UniWire Manual Supplement

This document discusses UniWire and how to use it with Receptor

Muse Research, Inc.
970 O'Brien Drive
Menlo Park, CA 94025
USA

Tech Support: (650) 326-6180
Main Office: (650) 326-5400

February 17, 2006

UniWire™ Manual Supplement

UniWire™ is a new way to connect one or more Receptors to a computer-based digital audio workstation using only Ethernet cables—no audio or midi cables are needed. In addition, UniWire instrument and effects plugins are installed on your host computer, which allow your digital audio workstation to communicate with Receptor as if it were both a virtual instrument and a virtual effect plugins. This means that, unlike "old school" external sound and effects modules, Receptor integrates with your digital audio workstation exactly like a locally hosted plugin — but without using your host computer's precious CPU cycles.

In addition, UniWire lets you use Receptor's built-in MIDI and superb audio I/O circuitry as another 'audio and midi interface' for your computer-based DAW.

Software Installation

There are two halves to the UniWire equation — the Receptor-side software and the host-side plugin software.

- **Receptor-side software**

  UniWire requires Receptor version 1.5 or higher. You can see which version of Receptor you have by pressing the SETUP button on Receptor's front panel, then rotating the top display knob until you see the "System Version" parameter. If the version displays 1.5.x or higher, your Receptor is UniWire compatible. If it shows 1.0.x, 1.1.x, 1.2.x, 1.3.x, or 1.4.x you will need to update your Receptor to v1.5 (or higher), which you can download from www.plugorama.com. Follow the online installation instructions to update your Receptor.

- **Host-side plugin software**

  UniWire is currently available as a VST plugin for Windows and as a VST and AU plugin for Macintosh OSX. In order to use UniWire with your compatible host DAW, you'll need to install these plugins on your host computer. To do so, simply double-click the desired UniWire plugin installer application and follow the on-screen instructions.

Hardware Connection Basics

If you're accustom to using external sound and effects modules, you're also accustom to routing a lot of audio and MIDI cables between these devices and the audio and MIDI interfaces connected to your host computer. No more. With UniWire, all audio and MIDI communication between your host computer and Receptor travels over a single Ethernet cable.

Requirements

Muse Research requires that your computer have, at least, 100Base-T Ethernet and that any switcher (or switched router) in your network also be certified for at least 100Base-T. If you plan to communicate with multiple Receptors using UniWire, Muse Research strongly recommends that your computer has Gigabit Ethernet and that you use a Gigabit Ethernet switcher, as this combination will give you more simultaneous tracks over UniWire.

UniWire does not support 10Base-T or wireless (802.11b or g), as these protocols are too slow to insure reliable, low-latency audio and midi communication.

Connecting a Single Receptor with UniWire

If you have a single Receptor and no other Ethernet hardware is connected to your computer, you can use a simple "Ethernet crossover" connection to connect Receptor to your host computer. Simply use a special type of Ethernet cable, called a *crossover* Ethernet cable, and connect it between Receptor's Ethernet port and your computer's Ethernet port. More detailed discussion of network connections are contained within the Receptor manual.

If there is already another Ethernet device (such as a cable/DSL modem, network printer, or additional computer) connected to your host computer's Ethernet port, you will need to use an Ethernet Router or Switcher to connect multiple Ethernet devices to your host computer. 100Base-T is the minimum recommended speed, but Muse Research suggests using a Gigabit switcher if ever plan to use more than one Receptor. If you use an Ethernet Switcher, simply connect all your external Ethernet devices (including Receptor) to the Switcher, then connect the Switcher to your computer. The concept is very similar to using USB hubs, but this is for Ethernet devices. Again, your Receptor manual contains more detailed information about networking.

Connecting Receptor for use as Audio/MIDI Interface

If you plan to use Receptor (and *only* Receptor) as your computer's audio and MIDI interface, then the single Ethernet cable connection will suffice. However, if you wish to use Receptor's audio ports *in combination* with your host computer's audio interface, then you must also connect a S/PDIF cable between Receptor and your host computer's audio interface. The S/PDIF cable is required to make sure that both audio interfaces sync to the same digital clock — the S/PDIF cable is not required if you do not plan to use Receptor's built-in Audio or MIDI ports.

Connecting Multiple Receptors with UniWire

If you have multiple Receptors, you will need to use an Ethernet Switcher to connect them all to the Ethernet port on your host computer. If you plan to use multiple Receptors, Muse Research strongly recommends that your host computer has Gigabit Ethernet and that your Ethernet Switcher also be Gigabit Ethernet. Each Receptor contains a 100Base-T Ethernet port, which is capable of carrying 32 channels of audio (16 stereo channels) to/from your host computer. By using a Gigabit Ethernet network, your host computer will be able to route up to 320 channels of audio across multiple Receptors on your network.

UniWire Overview

Before UniWire

Before UniWire, you connected a hardware sound or effects module to a DAW by connecting a pair of MIDI cables between the device and the MIDI interface connected to your host computer. Similarly, you connected multiple audio cables (analog or digital) between the external device and your host computer's audio interface.

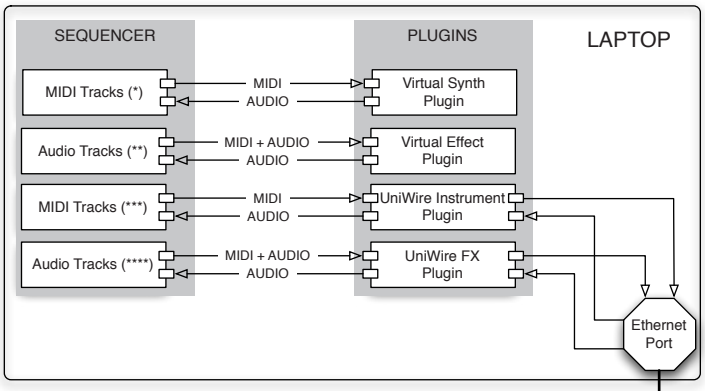
In software, you would then configure a MIDI track to send data to the external hardware (if it was a synth) and another track to receive audio from the device.

If your Receptor is currently connected this way, feel free to rip out those audio and MIDI cables and toss them into that cardboard box in the corner of your studio.

After UniWire

UniWire uses only an Ethernet cable to exchange all audio and MIDI data between one or more Receptors and a sequencer running on your computer of choice (hence the name "UNIwire"). All communication between these devices is handled by a UniWire plugin, which you instantiate just like any plugin on your host machine. The UniWire plugin handles all the audio and midi data transfers between your host sequencer and your Receptor(s), streamlining your studio and your workflow.

The following illustration shows just one way in which Receptor and UniWire might integrate with a laptop running a sequencer, and how Receptor can greatly expand the processing capabilities of your computer beyond the limits of its internal CPU.



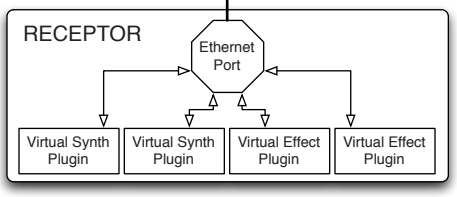
In a typical laptop, MIDI Tracks (*) and Audio Tracks (**) send data to and from plugins stored on the laptop (as shown in the above diagram).

But what happens if your laptop runs out of CPU? Receptor and UniWire to the rescue!

Simply instantiate a UniWire plugin on additional MIDI Tracks (***) and data from that track is sent to Receptor, where it's processed and returned -- just as if the plugin was running on your laptop.

Similarly, you can instantiate UniWire plugins on additional Audio Tracks (****) and UniWire will use Receptor's plugins to process those tracks.

Everything happens over Ethernet (no audio or midi cables are required) and, thanks to the UniWire plugin, Receptor integrates with your sequencer just as if it were built-in to your laptop.



Basic UniWire Operation

At this point, you should have:

- connected Receptor to your computer's Ethernet port (or the Ethernet network connected to your computer)
- updated Receptor to its latest system software
- installed the UniWire plugins on your desktop computer.

If not, see the previous sections to learn how to connect your hardware and install your software.

Setting Up Receptor to Use UniWire

The following is a basic overview of how to set up and use UniWire with your host sequencer of choice. This overview assumes the simplest and most basic UniWire operation — that is, you use only your host computer's audio and MIDI interfaces (not Receptor's audio and MIDI I/O), and that you communicate with Receptor via a UniWire plugin instantiated in your host sequencer. Other, advanced modes of operation are available and are discussed later in this manual, along with detailed descriptions of all UniWire parameters contained in both Receptor and in the UniWire plugins.

1 Your first step is to enable UniWire on Receptor.

To do this from Receptor's graphical interface, click the SETUP tab and check the ***Enable UniWire*** *option in the new UNIWIRE section in Setup View.*

To do this from Receptor's front panel, press the SETUP button and rotate the top display knob until the top line shows the "UniWire" option, then rotate the bottom knob to select `Enabled`.

Once UniWire is enabled, Receptor will sync to your host computer, meaning that Receptor's Master Sample Rate and Sample Buffer Size settings will be determined by UniWire. As such, these options will be disabled in Receptor's Setup view. In addition, the MIDI Tempo Source will be set automatically to "External-UniWire", meaning any synchrosonic effects (LFO's, arpeggiators, etc) running on Receptor will be synchronized to your host computer over UniWire.

2 For this basic tutorial, make sure that the **Enable Receptor Audio+MIDI I/O** option is disabled and the **Bypass MIDI Filter on single channels** option is turned ON. Both these options appear in Setup View.

To do this from Receptor's front panel, with the SETUP button lit, make sure that the `UniWire Receptr I/O` *parameter is set to* `Audio+MIDI Disabled`*, and that* `UniWire MIDI Filter` *is set to* `Bypassed`*.*

UniWire as an Instrument

This basic example shows how to set up UniWire so that your host sequencer works with Receptor as if it were a virtual instrument.

1 Enable UniWire on Receptor and configure the unit as described previously, on page 5.

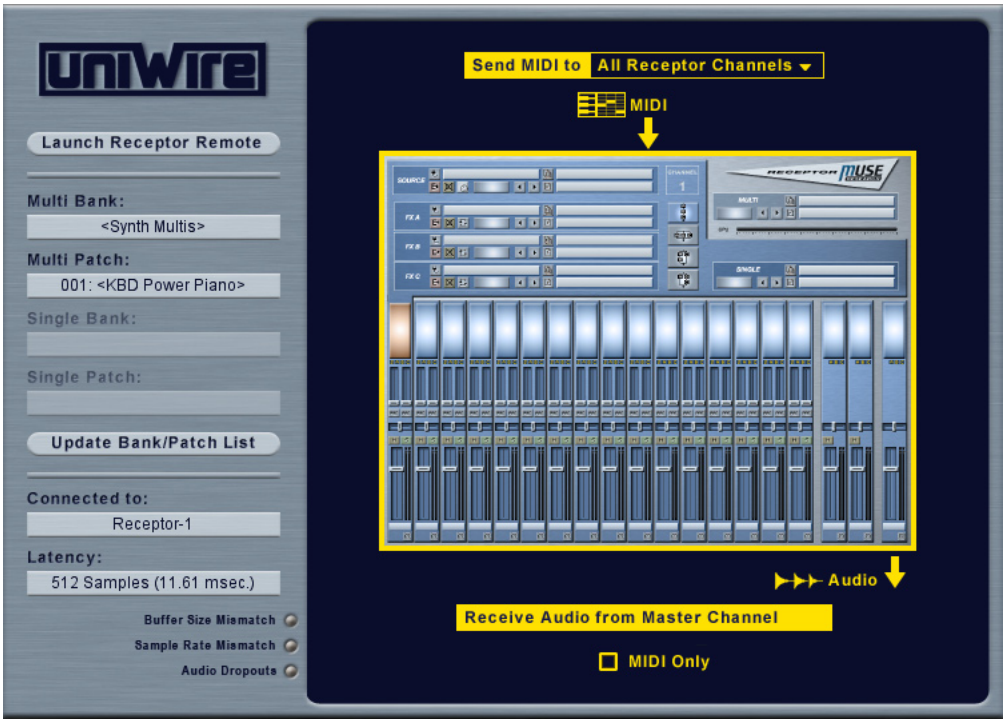
2 In your computer sequencer, create a new virtual instrument track and instantiate the **UniWire Instrument** on that track.

Every host application has a different way of doing this. See your sequencer manual to learn how to set up virtual instrument tracks and work with virtual instruments. In addition, some host-specific examples are included later in this documentation.

3 In your computer sequencer, open the edit window for the **UniWire Instrument** you just instantiated.

Again, this process is different for every host. See its manual to learn how to open graphical editor windows for plugins.

The UniWire plugin interface appears in your host application.



4 From the UniWire plugin's **Connected To** menu (in the left column), select the Receptor with which you want to communicate.

By default, UniWire communicates with the first Receptor in the list meaning, if you have only one Receptor, UniWire always selects it automatically.

5 From the UniWire plugin's **Latency** menu (in the left column), select the best latency setting for your intended use.

The Latency menu always displays values calculated from the sample rate and buffer sizes set in your host sequencer. The lowest possible latency is always 2x your host sequencer's buffer size.

Lower latencies limit Receptor's potential polyphony but allow instruments and effects to be played satisfactorily in real time. If, however, you're using Receptor for off-line sequencing and mixing, higher latencies will allow Receptor to process more data, and you can adjust your host sequencer's latency compensation settings accordingly.

6 Use the UniWire plugin's **Audio Dropouts** LED to monitor whether or not the current latency setting results in any dropped audio buffers. Clicking the **Audio Dropouts** LED resets it.

7 In the large routing area in the right portion of the UniWire plugin window, select **All Receptor Channels** from the **Send MIDI To** menu.

*When you select **All MIDI Channels**, the UniWire plugin addresses Receptor exactly as if you had actual audio and MIDI cables connected to Receptor. With this option selected, your host computer sends 16 MIDI channels over Ethernet (rather than MIDI cables), which Receptor routes according to the MIDI Filter settings in each Receptor Instrument Channel. Similarly, audio from Receptor's MASTER channel is routed over UniWire back into your host application, just as if you had connected a pair of audio cables between Receptor and your host computer's audio interface.*

*You'll notice that the **Send MIDI To** menu gives you the option of addressing each Receptor channel individually. When you choose one of these single-channel options, UniWire is able to send all 16 MIDI channels to each Receptor channel, meaning Receptor operates as if it had multiple MIDI ports, rather than just one. This mode of operation, which allows a single Receptor to respond to hundreds of MIDI channels, will be discussed later in this manual.*

8 From the UniWire plugin's **Multi Bank** and **Multi Patch** fields, select the Multi patch you wish to load into Receptor.

If you modified any of your banks or patches since opening the UniWire plugin, simply click the **Update Bank/Patch List** *button to make sure the plugin's Bank and Patch lists match those in Receptor.*

9 Play a MIDI keyboard connected to your host sequencer and, if its 'keyboard thru' feature is enabled and the track is configured properly, you should now hear Receptor just as if you had connected it to your computer using audio and MIDI cables.

Alternately, if your sequencer track contains some MIDI data, you can play your sequence and UniWire will send that MIDI data to Receptor, where it will play the instrument(s) instantiated on Receptor, then send the audio back into UniWire -- just as if you were using a Virtual Instrument on your host computer.

10 If you wish to edit Receptor, simply click the **Launch Receptor Remote** button.

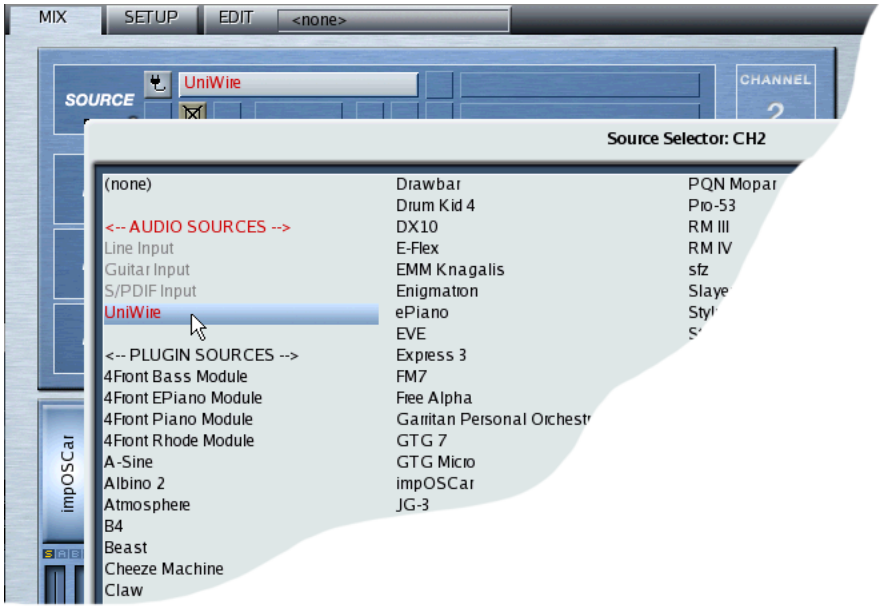
As its name implies, this will automatically launch the Receptor Remote application on your hard drive and allow you to graphically edit and configure Receptor over Ethernet.

UniWire as an Effect

This basic example shows how to set up UniWire so that your host sequencer works with Receptor as if it were a virtual effect.

1 Enable UniWire on Receptor and configure the unit as described previously, on page 5.

2 On one of Receptor's instrument channels, assign the Audio Input called "UniWire" to the Source slot then assign effect plugins to one or more FX slots (A, B, or C).



The previous illustration shows how to accomplish this with Receptor's graphical interface. To do this on Receptor's front panel, set Receptor to the desired channel (see your Receptor manual), press the SOURCE button, rotate the top display knob to select the `Source` *parameter and rotate the bottom display knob to select* `UniWire`*. Press the bottom display knob to apply your selection.*

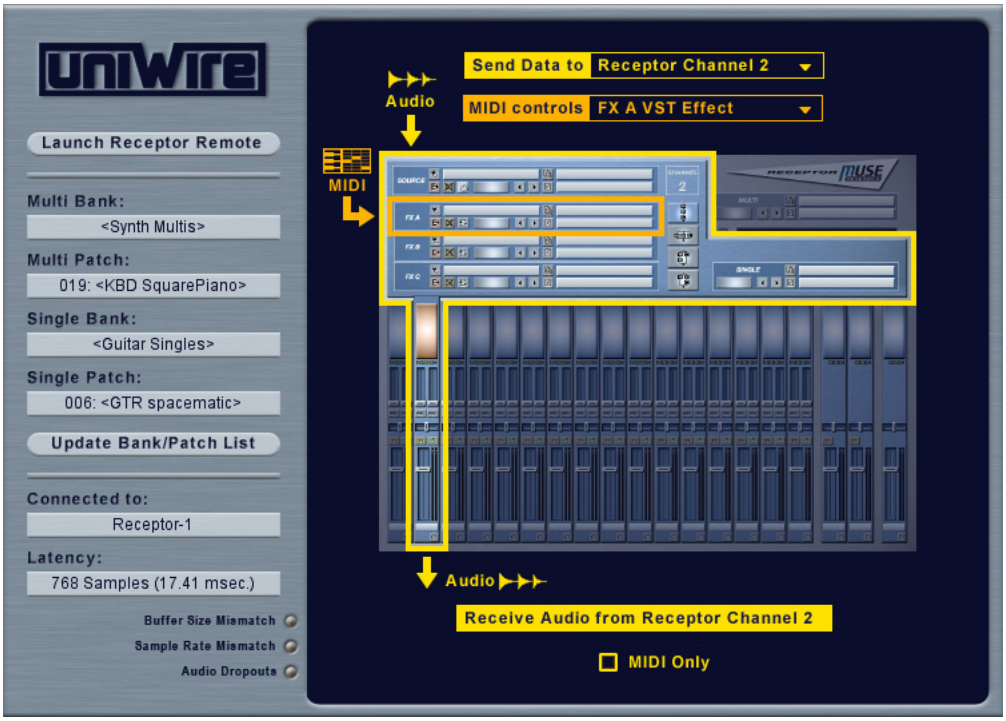
3 In your computer sequencer, select an audio track and instantiate a **UniWire FX** on that track.

Every host application has a different way of doing this. See your sequencer manual to learn how to route audio tracks through virtual effects. Some host-specific examples are included at the end of this documentation.

4 In your computer sequencer, open the edit window for the **UniWire FX** that you just instantiated.

Again, this process is different for every host. See its manual to learn how to open graphical editor windows for plugins.

The UniWire plugin interface appears in your host application.



5 From the UniWire plugin's **Connected To** menu (left column), select the Receptor with which you want to communicate.

By default, UniWire communicates with the first Receptor in the list meaning, if you have only one Receptor, UniWire always selects it automatically.

6 From the UniWire plugin's **Latency** menu (left column), select the best latency setting for your intended use.

The Latency menu always displays values calculated from the sample rate and buffer sizes set in your host sequencer. The lowest possible latency is always 2x your host sequencer's buffer size.

Lower latencies limit Receptor's potential polyphony but allow instruments and effects to be used in real time. If, however, you're using Receptor for off-line sequencing and mixing, higher latencies will allow Receptor to process more data, and you can adjust your host sequencer's latency compensation settings accordingly.

7 Use the UniWire plugin's **Audio Dropouts** LED to monitor whether or not the current latency setting results in any dropped audio buffers. Click the **Audio Dropouts** LED to reset it.

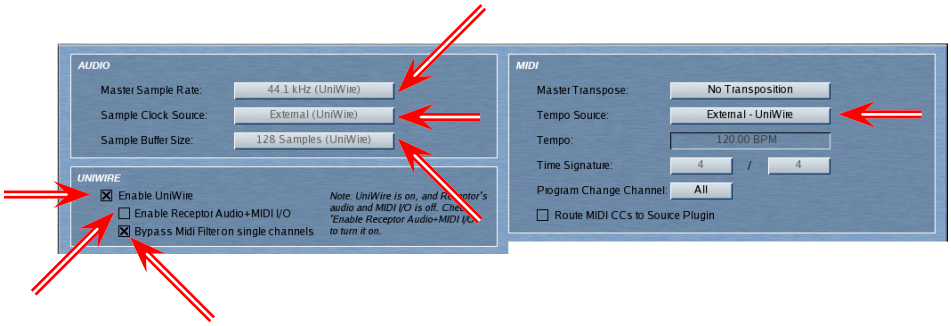
8 In the large routing area in the right of the UniWire plugin window, use the **Send Data to** menu to select the channel on which the desired effects chain appears (the example above shows Receptor Channel 2).

In general, when using Receptor as an effects rack, it's best to have the UniWire plugin communicate directly with a single Receptor channel, rather than All Receptor Channels. This lets you route different audio channels through different effects chains on Receptor These modes of operation (communicating with Receptor on a single channel vs. ALL channels) will be discussed later in this manual.

9 Play your host sequencer and you should hear the audio track you assigned to UniWire playing through the effect(s) you assigned on Receptor, just as if you had connected Receptor to your computer using audio and MIDI cables.

UniWire—Receptor Setup Details

The Setup View parameters indicated in the following illustration all have some effect when used with UniWire.



In addition, there is a new UniWire Audio Input option available in the Source Selector for each Receptor instrument channel. Each of these new UniWire options is discussed in the following sections.

Enable UniWire option

The first thing you'll notice is a new section on the Setup page, called UniWire. The first parameter in this section is called **Enable UniWire**. This is the master on/off switch for UniWire.

If this option is unchecked, then UniWire cannot be used and Receptor works exactly like it did before the existence of UniWire. That is, you must send MIDI data to Receptor via its built-in MIDI port, you send audio to Receptor using its built-in audio input jacks, and you get audio out of Receptor using its various analog and digital audio connections.

In order to use Receptor with the UniWire plugin, you must first enable UniWire on Receptor by checking the **Enable UniWire** option. When you enable UniWire, two sub-options become available: **Enable Receptor Audio+MIDI I/O** and **Bypass MIDI Filter on Single Channels**, both of which are discussed in the following sections.

Front Panel Operation: *On Receptor's front panel, this parameter is called simply "UniWire." To enable UniWire, press the SETUP button on Receptor's front panel and turn the top display knob until* `UniWire` *is displayed. Turn the bottom display knob to select between* `Enabled` *and* `Disabled`.

```
01  UniWire                    <>
III III  Enabled
```

Enable Receptor Audio+MIDI I/O option

Use this option to determine whether or not to use Receptor's own audio and MIDI inputs/outputs in addition to those provided by your host computer. Receptor and your host computer operate in very distinct and different ways depending on how you set this option as discussed below.

Front Panel Operation: *On Receptor's front panel, this parameter appears as "UniWire Receptr I/O". To turn it on, press the SETUP button on Receptor's front panel and turn the top display knob until* `UniWire Receptr I/O` *is displayed. Turn the bottom display knob to select between* `Audio+MIDI Enabled` *and* `Audio+MIDI Disabled`.

```
01  UniWire Receptr I/O<>
III III  Audio+MIDI Disabled
```

Enable Receptor Audio+MIDI I/O = OFF

When UniWire is enabled and this option is OFF, then your host computer and the UniWire plugin are in complete control of Receptor. Receptor's built-in audio and MIDI capabilities are disabled and Receptor becomes, essentially, an extension of your computer sequencer. This mode is the ideal choice for composers and producers who route all MIDI and audio into their host computer's audio and midi interfaces and wish to keep their computer as the center of their musical world. Specifically:

- Receptor's internal audio card is disabled. This means you cannot use any of Receptor's physical audio inputs or outputs — all audio input and output is handled by the audio interface connected to your host sequencer's computer. For example, if you have a guitar plugged into Receptor's front panel INPUT jack and have this option turned off, you will not hear your guitar. Rather, if you want to process your guitar, you would connect it to your computer's audio interface and route it to Receptor via UniWire.
- Receptor disables the **Master Sample Rate** parameter because the rate is now automatically set to match that used by the host sequencer.
- Receptor disables the **Sample Clock Source** parameter because the sample clock is now locked to UniWire.
- Receptor disables the **Sample Buffer Size** parameter because the buffer size is now automatically set to match that used by the host sequencer.
- **Tempo Source** is set to **External - UniWire**, but you may choose to use the **Internal** tempo instead. Clocking to **External - MIDI** is not available since, in this mode, Receptor's MIDI I/O is disabled.

Enable Receptor Audio+MIDI I/O = ON

When UniWire is enabled and this option is ON, then both your host computer and Receptor's built-in audio and MIDI capabilities are enabled. Receptor's audio and MIDI inputs are active and any audio routed to Receptor's physical outputs (either locally or through UniWire) will be heard. This mode is the ideal choice for performers who want to use Receptor's high-quality audio circuitry instead of (or in addition to) a computer audio interface. Specifically:

- Both Receptor's internal audio and your host computer's audio card are enabled but are not digitally synchronized, meaning they will drift apart slightly over time. To synchronize the two audio sources, Muse Research strongly recommends that you connect a S/PDIF cable between Receptor and your host computer's audio card, then sync Receptor to S/PDIF.
- Receptor disables the **Master Sample Rate** parameter because the rate is now automatically set to match that used by the host sequencer.
- Receptor disables the **Sample Buffer Size** parameter because the buffer size is now automatically set to match that used by the host sequencer.
- Receptor allows you to set the **Sample Clock Source** to either Internal Clock or S/PDIF sync. To prevent drift, Muse Research strongly recommends that you lock Receptor to your host computer using a S/PDIF cable and S/PDIF sync.
- **Tempo Source** is set to **External - UniWire**, but you may choose to use the **Internal** tempo or any MIDI clock that appears on Receptor's MIDI In jack.

Caveat: *Keep in mind that, in this mode, it's possible to send MIDI and/or audio to/from Receptor via both UniWire and standard audio/midi cables. Careless sequencer routing could result in doubled audio or MIDI signals being sent to/from Receptor, so consider your routing carefully when turning ON the* ***Enable Receptor Audio+MIDI I/O*** *option.*

Bypass MIDI Filter on Single Channels

There are two ways that UniWire can address Receptor:

- on ALL Receptor channels: Using this method, UniWire communicates with Receptor exactly the same as if you had connected a pair of MIDI cables between your host computer's MIDI interface and Receptor. Your host computer sends 16 MIDI channels to Receptor and Receptor responds to those MIDI channels using the various MIDI Filter settings provided on each Instrument Channel.
- on SINGLE, individual Receptor channels: Using this method, UniWire sends 16 MIDI channels to a ***single*** Receptor Instrument Channel. You can instantiate multiple UniWire instances and send 16 MIDI channel to each and every Receptor Instrument channel or even to each plugin within a single channel (effectively making Receptor a 92 MIDI port device capable of responding to 1472 MIDI Channels)!

The setting in which you determine which Receptor channels to address via UniWire is contained within the UniWire plugin, discussed later, but the Bypass MIDI Filter on Single Channels parameter determines how Receptor responds to MIDI data arriving over UniWire.

Obviously, when you communicate with Receptor on single, individual channels (as described above), each Receptor Instrument Channel is capable of operating as a complete 16-channel multitimbral synthesizer. Normally, this operation would require that you reconfigure the MIDI Filter on each Receptor instrument channel by opening it and setting the "Listen to" channel to "ALL" and the "Map to" channel to "THRU".

But there's an easier way: If you check the **Bypass MIDI Filter** option, then all of Receptor's MIDI Filters are bypassed and all 16 MIDI channels are automatically able to play all 16 MIDI channels on each Receptor Instrument Channel.

By checking the **Bypass MIDI Filter** option, Receptor works exactly like a plugin running on your host computer. That is, you're able to use your host application to route different tracks to different plugins on different MIDI channels and Receptor's built-in MIDI Filters won't work against you.

IMPORTANT: *The Bypass MIDI Filter option works* ***only*** *when UniWire is configured to send MIDI to a single Receptor channel, rather than ALL channels. When a UniWire plugin sends MIDI to ALL Receptor channels, then Receptor's MIDI Filtering and routing capabilities are still active, regardless of whether or not this option is enabled. This insures that a UniWire connection works exactly like a connection involving traditional MIDI and audio cabling. Muse Research recommends that you leave this option checked unless you're configuring some advanced MIDI routings and are comfortable handling the possible routing interactions between your host sequencer and Receptor.*

Front Panel Operation: *On Receptor's front panel, this parameter is called "UniWire MIDI Filter." To enable/disable the filter, press the SETUP button on Receptor's front panel and turn the top display knob until* `UniWire MIDI Filter` *is displayed. Turn the bottom display knob to select between* `Bypassed` *and* `Not Bypassed`.

Master Sample Rate

When UniWire is enabled, this parameter is automatically set to the sample rate used by your host computer. No other options are available and, as such, the parameter is disabled.

Sample Clock Source

If UniWire is enabled and the **Enable Receptor Audio+MIDI I/O** option is OFF, then Receptor automatically sets this parameter to **External (UniWire)** and disables all other options. This is because, in this mode of operation, Receptor's own audio input/output circuitry is disabled and its audio clock is slaved to that of your host sequencer (which is transmitted over UniWire).

If UniWire is enabled and the **Enable Receptor Audio+MIDI I/O** option is OFF, then Receptor's internal audio may sync either to itself (**Internal**) or to a **S/PDIF** input (if one it used). It's important to remember that, if you're using both Receptor's internal clock and your host computer's clock, they will not be in sync. For this

reason, Muse Research strongly recommends that customers who turn ON the **Enable Receptor Audio+MIDI I/O** also connect a S/PDIF cable between Receptor and their host computer and set Receptor's **Sample Clock Source** to **S/PDIF Sync Enabled**.

Sample Buffer Size

If UniWire is enabled, Receptor automatically sets its **Sample Buffer Size** to match the value defined by your host sequencer. No other options are available since the two buffer sizes must match exactly. For this reason, the **Sample Buffer Size** parameter is disabled when Enable UniWire is turned on. If you wish to have a larger or smaller buffer size, set the value in your host sequencer and Receptor will modify its setting to match.

Tempo Source, Tempo, and Time Signature

A third choice, called **External - UniWire**, has been added to the **Tempo Source** selector. This option is available only if **Enable UniWire** has been turned on. When you select **External - UniWire**, Receptor will sync to the host sequencer's tempo that is transmitted over UniWire. This insures that any tempo-based effects contained within Receptor stay synchronized to your host sequencer's tempo.

If **Tempo Source** is set to **External - UniWire**, then the **Tempo** field becomes disabled and displays the tempo generated by you host sequencer. Similarly, the **Time Signature** field also becomes disabled and displays the time signature generated by your host sequencer.

If **Enable Receptor Audio+MIDI I/O** is OFF, then Receptor's MIDI I/O is disabled and the **External - MIDI** option is, consequently, also disabled in the **Tempo Source** menu. If, however, the **Enable Receptor Audio+MIDI I/O** option is ON, then Receptor's MIDI I/O is active and the **External - MIDI** option becomes available as a tempo source for Receptor.

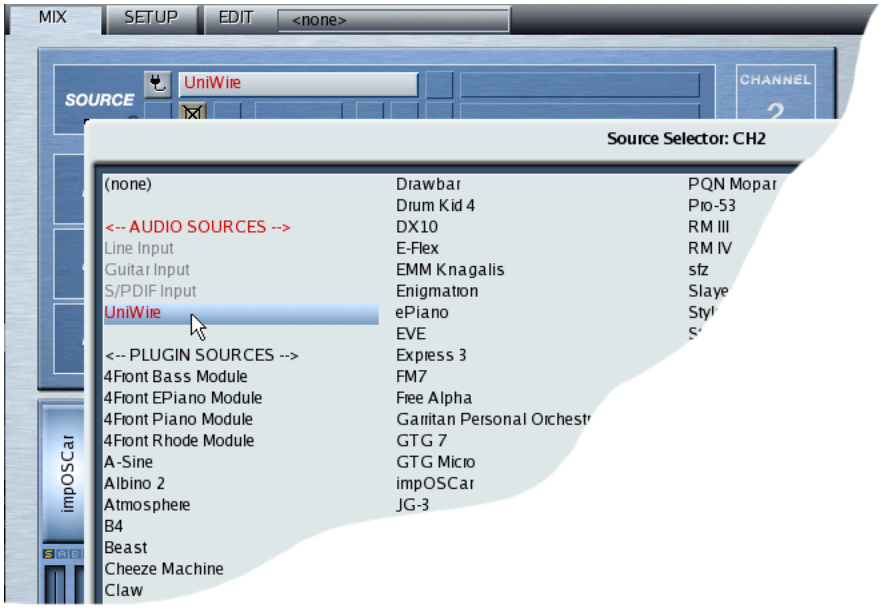
UniWire Audio Source

The Source Selector on each Receptor channel features a new Audio Source option, called "UniWire."

You can select "Line Input" on several Receptor channels and you will get the same audio on every channel. With UniWire every input is unique. If you select UniWire on channel 1 you have to send audio to channel 1 from your DAW.

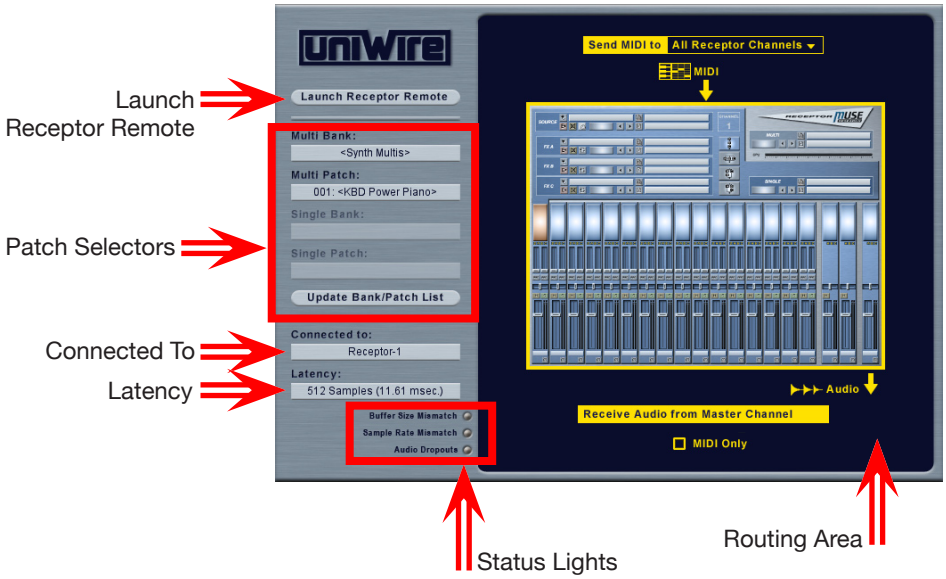
There is no main audio input for UniWire. So when you select "All MIDI Channels" on an effect, you are sending MIDI to the entire Receptor and receiving audio from the Master, but you cannot send any audio.

Then, why even have a UniWire effect that cannot send audio? You can insert this in your Input channel and send MIDI to Receptor and record the output of Receptor's Master channel. The UniWire instrument, which does the same (sends MIDI and receives audio), cannot be inserted into an audio channel, so you need the FX to do that.



UniWire—Plugin Details

In general, the UniWire Instrument plugin interface looks as follows, though variations in appearance exist depending on various parameter settings.



The following sections discuss these parameters in detail.

Connected To

The menu lists every Receptor currently connected to your network. UniWire will communicate with the Receptor you select from the **Connected To** menu.

By default, UniWire communicates with the first Receptor in the list. So, if you have only one Receptor, you can basically ignore this menu. If you have multiple Receptors, then you'll use this menu to choose which of your Receptors to communicate with.

Details about how your host sequencer saves information about which Receptor to connect to: *When you save a sequence in your host application, the "Connected to" Receptor is remembered by both its name and IP address. So, when you open a file on your host sequencer, UniWire will first try to communicate with whichever Receptor has the same name as the one saved with the sequence. If you've changed the name of your Receptor, your host sequencer will then try to connect to whichever Receptor has the same IP address as the one saved with the sequence. If you also changed your Receptor's IP address, then your host sequencer will not know which of your multiple Receptors you wish to communicate with, and the* ***Connected to*** *menu will display "No Selection." Use the menu to then select the desired Receptor.*

Latency

Latency is the amount of time it takes to send a MIDI note or audio recording over UniWire to Receptor, process it in Receptor, then send it back over UniWire to your host computer. When you use UniWire, your host sequencer and Receptor must be set to identical buffer sizes. For example, if your host sequencer uses a 128 buffer sample, then you must also set Receptor's buffer (located in Setup View) to 128 samples. Since your host computer and Receptor each have their own buffers, the minimum possible UniWire latency is two times their common buffer size. So, for example, if your host sequencer and Receptor both have their buffers set to 64 samples, then the minimum possible UniWire latency is 128 samples (2 x 64), which is equivalent to 2.9ms, assuming a 44.1kHz sample rate.

This fact is reflected in the **Latency** menu. It displays all of UniWire's possible latency values between 2x and 16x the sample buffer size. It automatically calculates and displays these latency values in both *samples* (calculated from the chosen buffer size) and in *ms* (calculated from the sample rate).

In general, if you're tracking in real time (for example, performing audio processing on a live audio input or playing a MIDI synth in real-time), then you'll want to set the **Latency** (and the corresponding buffers) as low as possible while still achieving sonically acceptable output. If you're mixing down a track and not processing any live inputs, you can set higher latencies for better performance, CPU usage, and track count.

Normally, when you run a soft synth (of effect) on your host computer, its latency is equal to the buffer size you select in your host sequencer. For example, if you set your host sequencer's buffer to 128 samples, then you will hear a note 128 samples after requesting it.

Latency Compensation

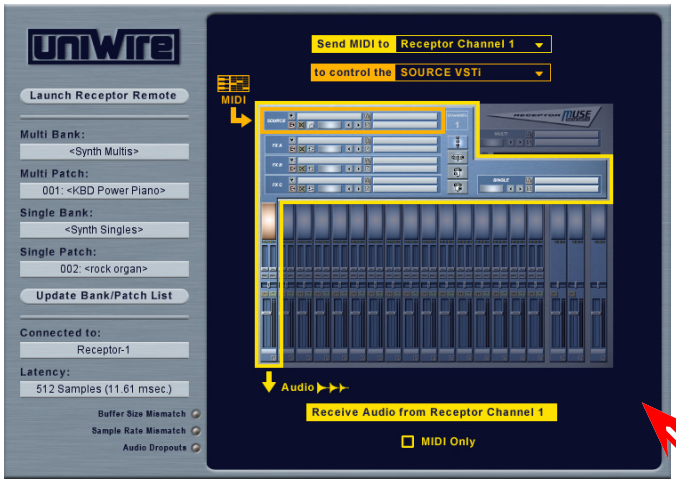
Most modern sequencers now feature automatic plugin delay compensation, meaning the sequencer will automatically shift tracks in time (to adjust for various latencies), so that they playback in sync.

If your host sequencer does not provide plugin delay compensation, then you need to delay any non-UniWired tracks by the number of samples specified in UniWire's Latency menu. For example, if you have an audio track and a UniWired MIDI track and UniWire is set to 256 samples of latency, then you need to delay your audio track by 256 samples, so that the tracks align. See your host sequencer's documentation to learn about its 'delay compensation' features.

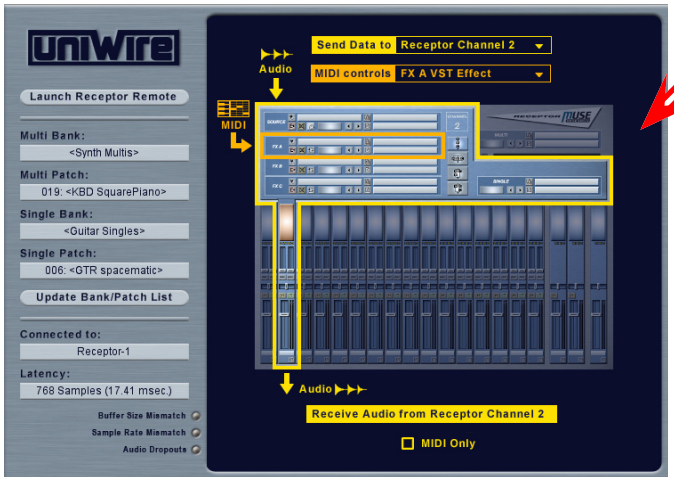
Routing Area

Use this area to define how audio and midi data flows between your host application and Receptor.

UniWire Instrument



UniWire FX



Routing Area

‘Send MIDI To’ Menu
‘Send Data To’ Menu

The name of this menu changes depending on whether you’ve instantiated a UniWire Instrument or a UniWire FX, but the function is basically identical. Use this menu to tell UniWire where, in Receptor, to send the data from your host sequencer. If you’re using a UniWire Instrument, the menu is called “Send MIDI To” since only MIDI data is transmitted from your host sequencer to your Receptor. If you’re using a UniWire FX, the menu is called “Send Data To” since you are sending both MIDI and Audio data from your host application to Receptor. You choices are:

- **All Receptor Channels**

  This is the ‘standard’ method for communicating with Receptor. When you choose to send data to **ALL** Receptor channels, then UniWire communicates with Receptor exactly the same as if you had connected a pair of MIDI cables between your host computer’s MIDI interface and Receptor. Your host computer sends 16 MIDI channels to Receptor and Receptor responds to those MIDI channels using the various MIDI Filter settings provided on each Instrument Channel. Audio from Receptor’s Master audio output is returned to your host application via UniWire, exactly as if you had connected audio cables between Receptor’s stereo output to your host computer’s audio interface.

- **Receptor Channel 1-16, Buss 1-2, Master**

  When you select one of these individual channel options, UniWire sends all 16 MIDI channels to the ***single*** Receptor Channel you select and, if you're using a UniWire FX, it also determines which Receptor channel receives that audio data. Furthermore, when you select one of these individual channel options, UniWire displays an additional Control menu immediately below the **Send MIDI To / Send Data To** menu. Use this additional menu to tell Receptor exactly where in that channel to send those 16 MIDI channels. This effectively makes Receptor a 92 MIDI port, 1472 MIDI channel device! The operation of the Control menu is discussed further in the next section.

  When UniWire communicates with Receptor on a single channel, audio is sent directly out of the selected Receptor Channel back into your host sequencer. This means, if you select Instrument Channels 1-16, that Receptor's two effects busses and the master channel are removed from the audio signal chain, so any plugins instantiated on those channels (or bussing to those channels) will have no effect (unless you instantiate additional UniWire plugins and route these channels back in to your host sequencer).

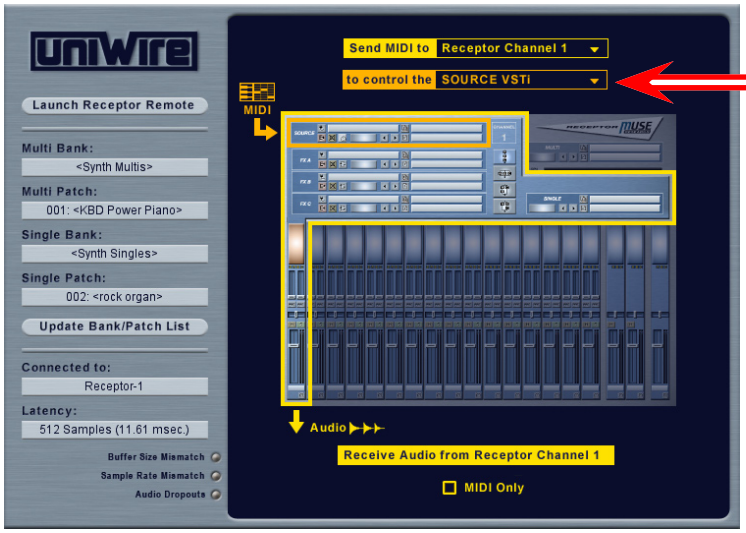
  Finally, when you use choose to communicate with Receptor on a single Channel, that channel's MIDI filter is bypassed by default. This allows your host application to control where all MIDI data is sent and insures that the selected channel responds to all 16 MIDI channels, regardless of the state of Receptor's Instrument Channel MIDI filter 'Listen To' and 'Play Plugin' settings. You may disable the MIDI filter bypass by un-checking the **Bypass MIDI Filter** option in Receptor's Setup view (or under the SETUP button on Receptor's front panel). If you do this, keep in mind that you must then set the Receptor channel's 'Listen To' and 'Play Plugin' settings manually, so as to match those in your host sequencer. Also, if you turn off the MIDI filter bypass and want to use a multitimbral plugin, you must set 'Listen To' to **All** and 'Play Plugin' to **Thru**. For more information about Receptor's Bypass MIDI Filter option, see its discussion earlier in this manual.

The Routing Area always displays a tiny representation of the Receptor User Interface to illustrate exactly what's happening. Notice that when you set **Send˛MIDI˛To** to any of the individual channel options, the graphic updates to show exactly what's being accessed via UniWire—where data goes into Receptor (over UniWire) and where the audio comes out (over UniWire). Similarly, when you set **Send˛MIDI˛To** to **All˛Receptor˛Channels**, the graphic illustrates that MIDI controls every Receptor channel and that audio comes from the Master Channel.

SHORTCUT: *You can click in the tiny virtual Receptor graphic to make changes without using the* ***Send˛MIDI˛To*** *menu. Specifically, click any channel in the graphic and the* ***Send˛MIDI˛To*** *value changes accordingly. If you click the "multi patch" area of the tiny virtual Receptor graphic (upper right corner of graphic), then the* ***Send˛MIDI˛To*** *parameter is set to* ***All˛Receptor˛Channels****.*

‘To Control The’ Menu
‘MIDI Controls’ Menu

This drop-down menu appears only if you address Receptor on an individual channel (meaning you select something other than ‘**All**’ from the **Send MIDI To / Send Data To** menu).



Note that the name of this menu varies slightly depending on whether you’re using a UniWire Instrument or an FX. If you’re using an instrument, the parameter is called “To control the.” If you’re using an FX, the parameter is called “MIDI Controls.” The functions, however, are nearly identical.

When you send data to an individual Receptor channel (as specified in the **Send MIDI to / Send Data to** menu), you must also specify where, in that channel, to send the MIDI data. Use the **To Control The / MIDI controls** menu to tell UniWire exactly which slot in the selected channel should receive and process the MIDI data. Specifically, you can send MIDI to:

- **SOURCE VSTi**

  This is the default selection for the UniWire Instrument plugin. It routes all MIDI to the plugin assigned to the selected channel’s Source slot.

- **FX A VST Effect**

  This is the default selection for the UniWire FX plugin. This selection routes all MIDI to the plugin assigned to the selected channel’s FXA slot. If the **Send MIDI To** parameter is set to either Buss 1, Buss 2, or Master, then this also becomes the default value (since these channels have no Source modules).

- **FX B VST Effect**

  This selection routes all MIDI to the plugin assigned to the selected channel’s FXB slot.

- **FX C VST Effect**

  This selection routes all MIDI to the plugin assigned to the selected channel’s FXC slot.

- **MIX Section**

  This selection routes all MIDI to the selected channel’s various MIX parameters.

SHORTCUT: *Drag the little MIDI Destination indicator (which appears to the left of the virtual Receptor graphic) up and down to change the MIDI control destination without using the drop-down menu.*

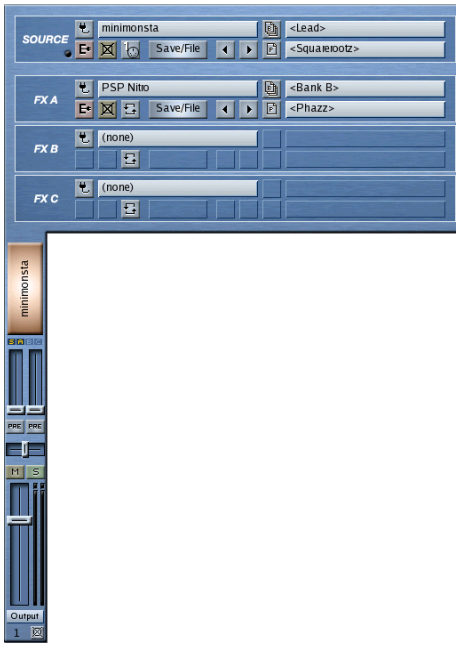
By default, all MIDI data is sent to the plugin assigned to Receptor’s **Source** slot. In most cases, this will be what you want, as all MIDI notes and controller data will be sent to the Instrument plugin assigned to the Source slot. If, however, you wish to route MIDI to control one or more of the insert effects, or if you wish to control Receptor’s own MIX parameters from your sequencer, then the other options exist.

NOTE: *Since each instantiation of UniWire is capable of sending 16 MIDI channels to Receptor and each instantiation can be routed to a specific plugin or mixer channel, this means Receptor is essentially a 92 MIDI port device (16 instruments channels w/ 5 MIDI destinations each + 2 effects channels with 4 MIDI destinations each + 1 Master channel with 4 MIDI destinations = 92 MIDI ports). And, since each virtual MIDI port can receive 16 MIDI channels each, Receptor is capable of responding to 1472 different MIDI channels.*

MIDI Only Option

Check this option to prevent the UniWire instance from transmitting audio data over UniWire. This is particularly useful if, for example, you're using a UniWire instance to control a specific plugin's MIDI parameters and another UniWire instance is already being used to transmit audio processed by that plugin.

For example, assume that you have the *Minimonsta* instrument assigned to Receptor CH 1's Source slot, the *PSP Nitro* effect assigned to the FXA slot, and you want to control both plugins from your host sequencer via UniWire.



To do this:

1 In your host sequencer, create two new virtual instrument tracks: Track 1 will be used to control the Minimonsta plugin assigned to Receptor's CH 1/Source slot and Track 2 will be used to control the PSP Nitro plugin assigned to Receptor's CH 1/FXA slot.

2 In Track 1 of your host sequencer, instantiate a UniWire plugin. Set the plugin's **Send MIDI To** parameter to '**Receptor Channel 1**' and the **To Control The** parameter to '**SOURCE VSTi**.'

All MIDI data contained in sequencer Track 1 (both notes and MIDI CC) will control the Minimonsta plugin on Receptor Channel 1 and audio will be routed out of Receptor CH 1 back into your host sequencer.

3 In Track 2 of your host sequencer, create a new instantiation of the UniWire plugin (do not use the same instantiation you created to control the Minimonsta). Set this plugin's **Send MIDI To** parameter to '**Receptor Channel 1**' and the **To Control The** parameter to '**FX A VST Effect**.'

4 Enable (check) the **MIDI Only** option.

All MIDI data contained in sequencer Track 2 will control the PSP Nitro plugin on Receptor's CH 1/FXA but no redundant audio from Receptor CH 1 will be returned to the host sequencer.

Using this method, you can create as many UniWire instantiations as you need to control as many Receptor-hosted plugins or Receptor Mix parameters as you want.

Status Lights

These are 'warning' lights that let you know if there are any problems with UniWire communication between your host computer and Receptor.

Buffer Size Mismatch

When using UniWire, Receptor and your host sequencer must have exactly the same plugin buffer sizes. As such, Receptor will automatically set itself to the buffer size used by your host sequencer. If, however, your host sequencer is set to a buffer size not supported by Receptor, then the **Buffer Size Mismatch** LED will light.

If the Buffer Size Mismatch LED lights, you'll need to set your host sequencer to a buffer size supported by Receptor.

Sample Size Mismatch

When using UniWire, Receptor and your host sequencer must have exactly the same sample rates. As such, Receptor will automatically set itself to the sample rate used by your host sequencer. If, however, your host sequencer is set to a sample rate not supported by Receptor, then the **Sample Size Mismatch** LED will light.

If the Sample Size Mismatch LED lights, you'll need to set your host sequencer to a sample rate supported by Receptor.

Audio Dropouts

The **Audio Dropouts** LED will light whenever Receptor can't process all the data fast enough and drops an audio buffer. Many things affect whether or not you get audio dropouts, including:

- Sample Buffer Size (as set in both Receptor and your host sequencer) -- lower values result in less latency, but more likelihood of audio dropouts.
- UniWire Plugin's Latency setting -- this is a multiplier of the Sample Buffer Size setting. The minimum possible latency is 2x the Sample Buffer Size. Lower values result in less latency, but more likelihood of audio dropouts.
- Amount of data being sent over UniWire to/from Receptor -- the more data you try to process within a given buffer size, the greater the chance of audio dropouts.

In general, if the Audio Dropouts LED lights, you should:

- Increase the latency (either by using the Latency setting in UniWire or by setting a larger Sample Buffers Size in your host sequencer), or...
- Record some of the tracks currently being processed by Receptor, freeing the unit to perform the necessary real-time processing, or...
- Add additional Receptors to your UniWire network and distribute processing across multiple Receptors.

Once the Audio Dropouts LED lights, you simply click it to reset it.

Patch Selectors

You can use the UniWire plugin to remotely select Receptor Multi patches. And, if UniWire is configured to address individual channels (that is, the **Send MIDI To** menu is set to anything other than "ALL"), then you may also remotely select Receptor Single patches.

If you add, delete, rename, or modify any patches within Receptor, you must click the **Update Bank/Patch List** button in order to see those changes reflected in the UniWire window.

NOTE: *When you select Multi and Single patches from Receptor itself (using its Graphical User Interface or its front panel), these changes are not automatically reflected in the UniWire plugin's Patch Selector menus. Receptor will never volunteer this information so, if some patches change in Receptor, you should press the Update Bank/Patch List button to tell the plugin that something has changed. There are some plugin actions that will automatically update the Patch lists. For example, the UniWire plugin will ask the Receptor for a list of Multis and Singles when a new Receptor is 'connected to' or when a new UniWire plugin is instantiated. Also, the plugin will ask which Single is current whenever you change the channel.*

Important: *When you save a file in your host sequencer, it will remember only which patches were used by Receptor -- it will* ***NOT*** *remember any unsaved edits contained within Receptor. That is, if you modify any Receptor patches, you must save those edits as a Receptor Single or Multi patch if you want them to be properly recalled when you open the sequence file in your host application.*

Note that a single sequence might contain numerous UniWire instantiations and that each UniWire instance is capable of remembering its own patch assignments. If you recall a sequence file that uses multiple UniWire instances, Receptor will respond first to any Multi patch requests then to any Single patch requests.

Launch Receptor Remote

Click this button to launch the Receptor Remote application. Opening Receptor Remote allows you to graphically edit Receptor over the same Ethernet cable that UniWire uses.

Using Receptor as an Audio Interface over UniWire

Throughout this document, we have mostly discussed how to use Receptor in the following configuration:

- Your host computer is used for all audio and MIDI connections and Receptor, connected via UniWire, becomes a 'slave' to your host sequencer -- essentially acting like an external 'DSP farm' for your host computer. In this mode, Receptor's built-in audio and midi connections are disabled.

It's possible, however, to use UniWire along with Receptor's own audio and MIDI connections in one of the following two ways:

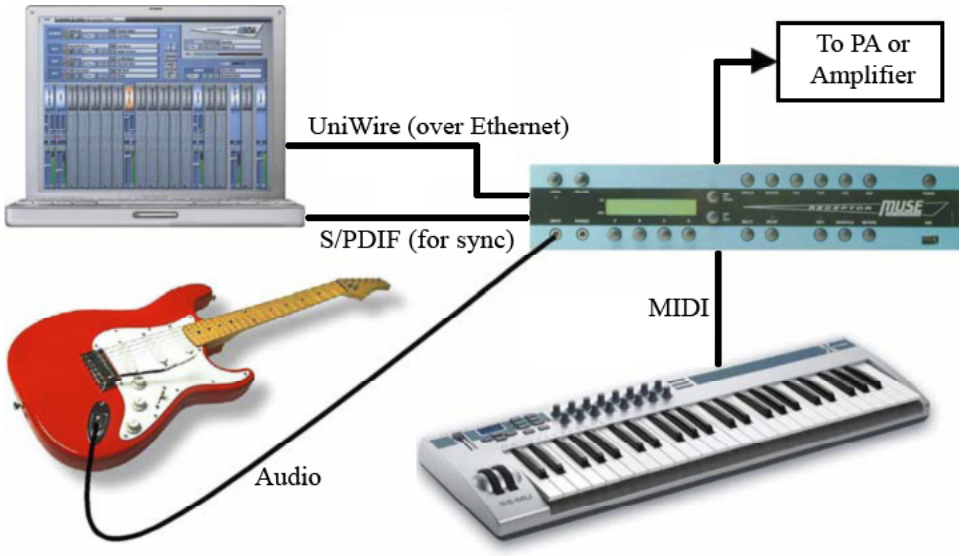
1) Receptor is used for *all* audio connections and no audio interfaces are used on your computer. In this mode, you have no audio interfaces connected to your host computer and, instead, use Receptor's audio connections. This mode can be particularly useful for laptop users who play live and wish to simplify their rig by using Receptor's high fidelity audio circuity rather than carrying an additional audio interface.
2) *Both* Receptor's audio and MIDI connections *and* your computer's audio and MIDI connections are used simultaneously. This mode is the most flexible but complex connection.

In both scenarios, Muse Research strongly recommends that you also route a S/PDIF cable between Receptor and your computer in order to lock their digital clocks together and achieve sample accurate sync.

The following sections will discuss these two methods in more detail.

Receptor Replaces Your Computer's Audio Interface

In this scenario, Receptor is used for *all* audio connections and no audio interfaces are used on your computer.



The following are basic instructions. See your host sequencer's documentation to learn, specifically, how to route audio/midi data and instantiate/edit plugins:

1 Connect audio and MIDI cables to Receptor.

2 Connect a crossover Ethernet cable between Receptor and your host computer or, if using multiple Ethernet devices, connect a standard Ethernet cable between Receptor and your host computer's Ethernet Router/Switcher.

3 If possible, connect a S/PDIF cable between Receptor and your host computer. Most audio cards and some computers have built-in S/PDIF connectors. Synchronize the two devices either by 1) setting Receptor's **Sample Clock Source** to use S/PDIF Sync (meaning your computer is the master clock), or 2) setting your computer to sync to incoming S/PDIF from Receptor. Without synchronization, you may experience occasional glitches during audio playback since, even though you're using Receptor for all your audio ins and outs, your host sequencer is still getting its timing from the computer's built-in audio.

3 On Receptor, make sure UniWire mode is turned on.

On the front panel, press the SETUP button, rotate the top display knob to select the `UniWire` *option, then rotate the bottom display knob to select* `Enabled`.

On the graphical interface, press the **Setup** *Tab and, in the UniWire section, check the* **Enable UniWire** *option.*

4 Enable Receptor's built-in Audio + MIDI I/O.

On the front panel, press the SETUP button, rotate the top display knob to select the `UniWire Receptr I/O` *option, then rotate the bottom display knob to select* `Audio+MIDI Enabled`.

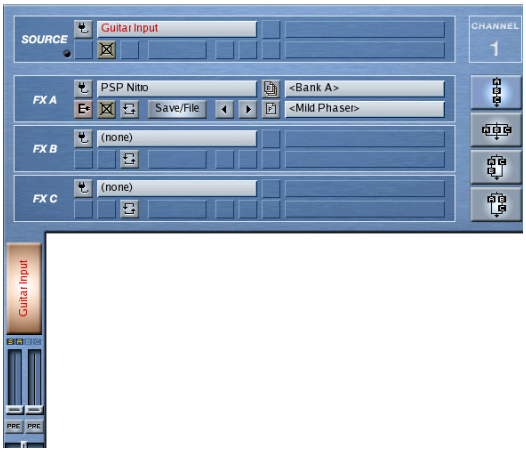
On the graphical interface, Press the **Setup** *Tab and, in the UniWire section, check the* **Enable Receptor Audio+MIDI I/O** *option.*

Note that Receptor does not output MIDI data over UniWire. It *receives* MIDI data from your host computer and it both sends and receives *audio* data, as discussed in the following sections:

To send audio from Receptor to your host sequencer over UniWire

If you want to send any audio received by Receptor to your host sequencer (for recording or for additional effects processing by your host sequencer), you would do the following:

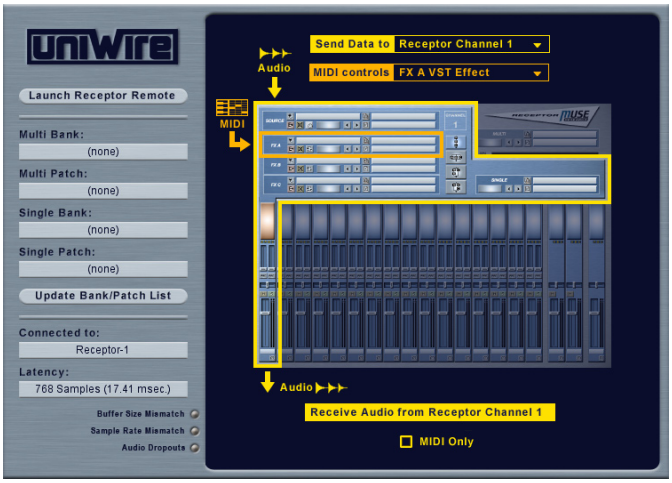
1 In Receptor, configure a channel with the Source set to the desired audio input.



2 In your host sequencer, instantiate a UniWire FX plugin on an audio track.

3 Open the edit window for that UniWire FX plugin, and tell it to **Send Data to** the same Receptor channel you just configured.

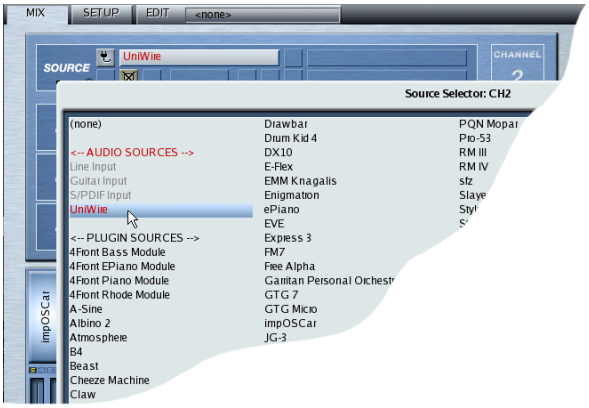
This will cause UniWire to send audio from that Receptor channel over UniWire and into your host sequencer.



To send audio from your host sequencer to Receptor over UniWire

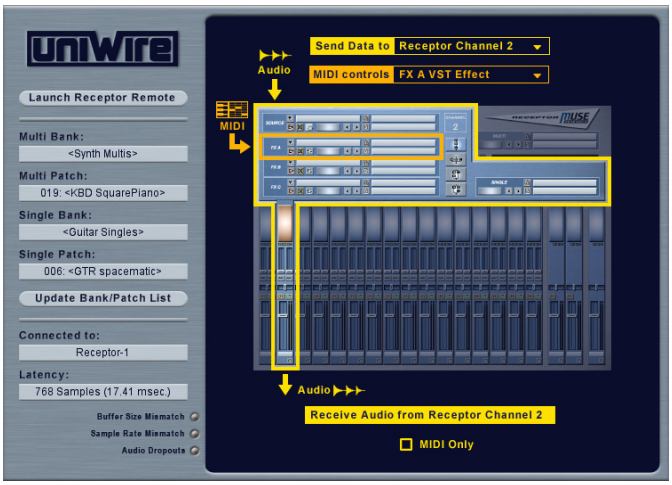
If your sequence contains some backing tracks that you wish to output through Receptor or, if you've processed some Receptor audio through your host sequencer and wish to return it to Receptor, then you'll need to do the following:

1 On Receptor, configure a channel with the Source set to use the UniWire audio input.



2 In your host sequencer, instantiate a UniWire FX on the Audio track that contains the data you want to send to Receptor.

3 Open the editor for that UniWire FX and tell it to **Send data to** the Receptor channel you just configured to listen to the UniWire audio input.



To send MIDI from your host sequencer to Receptor over UniWire:

1 On Receptor, configure those channel(s) on which you want to receive and process the incoming MIDI.

For example, if you want the data to play the Minimonsta plugin on Receptor Channel 1, then choose **Minimonsta** *from that channel's Source selector.*

2 In your host sequencer, instantiate a UniWire Instrument on the MIDI track that contains the data you want to send to Receptor.

3 Open the editor for that UniWire Instrument and tell it where, on Receptor, you want to route this MIDI data.

Following through with the previous example, this would be Receptor Channel 1, since it contains the Minimonsta instantiation you wish to play.

Audio/MIDI from Receptor and Your Computer Are Used Simultaneously

In this scenario, *both* Receptor's audio and MIDI connections *and* your computer's audio and MIDI connections are used simultaneously. To do this, you must first turn on UniWire and enable Receptor to use its own Audio+MIDI ports in conjunction with UniWire:

1 Connect audio and MIDI cables to Receptor and to your host computer's audio/midi interfaces.

2 Connect a crossover Ethernet cable between Receptor and your host computer or, if using multiple Ethernet devices, connect a standard Ethernet cable between Receptor and your host computer's Ethernet Router/Switcher.

3 On Receptor, make sure UniWire mode is turned on.

On the front panel, press the SETUP button, rotate the top display knob to select the `UniWire` *option, then rotate the bottom display knob to select* `Enabled`.

On the graphical interface, press the **Setup** *Tab and, in the UniWire section, check the* **Enable UniWire** *option.*

4 Enable Receptor's built-in Audio + MIDI I/O.

On the front panel, press the SETUP button, rotate the top display knob to select the `UniWire Receptr I/O` *option, then rotate the bottom display knob to select* `Audio+MIDI Enabled`.

On the graphical interface, Press the **Setup** *Tab and, in the UniWire section, check the* **Enable Receptor Audio+MIDI I/O** *option.*

IMPORTANT: *Since you wish to use* **both** *Receptor's audio circuitry* **and** *your computer's, you essentially have two different audio interfaces connected to your computer. As such, you must synchronize them by connecting a S/PDIF cable between Receptor and your computer's audio interface.*

To synchronize Receptor's audio to your host computer's audio interface:

1 Connect a cable from the S/PDIF OUT port on your computer's audio interface to the S/PDIF IN port on Receptor.

2 Set Receptor's **Sample Clock Source** to use S/PDIF Sync.

To do this from the front panel, press the SETUP button, then rotate the top display knob to select the **Digtl Clock Source** *parameter. Rotate the bottom display knob to select* **S/PDIF**.

To do this from the graphical editor, click the Setup tab and, in the Audio section of Setup view, set the **Sample Clock Source** *to* **S/PDIF Sync Enabled**.

Once the two audio interfaces are synchronized over S/PDIF, you can simultaneously use Audio+MIDI on both Receptor and your computer.

Note that, if you prefer, you may also use Receptor as the master clock source and slave your computer to it.

Once you have synchronized the two audio interfaces, you're free to route data back and forth between Receptor and your computer using any of the techniques discussed in this document. Keep in mind that, because MIDI and audio are active on both devices, it's easy to create doubled audio, infinite loops, and other undesirable situations without careful consideration of your signal routings.

Host-Specific Examples

The following sections discuss, on a host-by-host basis, how to communicate with Receptor via UniWire. Each example assumes you are familiar with your host sequencer and how to use it with native instrument and effect plugins.

Using UniWire with Cubase SX or Nuendo

This section assumes you know how to instantiate, play and record virtual instruments and effects within Cubase SX. If not, please consult your Cubase manual and make sure Cubase SX is working properly with host-based plugins before you use UniWire.

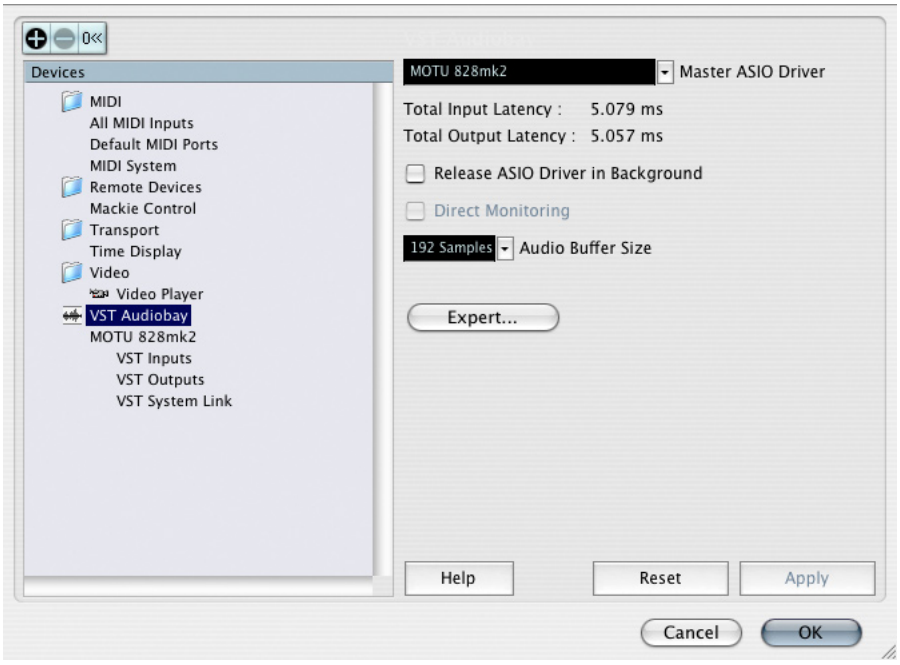
NOTE: *Since Cubase SX and Nuendo share common operating procedures, the methods discussed in this section apply to both Cubase and Nuendo.*

Disable Multi Processing option in Cubase

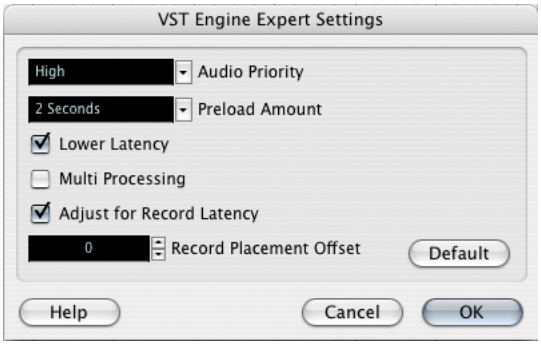
IMPORTANT!: *If you're using a multiple processor computer with Cubase SX then, as of this writing, UniWire works much more efficiently within Cubase if its multi processor support is turned OFF. To do this:*

1 In Cubase, select **Devices>Device Setup**.

2 In the Device Setup dialog, click the **VST Audiobay** options (in the left **Devices** column) to show the related options in the right-half of the dialog box



3 Click the **Expert** button to open the VST Engine Expert Settings dialog.



4 Disable the **Multi Processing** option and click **OK**.

Cubase Uses Receptor as an External UniWire Instrument

The following describes how to use UniWire to make Receptor act as an external instrument within Cubase SX3:

1 Before beginning, make sure UniWire is enabled on Receptor.

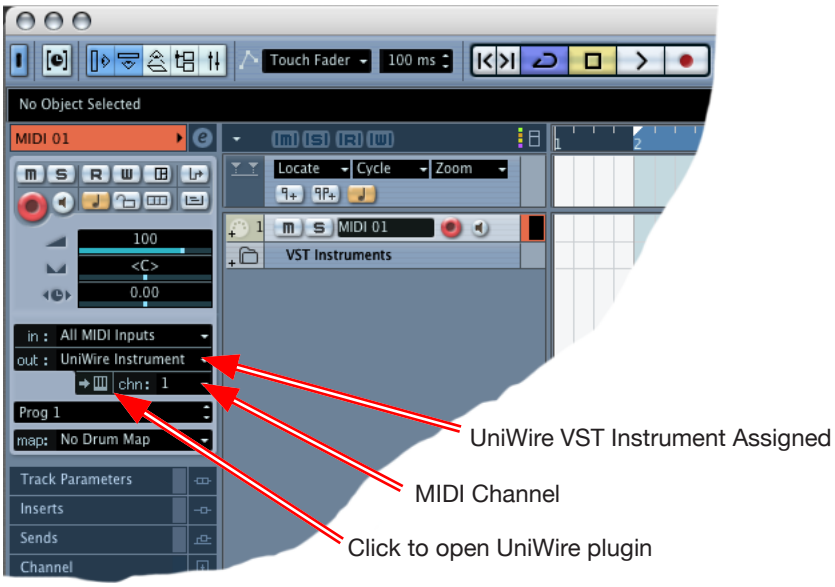
To do this from Receptor's graphical interface, click the SETUP tab and check the ***Enable UniWire*** *option in the new UNIWIRE section in Setup View.*

To do this from Receptor's front panel, press the SETUP button and rotate the top display knob until the top line shows the "UniWire" option, then rotate the bottom knob to select `Enabled`.

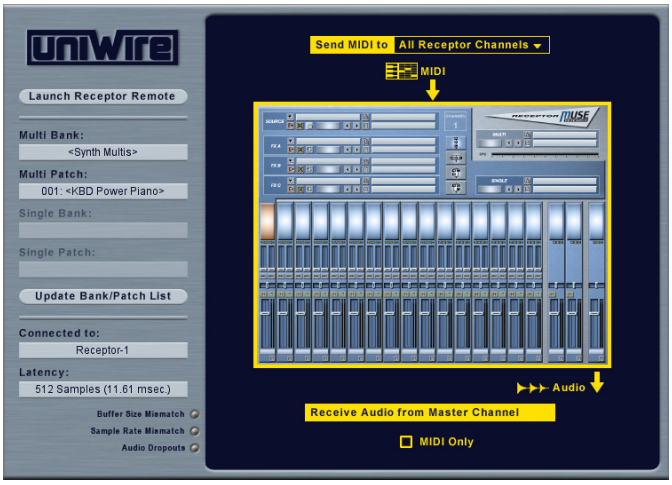
2 In Cubase, create a new MIDI Track by choosing **Project>Add Track>MIDI**.

3 Choose **Devices>VST Instrument** and, in a blank instrument selector within the resulting window, choose **UniWire Instrument** from the **Muse** folder.

4 In Cubase's Track Inspector, assign the desired MIDI Input to this track; assign the previously instantiated UniWire Instrument plugin as the MIDI out, and set a MIDI channel.



5 Click the "Open Device" icon (to the left of the MIDI channel setting) to open the graphical editor for the UniWire plugin.



6 From the **Connected To** menu, select which Receptor you wish to use.

*If you have only one Receptor, it will appear automatically in the **Connected To** menu. If you have multiple Receptors, the first will appear, but you can select a different one in the **Connected To** menu.*

7 Set the desired **Latency**. This setting is global across all UniWire instances.

8 Configure your UniWire and Receptor options as previously discussed. Some options include:

- *If you want to use an existing Multi patch and treat Receptor as if it were connected to your host computer via MIDI and audio cables, simply keep the **Send MIDI to** menu set to **All Receptor Channels** and select the desired Multi patch using UniWire's **Multi Bank** and **Multi Patch** selectors.*
- *If you want to communicate with each Receptor channel as if were its own external sound module, use the **Send MIDI to** menu to select which Receptor channel to communicate with. You will then have access to the **Single Bank** and **Single Patch** fields should you wish to select existing Receptor Singles via UniWire.*
- *If you want to configure Receptor manually, simply click the **Launch Receptor Remote** button and Receptor's graphical User Interface will appear in another window. Make sure, when you edit Receptor in this way, to save your edits to either a Receptor Single or Multi patch, so that they are automatically recalled by your host sequencer.*

Detailed discussions of every UniWire parameter were discussed earlier in this document.

9 Play a MIDI keyboard connected to Cubase and, if the keyboard is selected in Cubase's **MIDI Input** field and the channel is being monitored, then you should hear Receptor just as if you had connected it to your computer using audio and MIDI cables.

Cubase Uses Receptor as an External UniWire Effect

The following describes how to use UniWire to make Receptor act as an external effect within Cubase¸SX3:

1 Before beginning, make sure UniWire is enabled on Receptor.

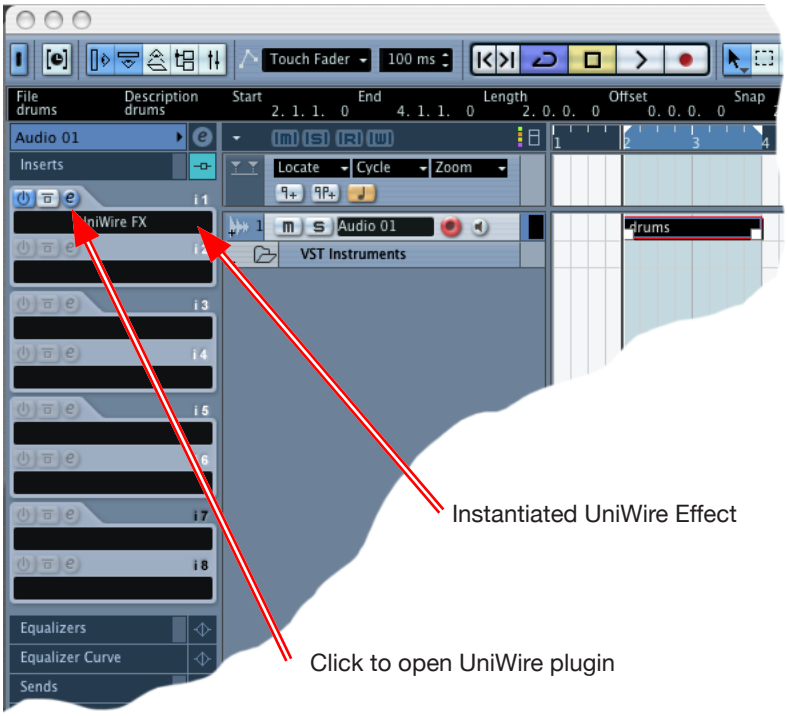
*To do this from Receptor's graphical interface, click the SETUP tab and check the **Enable UniWire** option in the new UNIWIRE section in Setup View.*

To do this from Receptor's front panel, press the SETUP button and rotate the top display knob until the top line shows the "UniWire" option, then rotate the bottom knob to select `Enabled`.

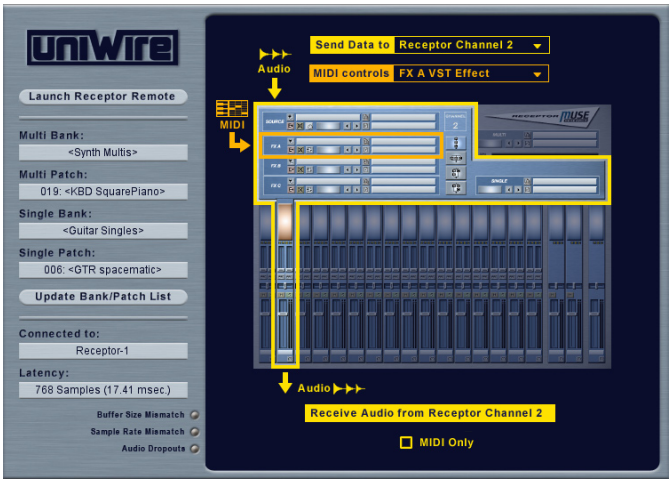
2 In Cubase, open the Project Window and select the audio track you wish to process with Receptor.

3 Click the **Inserts** tab in the Track Inspector (located in the left column in Cubase's Project Window).

4 Click a blank plugin field and choose **UniWire FX** from the **Muse** folder.



5 Click the plugin's EDIT button to open the graphical editor for the UniWire plugin.



6 From the **Connected To** menu, select which Receptor you wish to use.

*If you have only one Receptor, it will appear automatically in the **Connected To** menu. If you have multiple Receptors, the first will appear, but you can select a different one in the **Connected To** menu.*

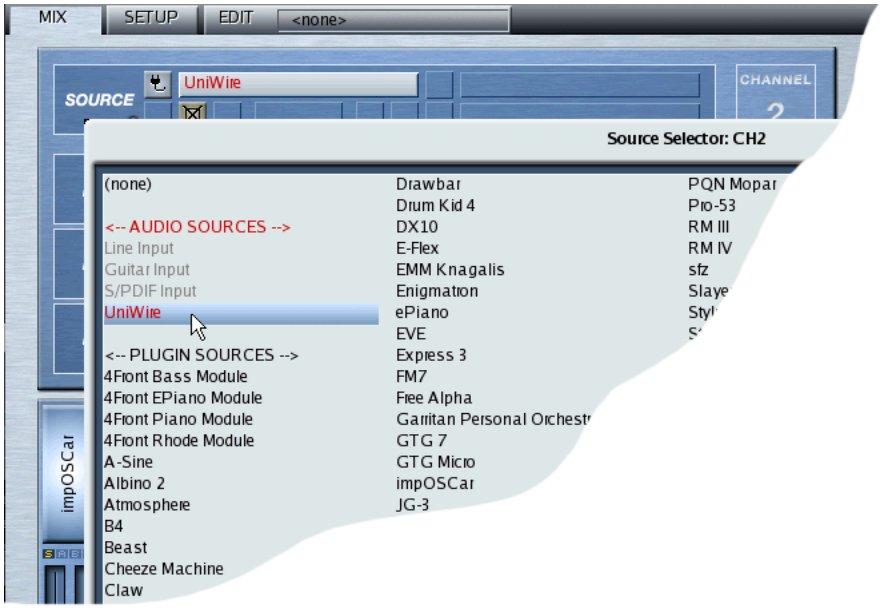
7 Set the desired **Latency**. This setting is global across all UniWire instances.

8 From the **Send Data To** menu, select which Receptor channel you want to use as an effects processor.

Audio will be sent from Cubase into this channel for processing.

9 Click the **Launch Receptor Remote** button to open Receptor's graphical user interface.

10 Using the channel you selected from the **Send Data to** menu, assign its Source to **UniWire.**



11 Assign effect plugins to one or more FX slots (A, B, or C)

12 Play your Cubase sequence and you should hear audio from the desired Cubase channel run through the Receptor effects chain, just as if the effects were running within the host application.

13 Save the Receptor effects chain as either a Single or a Multi patch, so that it will be recalled automatically when you open your sequence.

Bouncing to Disk in Cubase

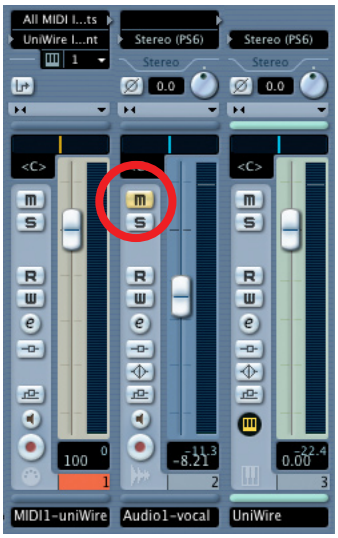
Obviously, there will come a point when you'll want to bounce your virtual UniWire tracks down to audio files. This process is nearly identical to any other bounce operation that uses external hardware. Essentially, you must render all UniWire tracks to disk in *realtime*. You cannot freeze UniWire tracks nor can you render them in non-realtime. If you've only ever used host-based instruments and effects, you might be confused by this limitation. But, if you think about it, both freezing and non-realtime rendering are operations that must occur using code that's running on the host computer: UniWired instruments, obviously, are being processed by remote hardware (Receptor), meaning they can only be bounced to disk in realtime.

Here's a simple example that illustrates how to bounce UniWired tracks to audio in Cubase:

1 For the sake of example, assume your Cubase sequence contains 2 tracks: a MIDI track (using a UniWire Instrument plugin); and an AUDIO track (using a file stored on your hard drive). You want to bounce the MIDI (UniWire) track to audio.

Notice that, in this example, both tracks are bussed to the same audio outputs ("Stereo (PS6)" in this case)

2 Mute the Audio track.



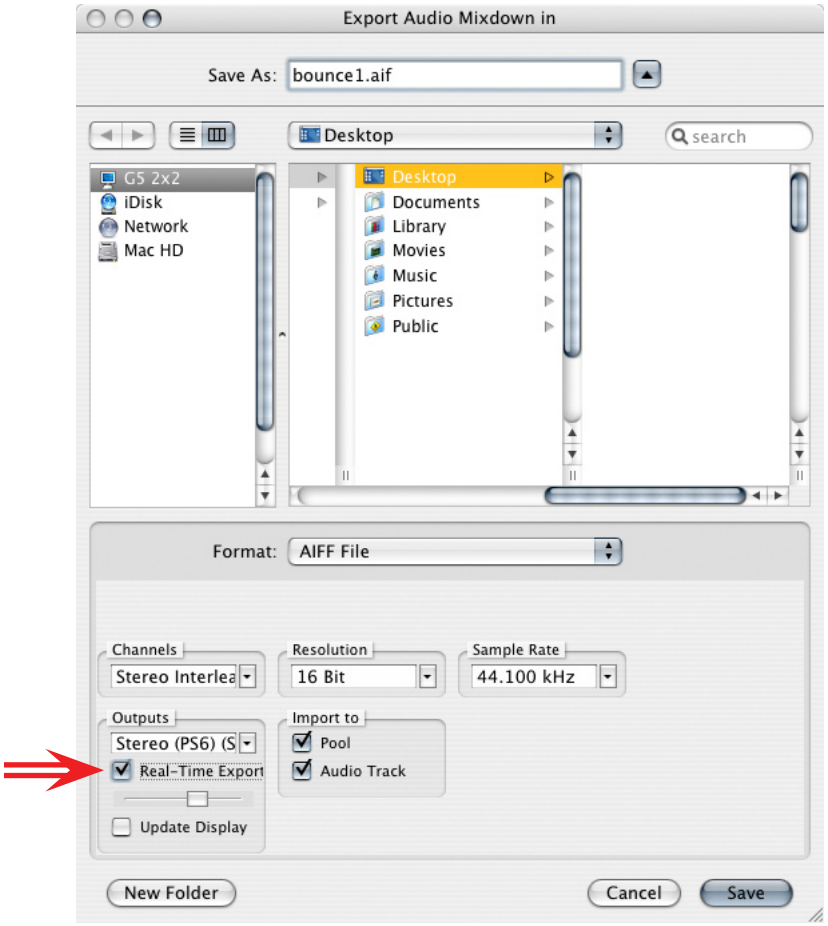
It's important that you mute any tracks whose signal you do not want captured in the bounce. The reason for this is that, when you perform a realtime bounce operation in Cubase, you record all audio that appears on a particular buss. So, in this example, since we simply want to bounce the UniWired MIDI track to disk, we've muted the Audio track so that only audio from Receptor appears on the "Stereo (PS6)" buss.

4 Select the region you wish to bounce to disk.

In this example, we've selected 4 bars, beginning with bar 1.

5 Choose **File>Export>Audio Mixdown**.

6 In the resulting dialog box, enter a name for the mixed file, select where you'd like to save it, select which buss (Outputs) to render to disk, configure the desired format options and, ***most importantly***, make sure to check the **Real-Time Export** check box.



7 You'll hear the audio play back in realtime as Cubase captures it, and you'll see a progress dialog box as the audio is captured.

7 When you have finished capturing the audio, you can choose to have it automatically imported into your current Cubase sequence.

As you can see in the following illustration, your UniWired track has been captured as an audio file and imported into Cubase. If you mute the UniWire (MIDI) track, you'll hear the bounced version, now fully rendered to disk.

To learn more about how to route audio signals within Cubase and how to render files to disk, please see your Cubase documentation.

Using UniWire with Ableton Live

This section assumes you know how to instantiate, play and record virtual instruments and effects within Ableton Live. If not, please consult your Live manual and make sure Live is working properly with host-based plugins before you use UniWire.

Live uses Receptor as an External UniWire Instrument

The following describes how to use UniWire with Ableton Live version 5:

1 Before beginning, make sure UniWire is enabled on Receptor.

*To do this from Receptor's graphical interface, click the SETUP tab and check the **Enable UniWire** option in the new UNIWIRE section in Setup View.*

To do this from Receptor's front panel, press the SETUP button and rotate the top display knob until the top line shows the "UniWire" option, then rotate the bottom knob to select `Enabled`.

2 In Live's Session View, create a new MIDI Track by selecting **Insert>Insert MIDI Track**.

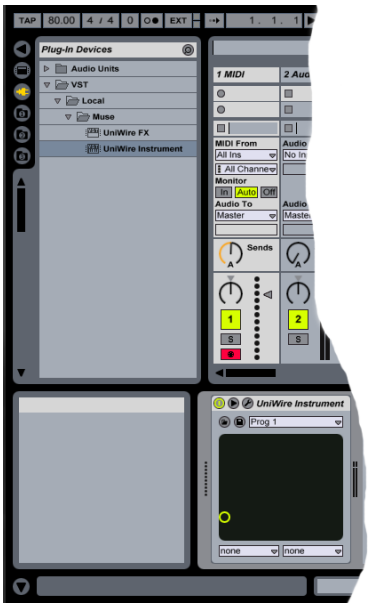
3 Click the Plugin Device Browser button and, using Live's built-in browser, navigate to the Muse folder contained within your plugin directory .

4 Click the little triangle next to the Muse folder to see the various types of UniWire plugins available.

You'll see a plugin called UniWire FX and another called UniWire Instrument. Since, in this example, you're using Receptor as a sound module, you'll use the UniWire Instrument plugin.

5 Drag the UniWire Instrument plugin from Live's Plugin Device Browser onto the MIDI track you created in step 1.

You'll see the UniWire Instrument plugin appear in the MIDI Track View at the bottom of Live's interface.



6 At the top of the UniWire plugin, click the Wrench Icon to open the UniWire Instrument's graphical editor .

7 From the **Connected To** menu, select which Receptor you wish to use.

*If you have only one Receptor, it will appear automatically in the **Connected To** menu. If you have multiple Receptors, the first will appear, but you can select a different one in the **Connected To** menu.*

8 Set the desired **Latency**. This setting is global across all UniWire instances.

9 Configure your UniWire and Receptor options as previously discussed. Some options include:

- *If you want to use an existing Multi patch and treat Receptor as if it were connected to your host computer via MIDI and audio cables, simply keep the **Send MIDI to** menu set to **All Receptor Channels** and select the desired Multi patch using UniWire's **Multi Bank** and **Multi Patch** selectors.*
- *If you want to communicate with each Receptor channel as if were its own external sound module, use the **Send MIDI to** menu to select which Receptor channel to communicate with. You will then have access to the **Single Bank** and **Single Patch** fields should you wish to select existing Receptor Singles via UniWire.*
- *If you want to configure Receptor manually, simply click the **Launch Receptor Remote** button and Receptor's graphical User Interface will appear in another window. Make sure, when you edit Receptor in this way, to save your edits to either a Receptor Single or Multi patch, so that they are automatically recalled by your host sequencer.*

Detailed discussions of every UniWire parameter were discussed earlier in this document.

10 Play a MIDI keyboard connected to Live and, if the keyboard is selected in Live's **MIDI From** field and the channel is record enabled, then you should hear Receptor just as if you had connected it to your computer using audio and MIDI cables.

Live uses Receptor as an External UniWire Effect

The following describes how to use UniWire to make Receptor act as an external effect within Ableton Live version 5:

1 Before beginning, make sure UniWire is enabled on Receptor.

*To do this from Receptor's graphical interface, click the SETUP tab and check the **Enable UniWire** option in the new UNIWIRE section in Setup View.*

To do this from Receptor's front panel, press the SETUP button and rotate the top display knob until the top line shows the "UniWire" option, then rotate the bottom knob to select `Enabled`.

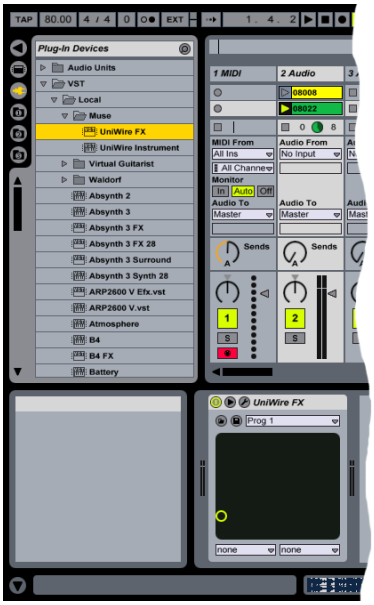
2 Click the Plugin Device Browser button and, using Live's built-in browser, navigate to the Muse folder contained within your plugin directory.

3 Click the little triangle next to the Muse folder to see the various types of UniWire plugins available.

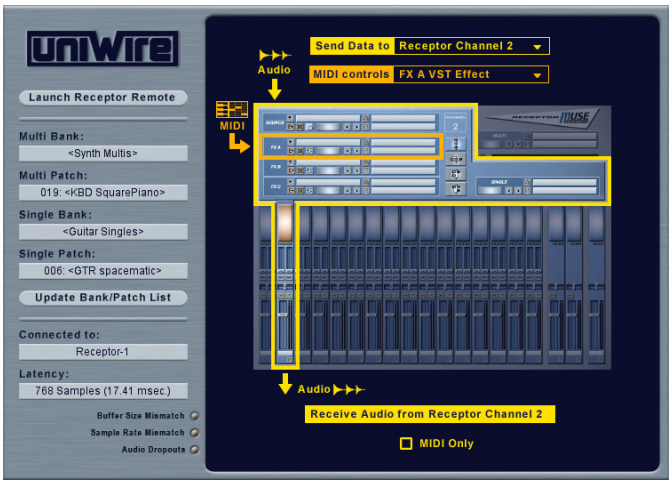
You'll see a plugin called UniWire FX and another called UniWire Instrument. Since, in this example, you're using Receptor as an effect, you'll use the UniWire FX plugin.

4 Drag the UniWire FX plugin from Live's Plugin Device Browser onto the Audio track you wish to effect.

You'll see the UniWire FX plugin appear in the Audio Track View at the bottom of Live's interface.



5 At the top of the UniWire plugin, click the Wrench Icon to open the UniWire Instrument's graphical editor .



6 From the **Connected To** menu, select which Receptor you wish to use.

*If you have only one Receptor, it will appear automatically in the **Connected To** menu. If you have multiple Receptors, the first will appear, but you can select a different one in the **Connected To** menu.*

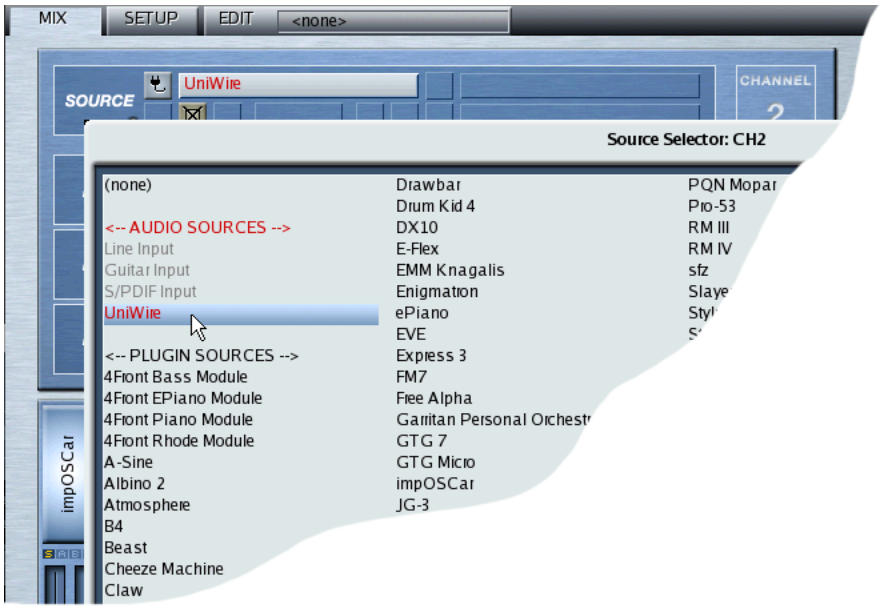
7 Set the desired **Latency**. This setting is global across all UniWire instances.

8 From the **Send Data To** menu, select which Receptor channel you want to use as an effects processor.

Audio will be sent from Live into this channel for processing.

9 Click the **Launch Receptor Remote** button to open Receptor's graphical user interface.

10 Using the channel you selected from the **Send Data to** menu, assign its Source to **UniWire.**



11 Assign effect plugins to one or more FX slots (A, B, or C)

12 Play your Live sequence and you should hear audio from the desired Live channel run through the Receptor effects chain, just as if the effects were running within the host application.

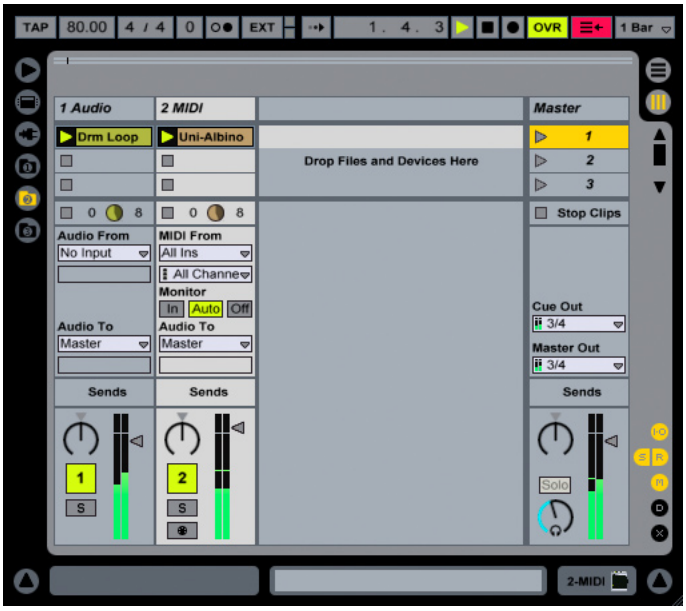
13 Save the Receptor effects chain as either a Single or a Multi patch, so that it will be recalled automatically when you open your sequence.

Bouncing to Disk in Ableton Live

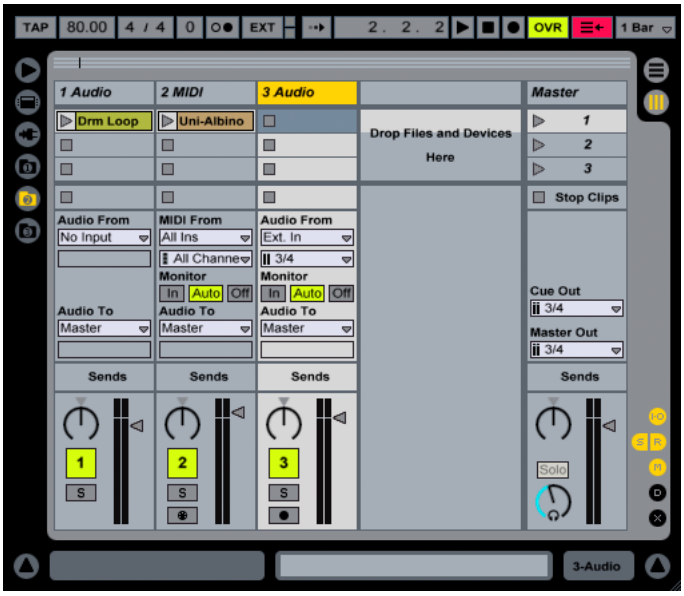
Obviously, there will come a point when you'll want to bounce your virtual UniWire tracks down to audio files. This process is nearly identical to any other bounce operation that uses external hardware. Essentially, you must render all UniWire tracks to disk in *realtime*. You *cannot* freeze UniWire tracks nor can you render them in non-realtime using Live's **Render to Disk** command. If you think about it, this makes sense, because both freezing and non-realtime rendering are operations that must occur using code that's running on the host computer: UniWired instruments, obviously, are being processed by remote hardware (Receptor), meaning they can only be bounced to disk in realtime.

Here's a simple example that illustrates how to bounce UniWired tracks to audio in Live:

1 For the sake of example, assume your Live sequence contains 2 tracks: a MIDI track (using a UniWire Instrument plugin); and an AUDIO track (using a file stored on your hard drive). You want to bounce the MIDI (UniWire) track to audio.

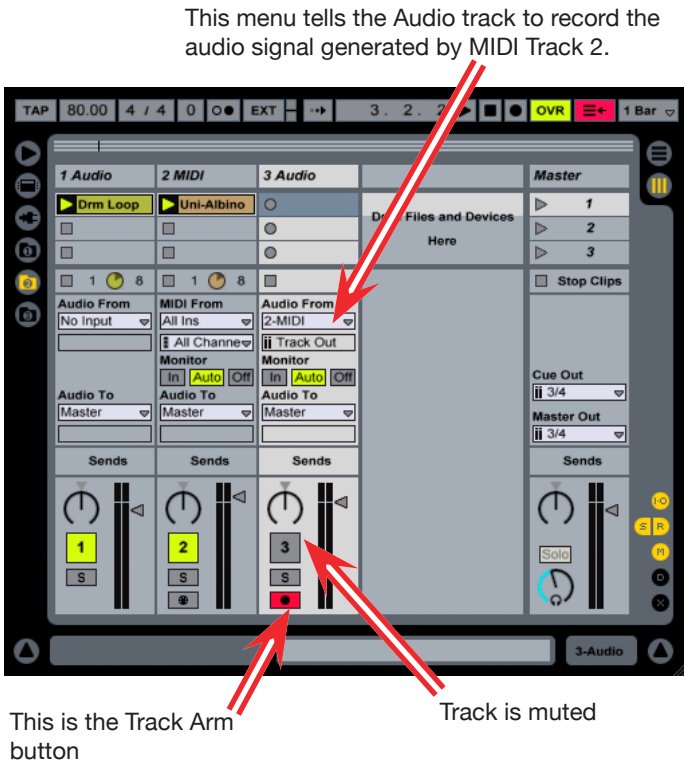


2 Create a new audio track by choosing **Insert>Insert Audio Track**.



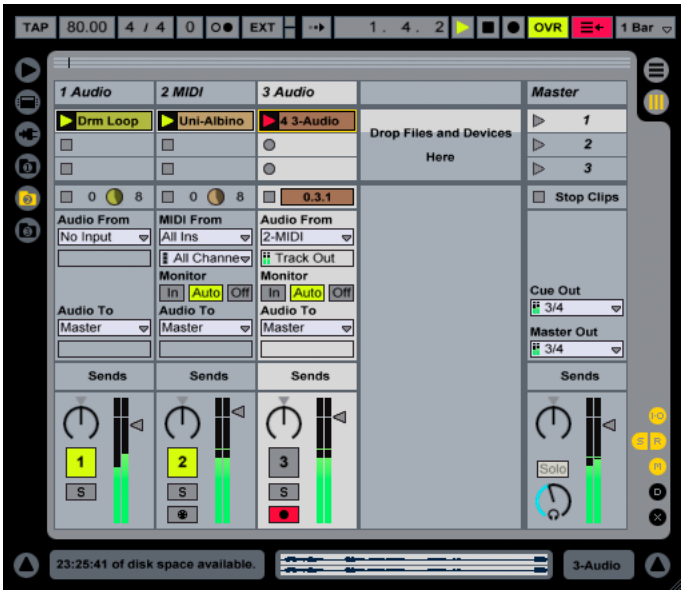
3 In the new audio track, use its **Audio From** menu to select the output from the MIDI track that contains the UniWire instance you wish to render. Then record enable the new audio track by clicking its **Track Arm** button.

You might also wish to mute the output of the new audio track, so that you don't hear both the UniWire (MIDI) and the bounced (audio) track play during capture.



4 Decide which slot should contain the rendered clip, then click the Record button for that slot.

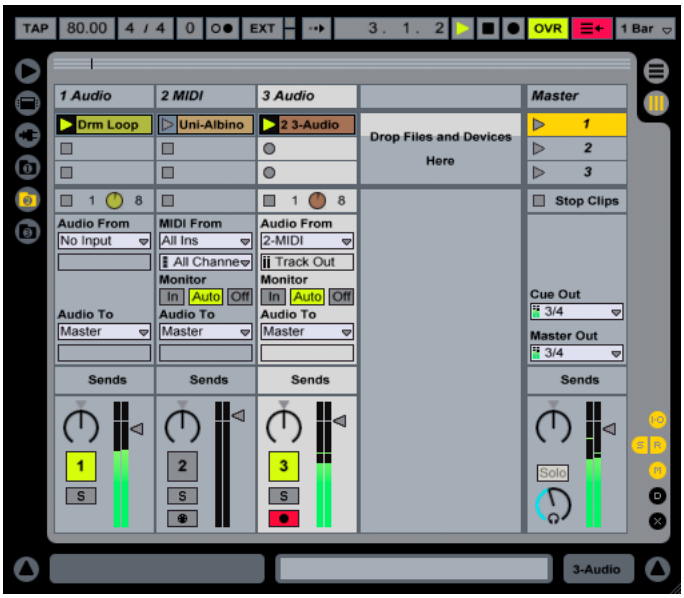
Live records the output of the MIDI track as a new Audio clip, in real-time.



5 Adjust the length of the resulting audio clip as desired.

6 Mute the MIDI (UniWire) track, un-mute the new Audio (rendered) track, and play back the sequence.

You'll hear the bounced version, now fully rendered to disk.



To learn more about how to route and record signals within Ableton Live, please see your Live documentation.