JackRouter also.

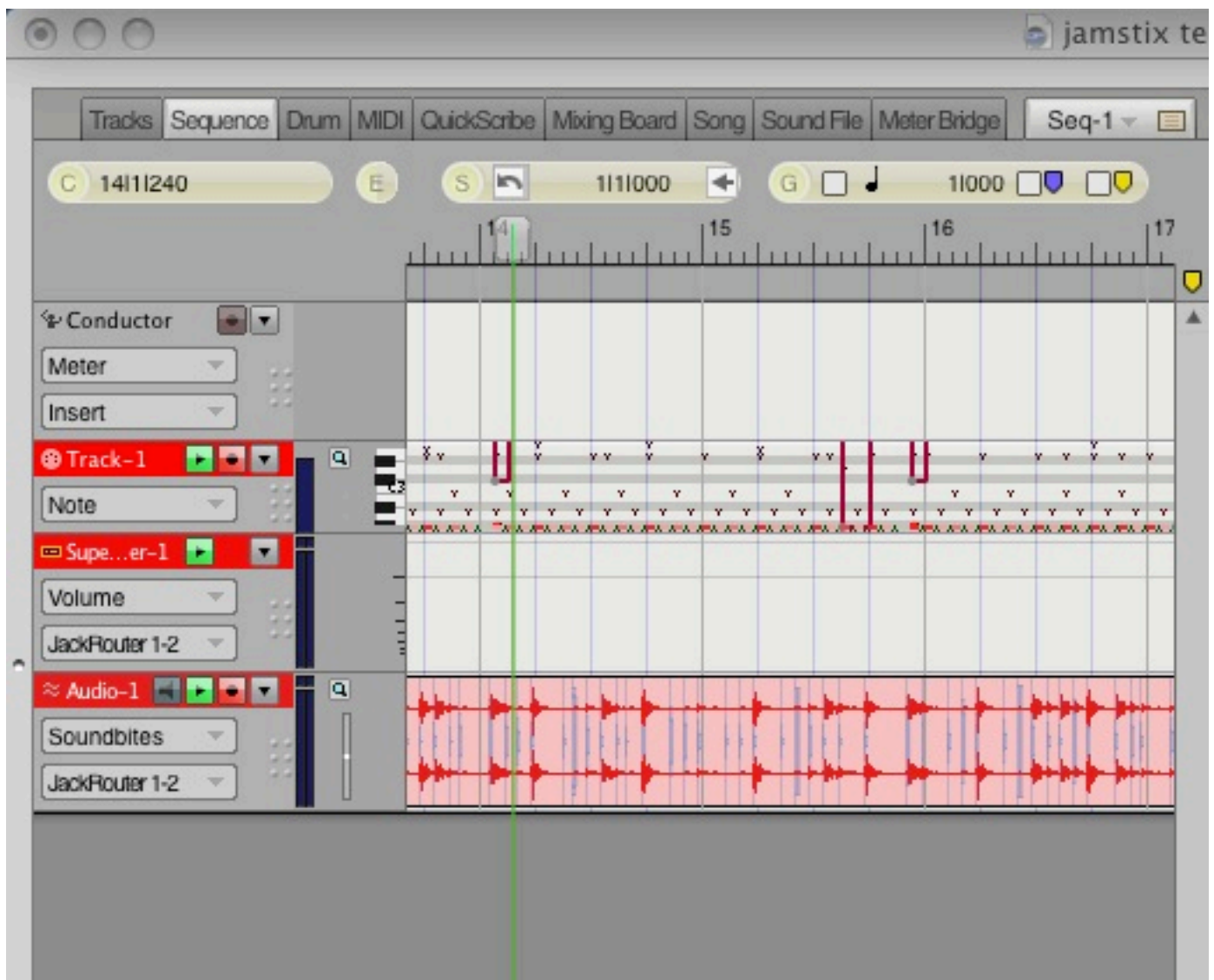


After doing that you should see some new ports show up in the JackPilot Router. You will need to connect the outputs from Reaper to the inputs of Digital performer using JackPilot Router. As before, select Reaper->Output1 and then double click on Digital Performer->in3 and do the same for the second channel.



After doing that, the output from Reaper should be routed into DP instead of being routed directly to the audio card. You should be able to hit kit pieces in Jamstix/Reaper and hear them play through DP and see the mixer led's light up in DP that its receiving that audio. If not, then check the JackPilot connections and everything else to make sure its correct before proceeding. Remember, Reaper connects to DP, DP connects to system out.

Now you're ready to go. Make sure the midi and audio tracks are record enabled in DP. Hit the RECORD button in DP and then the PLAY button. The green PLAY button will blink. Go back to Reaper, rewind and hit PLAY button. You should start hearing both Jamstix sounds and your drumsound VST sounds being played together through DP's mixer. Additionally, the midi data will be recorded into the DP midi track and the Jamstix audio will be recorded into the DP audio track, looking like this:



Follow-up notes.

Whew. Not simple. Once you do it a few times, its not that bad. Also you can save Reaper/Jamstix setups to get you going each time. If you need to do something like send midi back from DP to Jamstix to trigger Jamstix sounds, I leave it to you to play around with virtual midi ports and see what you can do.

A few things to watch out for: It matters a lot the order by which you startup applications and to a certain degree shut them down. ALWAYS start JackPilot first and stop it last. Additionally, in this setup you should start DP before starting Reaper since DP is hosting the virtual midi ports.

Its not a bad idea to “archive” the CrossoverGames bottle after you get it all working too. In the future if you change to a new version of CXGames or hose your setup somehow, you can restore the archive and be back to where you started. See the CXGames manual for help on that.

Notes about Virtual Midi and Reaper

After playing around with this setup a while, it has become clear that virtual midi ports on OSX, together with Wine, is a flakey business. I had a lot of problems with certain versions of Reaper. So far, Reaper 3.0 seems to be working rather flawlessly without any finicky problems with the virtual midi ports. So I highly recommend you use Reaper 3.x in order to avoid finicky virtual midi ports.

Notes about Virtual Midi timing

I have also found that the virtual midi data transmission has inconsistent latency and can depend a lot on what is happening on your system. That leads to midi timing in the host DAW which can fluctuate by a little or a lot from what was sent from Reaper. It's not consistent, so there is no way to correct it.

What I recommend is that you use this virtual midi setup only for the *composing* process when you need to hear your drum trucks together with other tracks you have in your DAW and you're still working on the drum tracks. And also when you are using jamstix to drive OSX superior, or something like that.

It will be tight enough to work this way. Actually some people won't be able to tell the difference. Most people can't hear a few milliseconds timing difference. Nonetheless, I recommend that once you have the drum track worked out in Jamstix, drag and drop the jamstix part into the Reaper track to create an actual midi part. Then export that midi part to the file system, which you can then drag into a DP midi track; and it will be rock solid perfect for your final mix.

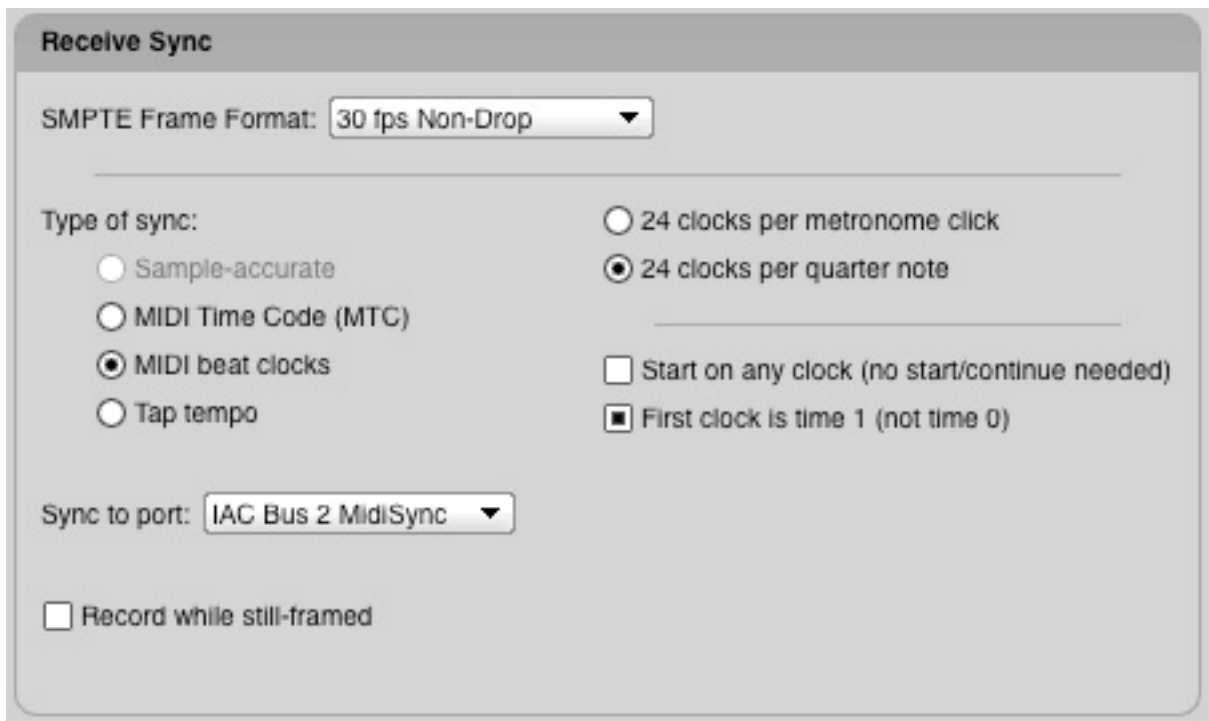
I recommend you do the same thing with audio. If you are wishing to use Jamstix audio in your project, then use the JackOSX routing to hear it while you work on it, through your host DAW, but once you have the drum parts worked out, render them in Reaper to a WAV file and then import that WAV file into a DP audio track. They will then be basically sample-accurate.

In addition to the problem mentioned earlier about midi jitter, the audio sent over JackOSX is also not reliable in terms of timing. I did a lot of problem solving to try to track this down and it turns out that the culprit is related to midi sync. In the setup I described, midi sync is being used to synchronize Reaper and DP, with DP as the slave. The problem is that it takes an unpredictable amount of time to send the first midi clock start command over virtual midi to DP and then for DP to start playing/recording in sync with Reaper. Reaper continues to send 24 pulses per beat to keep DP in sync, but the gist of it is that DP is receiving those events late, by an unpredictable amount of latency. DP will always be running a little bit behind Reaper, by an unpredictable amount because it won't start playing/recording until it receives the first midi sync clock. What's even worse is that when you hit play in Reaper, it starts sending audio immediately over to DP, but DP doesn't start recording until later when the first midi sync message finally gets there. So the first bit of audio is never recorded and everything that is recorded will be sitting EARLY on the DP track from what it should be, because the audio is getting there on time, but DP is running behind.

There is no way to fix this. The best thing is to render the WAV in Reaper when it's done and import that into DP for rock solid timing.

That being said, I did find a few tricks that can at least get the JackOSX audio to be synchronized close enough to reality that you can't tell any difference and can work out your song without worrying about it. But I still recommend you render the final jamstix audio in Reaper.

For one thing, go into the DP Receive sync preferences, and make sure to select the option, **First clock is time 1 (not time 0)**



That will at least make it so that when DP receives the first midi clock pulse, it will consider it to be the second pulse, rather than the first and so will be LESS behind then if it assumed it to be the first pulse. However it generally won't be enough, it will still probably be late compared to the audio that is coming from JackOSX.

I configured JackOSX with a buffer setting of 128, and this seemed to create a bit more audio latency such that the audio coming across to DP was arriving late also, much closer to DP's clock. Because this is all dependent on inconsistent virtual midi timing, it will never be the same on each run, but I have found that its close enough, within a couple milliseconds, to work on the song. If you make the JackOSX buffer smaller, then the audio will be ahead of DP. If you make it bigger, then the audio will behind DP. I found 128, along with the above mentioned receive sync setting, brought it within a millisecond or two early, which is actually perfectly good enough, practically speaking. If I could have set it at 192, that might have been perfect or a few ms late, which I would prefer to early.

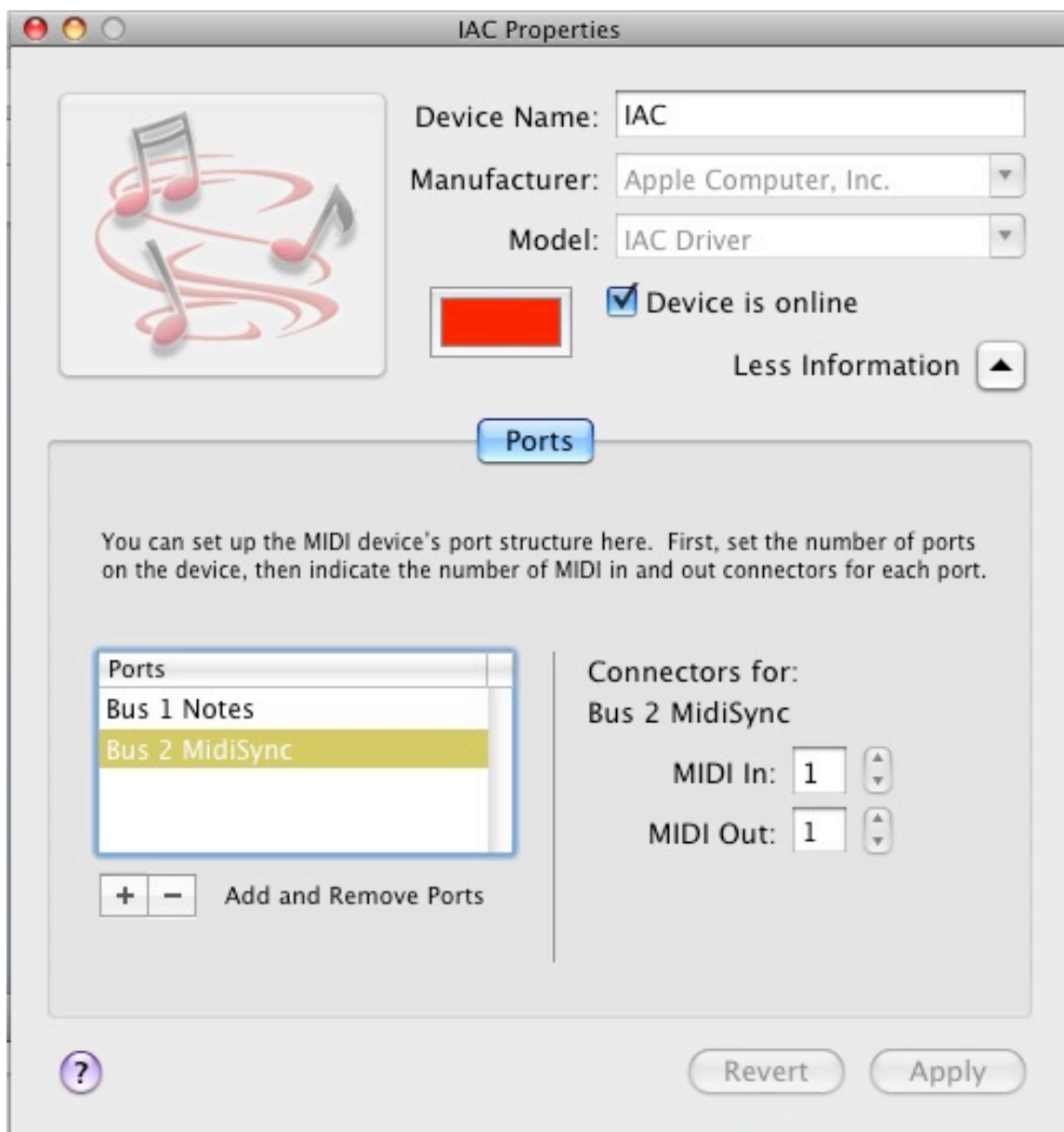
It will depend a lot on your system, so experiment by importing a rendered wav from Reaper and putting it on another track, then try to record audio from JackOSX like I did in this tutorial and compare the two wave forms to see how early or late it is. Adjust the JackOSX buffer until it's the closest you can get it and then live with that. I got mine within 2ms early.

Remember, that is only for while you are using virtual midi and JackOSX to work on your drum tracks in sync with DP. Render the final wav from Reaper and import it into DP, that will be rock solid perfect timing. If you still insist on recording from the JackOSX pipe into DP, then remember that if the JackOSX audio is early, then the very beginning of bar 1 will not get recorded. So make sure to leave bar one empty or create some preroll.

Notes about using other OSX DAW's besides DP

Many people have asked the question about using other hosts besides DP. Any other sequencing host will work, as long as its supporting CoreMidi. At the time I wrote this guide, I could not get IAC to work, I think related to Reaper midi bugs, but at this time, IAC is working perfectly fine, so you do not even need to create DP virtual midi ports as I described on page 19, which is a DP-only feature.

The easiest solution is to simply create a couple IAC busses using the Apple Audio/Midi Utility. Open that utility, double click the IAC icon, and create a few like this:



Make sure they are created before launching Reaper and then enable them in Reaper's preferences. They have been working great for me. You can also experiment with a free software called MidiPipe, which provides some helpful monitoring capabilities on top of virtual midi port creation.