Qtractor Manual & How-To’s

Qtractor Wiki

rncbc *et al.* 2022-05-31 14:45+0100

Manual - Table of Contents

This wiki version of the manual is based on a document originally authored by Rui Nuno Capela, James Laco Hines and Stephen Doonan.

Table of Contents

4. Qtractor–An Overview

Manual - 1 Introduction

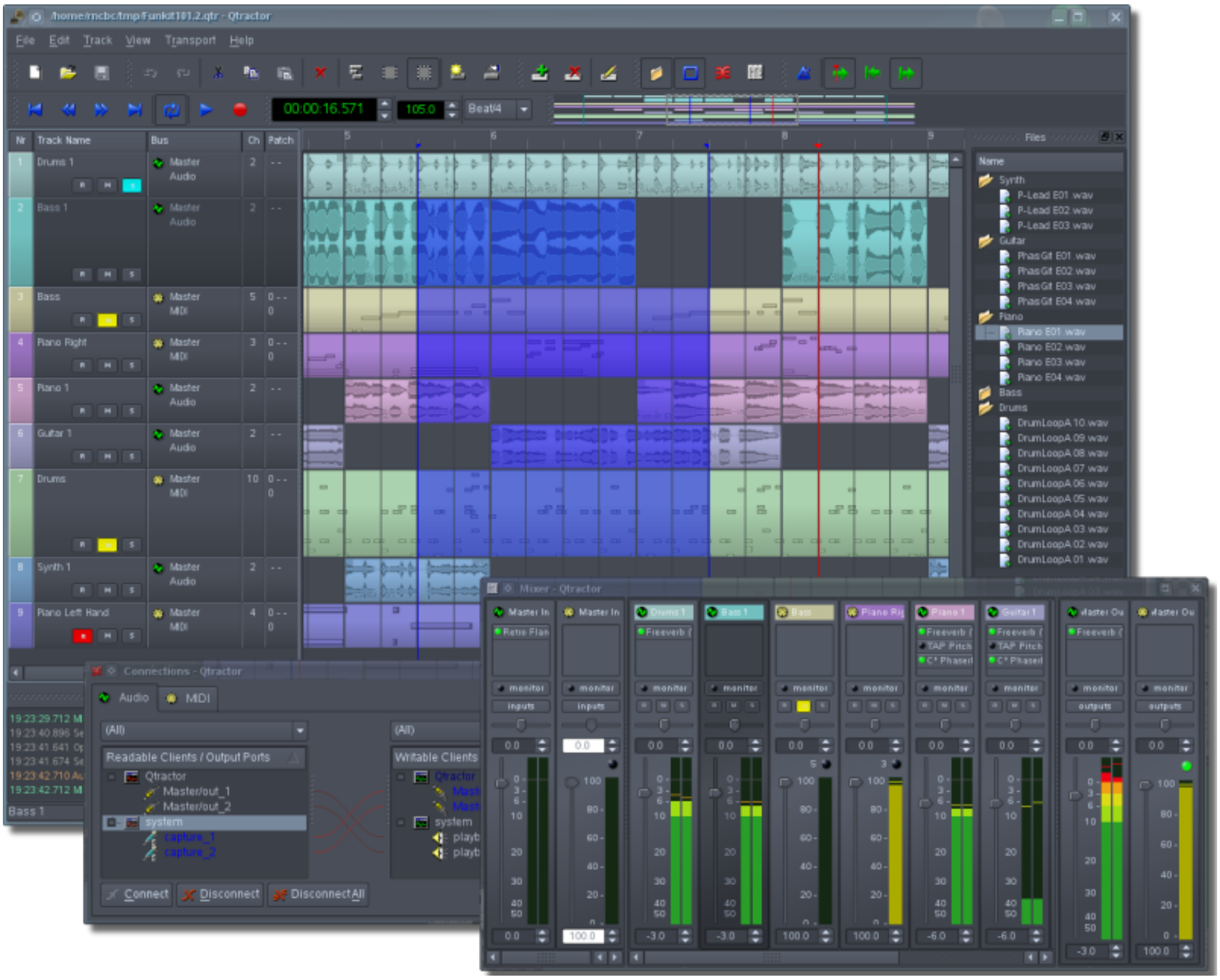
Manual - Table of Contents

1. Introduction

1.1. Abstract

Qtractor is a multi-track audio and MIDI recorder and editor. The program is written in C++ and uses the Qt toolkit for the GUI (graphical user interface) elements. Qtractor is free and open source software, licensed under the GNU General Public License version 2 or later.

Qtractor runs exclusively under GNU/Linux and depends upon ALSA (Advanced Linux Sound Architecture) and JACK (the Jack Audio Connection Kit) to provide its MIDI and audio IO infrastructure. It currently has one developer, the originator of the project, Rui Nuno Capela. Development was started in April of 2005, initially as a Qt 3 application. Since October 2015, it is officially a Qt 5 application.



Main GUI window showing audio & MIDI tracks, Mixer & Connections windows

The project began as a sequencer intended mainly to be used with MIDI hardware devices but has since developed comprehensive support for plugins and gained moderately powerful audio recording and editing facilities. It now aims to be a lightweight but reasonably powerful solution for hobbyist and semi-pro musicians. It does not aim to be a “do-it-all monolith DAW” (such as you may find in the commercial world), nor is it completely “modular” - rather, it should be considered something of a “hybrid”, which contains the kind of features that are most likely to be useful for its intended user-base. In the words of its developer, rather than a fully-fledged DAW, Qtractor is *a sequencer with DAW-like features*.

If your intention is to mimic the functionality of a recording studio, conducting a great deal of live audio recording and performing complex routing operations, something like Ardour may be more suited to your needs.

1.2. Introduction

Whilst Qtractor is still in its beta phase of development, it can already be comfortably used by hobbyists as a personal home recording studio or “bedroom studio”. It can record, import, arrange and edit both digital audio and MIDI data. Its interface will be familiar to users of other popular multi-track recording/editing applications.

In addition to recording digital audio and MIDI, Qtractor provides an environment for multi-track clip-oriented composing techniques common in modern music-making and aims to be intuitive and easy to use, yet powerful enough for the serious recording enthusiast.

Qtractor is a non-destructive audio editor. This means that your session’s audio files remain unchanged on disk after you have edited them. Instead, all edits to audio files are stored within the Qtractor session file.

As part of the GNU/Linux audio and MIDI ecosystem, Qtractor relies on other projects to provide functionality in various areas. Due to this approach, the user must consider his/her needs in areas such as synths, samplers, drum machines, effects processors and sounds (audio samples, sample packs (sf2/sfz/gig etc.), synth patches and so on) and use these other projects in tandem with Qtractor by referencing the information in this wiki. If you have no idea where to start and find the information in the wiki lacking, try searching online or checking sites such as the Linux Audio Wiki and LinuxMusicians. There are also various projects on Rui Nuno Capela’s site which should enable you to get up and running reasonably quickly.

Manual - 2 Installing and Configuring Qtractor

Manual - Table of Contents

2. Installing and configuring Qtractor

2.1. Compiling Qtractor

If Qtractor is not available from your Linux distribution's package manager/repository and nor can you find a standalone package (such as a .deb package for Debian, Ubuntu or other Debian-based system or an .rpm package for Red Hat, Fedora, SUSE, etc.), then you will have to build it from source. Doing so requires you have both a C++ compiler (such as g++, the GNU C++ compiler) as well as several other dependencies, the most important of which are listed below.

Those with experience in compiling Linux/UNIX software using autoconf should find the build process straightforward. Building Qtractor is fairly easy but both the compilation process and setting up a build environment (a collection of programs necessary for compiling software) can be a bit daunting to those new to Linux. You don't need to be a programmer to compile software but you will at least need to know the basics of using the Linux command line and how to use your distribution's package manager.

2.2. Build dependencies

In order to compile Qtractor you are required to have at least all the packages listed under section 2.2.1 installed. Everything listed under section 2.2.2 is optional but recommended to get the most out of Qtractor.

Note that some Linux distributions (such as Arch) include the development files in their packages whereas other distributions split packages up into binary and dev components. For example, the ALSA library under Debian and Ubuntu is contained in the package libasound2 but its development files are contained within libasound2-dev.

If you are compiling Qtractor under a Debian or Ubuntu-based distro you can install all the required build dependencies with one command by running:

```
sudo apt-get build-dep qtractor
```

2.2.1. Mandatory packages

- **Qt** framework (core, gui, xml) - C++ class library and tools for cross-platform development and internationalization https://qt.io/
- **JACK** - JACK Audio Connection Kit http:s//jackaudio.org/
- **ALSA** - Advanced Linux Sound Architecture https://www.alsa-project.org/
- **libsndfile** - C library for reading and writing files containing sampled sound http://www.mega-nerd.com/libsndfile/

- **LADSPA** - Linux Audio Developer's Simple Plugin API http://www.ladspa.org/
-

2.2.2. Optional packages

- **libvorbis** (enc, file) - Ogg Vorbis audio compression https://xiph.org/vorbis/
- **libmad** - High-quality MPEG audio decoder https://www.underbit.com/products/mad/
- **libsamplerate** - Secret Rabbit Code; C library for audio sample rate conversion http://www.mega-nerd.com/SRC/
- **librubberband** - Rubber Band Audio Time Stretcher; audio time-stretching and pitch-shifting library https://breakfastquay.com/rubberband/
- **liblo** - Lightweight OSC implementation (required for DSSI GUI and NSM support) http://liblo.sourceforge.net/
- **DSSI** - API for soft synth plugins with custom user interfaces http://dssi.sourceforge.net/
- **VST-SDK** - Steinberg's Virtual Studio Technology plugin format https://www.steinberg.net/
- **LV2** - Audio plugin standard; extensible successor to LADSPA https://lv2plug.in/
- **liblilv** - Lightweight LV2 implementation stack https://drobilla.net/software/lilv/
- **libaubio** - Library for real time audio labeling https://aubio.org/

2.3. Downloading Qtractor

2.3.1. Official releases

Qtractor is considered to be in its beta stage of development but is already fully functional. The latest official releases are publicly available from the qtractor.sourceforge.net project web site:

https://qtractor.org/

2.3.2. Qtractor for the experienced and adventurous

The latest "bleeding-edge" source code can be obtained from the qtractor git repository with the following command:

```
git clone https://git.code.sf.net/p/qtractor/code qtractor-code
```

Then go to the just created `qtractor-code` source tree directory:

```
cd qtractor-code
```

You should now be ready to build and install Qtractor.

NOTE: After the initial checking-out of the source code, if you want to update to the latest version in future you can fetch/merge the latest changes by using the following command from within the `qtractor-code` directory:

```
git pull
```

2.4. Compiling and installing Qtractor

2.4.1. Standard compilation and installation

The installation procedure follows the cmake build system (make sure you have version 3.15 or higher). From within the Qtractor source directory, run:

```
cmake -B build .
cmake --build build
```

and optionally, as root:

```
cmake --install build
```

2.4.2. Compiling Qtractor with debugging support

If you find and report a bug in Qtractor, you may be asked to provide some debugging information. This short guide will explain how to build Qtractor with debugging enabled, making it easier to pinpoint the problematic code.

Rebuild it all from scratch, with:

```
cmake -DCMAKE_BUILD_TYPE=Debug -B build
cmake --build build
```

Enable core dumps in a shell session:

```
ulimit -c unlimited
```

From the same shell command line, run the program until it crashes. You'll see something like this in the output when it happens:

```
Segmentation fault (core dumped)
```

Locate the dumped core file. Depending on your environmental settings it might be named just `core`, or something like `core.1234` (1234 is the process-id number of the crashing program), located in the last directory where the program was.

Load the core dump file into gdb:

```
gdb ./build/src/qtractor /path/to/core
```

At the gdb prompt just enter:

```
gdb> bt
```

or:

```
gdb> thread apply all bt
```

2.5. Qtractor's Configuration Settings File

Qtractor keeps its settings and configuration file in:

```
$HOME/.config/rncbc.org/Qtractor.conf
```

Normally, there is no need to manually edit this file. Most of Qtractor's options can be adjusted from the Options window under the View menu.

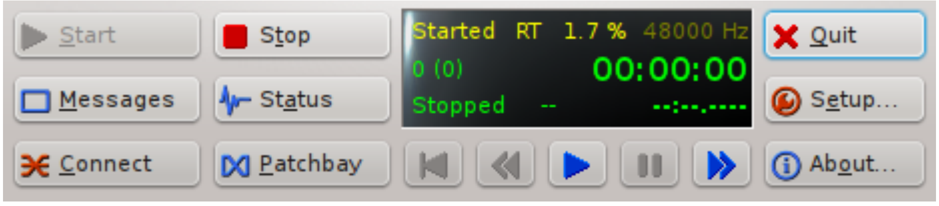
2.6. QJackCtl

In order to use Qtractor, JACK must be running in the background. JACK is a kind of patchbay for Linux (technically a "sound server"), enabling you to route audio and MIDI data between Qtractor and other JACK applications.

Some distributions may configure Qtractor such that JACK starts automatically when Qtractor is launched; others may not. If you launch Qtractor from your application menu and receive JACK errors in the message log at the bottom of the main workspace, quit Qtractor, start JACK manually, then launch Qtractor again.

The easiest way to start JACK is with QJackCtl ("Q JACK Control"), a control panel and global, synchronized timecode display. If this is not installed, install it from your distribution's repository or from the project's site.

If necessary, launch QJackCtl and click the Start button. This should begin a global, synchronized timecode clock that will allow Qtractor to use all available inputs and outputs just like a studio mixing board.

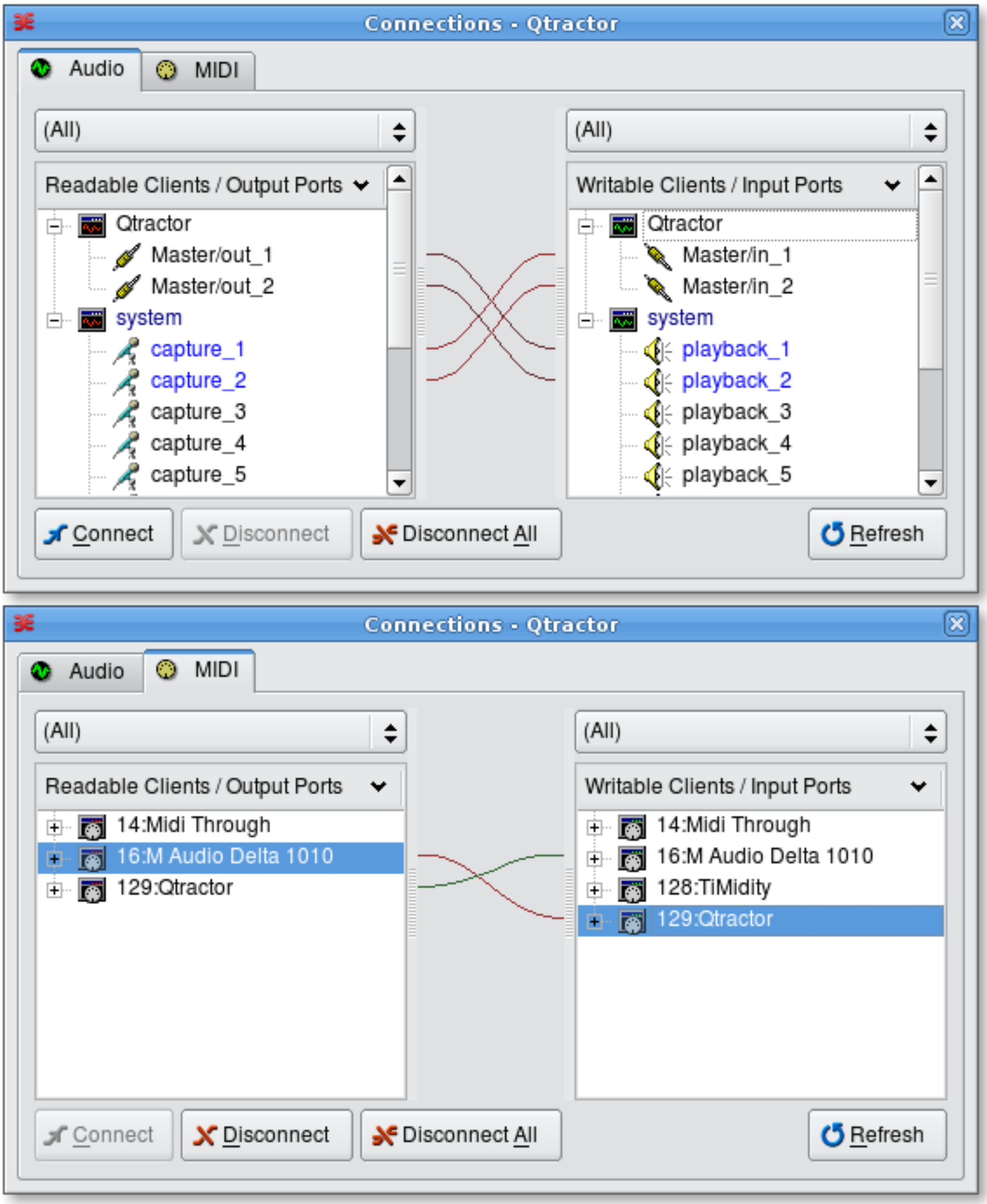


QJackCtl in operation

2.7. Audio and MIDI Input

Qtractor can record both digital audio and MIDI data, but it does not magically know what audio or MIDI data you wish to record; you must route audio and MIDI data to it manually. This can be done with QJackCtl, via the command line or using Qtractor's Connections window, pictured below.

Qtractor depends upon ALSA and JACK to utilize your audio hardware. It uses ALSA to communicate with MIDI hardware and JACK to route audio to/from various hardware and software "ports".



Qtractor's Connections window, showing both the Audio and MIDI tabs. "Readable" ports are sources of data (where audio or MIDI data can come from, e.g. a microphone input or a MIDI keyboard controller) while "Writeable" ports are places that data can be routed to, such as a track, an audio output or a MIDI synthesizer

Manual - 3 Learning Qtractor–An Example Session

Manual - Table of Contents

3. Learning Qtractor–An Example Session

This chapter is written for those who may not be very familiar with digital audio recording or MIDI “sequencing” applications, or who wish to gain a quick overview of how Qtractor works. It can be used before exploring the program in greater depth and learning its features in detail. It describes an example Qtractor session and serves as a basic walk-through of the program.

3.1. Preparation

You’ll use Qtractor to import a pre-recorded audio file and record a MIDI track. A MIDI-triggered tone generator (in this case, a hardware tone generator outside the computer, although it could just as easily be a “soft synth” inside the computer) will produce the sound for the MIDI track as Qtractor plays it back.

To begin:

1. Start the QJackCtl application (recommended), or start `jackd` from a shell. After launching QJackCtl, press the **Start** button to start JACK

2. Launch Qtractor from your applications menu, dock, or launcher

3. It’s good practise to **start every session by saving**. It might seem strange to save an as-yet empty session, but it’s better to save an empty session than to start creating your masterpiece and have data files and MIDI files scattered all throughout your hard drive. Saving first is a good way to instantiate an environment in which you can keep all of your files and sounds organized and consolidated

To save, click the **File** menu and select **Save As**. This opens the Session Properties window.

In the **Name** field, name your session. For the **Directory** field, click the directory icon to create a new, empty directory for your session files and click **OK**. Save your session by clicking the **OK** button; a **Save Session** dialogue will open so that you can navigate to the directory you’ve created and name your session file, which will appear as a `.qtr` file. Click the **Save** button to confirm.

Now that the session has been saved, open the **Session Properties** window again using the menu item **File** / **Properties**. This time, click the **Properties** tab and set both the time signature and the tempo so that you can use the metronome to help you keep MIDI parts synchronized as you record them.

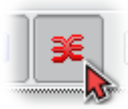
3.2. Importing an Audio File

The first thing you may like to do is import a pre-recorded audio file of drums and other percussion, to form the basis of the recording session. To bring audio into your project file, right-click in the *Files* pane on the right hand side of the main workspace and select *Add Files*, then choose the file or files you wish to import from the **Open Audio Files** window that appears. If the Files pane isn't visible, you can enable it by clicking on the yellow file icon, the one without an arrow above it. To place an audio file in a track, drag and drop it from the Files pane into the workspace. You can add it to an existing track, or drop it directly into an empty area and a new track will be created automatically.

3.3. Connecting the MIDI data source to Qtractor

Your MIDI piano-like keyboard normally routes its MIDI data (created when you strike the keys, for example) to its own internal tone generator, which then sounds like you're playing a real piano, or electric piano, harpsichord, bass guitar, etc. However, for this project in Qtractor you want to route the external keyboard's MIDI data to Qtractor. So, using a standard MIDI cable (or perhaps a MIDI to USB converter, depending on your set-up), connect the keyboard's output to the input of your sound card. Inside the computer, you will route the MIDI data from the sound card to a Qtractor MIDI input bus.

Next, open Qtractor's Connections window by clicking on the red icon near the top of the main workspace.



Connections icon

In the **Connections** window's **MIDI** tab, connect the MIDI output of the sound card (in the left pane, marked *Readable Clients / Output Ports*) to Qtractor's listing in the right pane (marked *Writable Clients / Input Ports*). If your sound card doesn't appear in the list, click the **Refresh** button in the bottom right to update the list of connections.

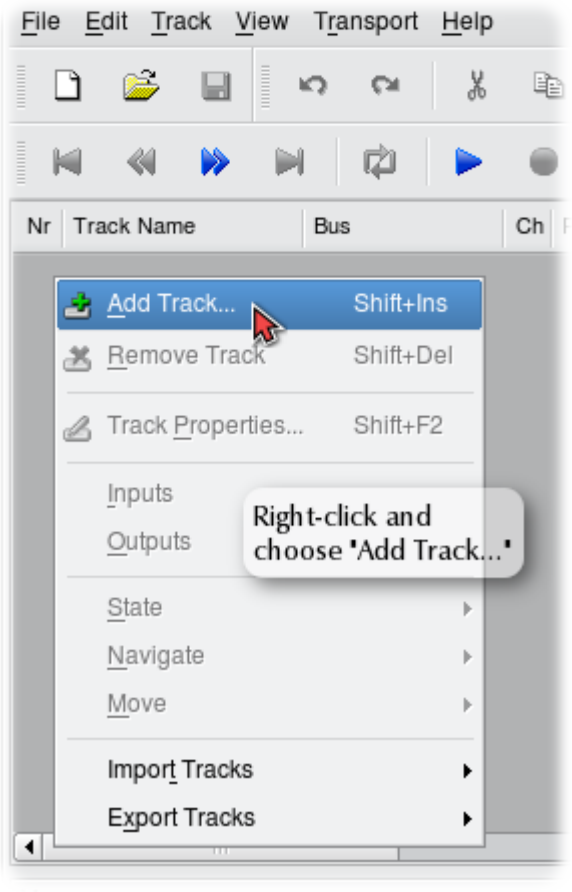
You can connect an output port (the source of the data, MIDI or audio) to an input port (the port that will receive the data) in several ways. One method is to left-click a port in the left pane, left-click a port in the right pane, then either left-click the **Connect** button or right-click elsewhere in the window and select *Connect* from the pop-up menu. Another method is to click a port in the left pane and, with the mouse button held down, drag the cursor to a port in the right pane until the port is highlighted, then release the mouse button. Regardless of the method you use, a line representing a "virtual cable" will appear between the two ports in the middle section of the window.

After connecting your sound card to Qtractor, you'll then need to connect Qtractor's output to your destination source, so that it can send the MIDI information back out again. Again in the **Connections** window, connect Qtractor's *Master/Out* listing in the left pane to your sound card in the right pane.

Once you've made your connections, close the Connections window by clicking its icon once more.

3.4. Creating a MIDI track

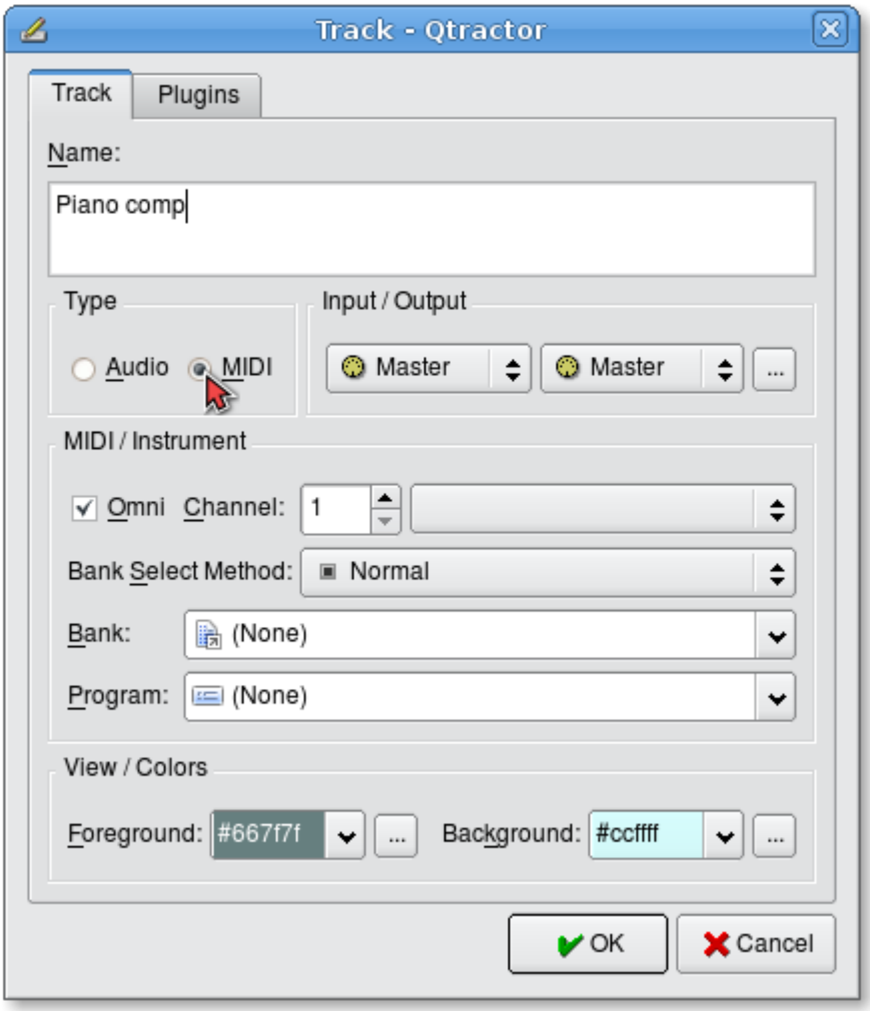
Now that Qtractor can receive and send MIDI data, it's time to create the first track in order to record that data. Right-click in the blank pane on the left of Qtractor's main window and, from the pop-up menu, choose *Add track…*. You can also do this via the menu item **Track / Add Track…**.



Right-click and choose 'Add Track...'

Adding a track

Qtractor’s **Track Properties** window will then open.



Name: Piano comp

○ Audio ● MIDI

Input / Output: Master, Master

☑ Omni Channel: 1

Bank Select Method: Normal

Bank: (None)

Program: (None)

Foreground: #667f7f

Background: #ccffff

Track Properties window

In the **Track Properties** window, click the **MIDI** radio button: you want this track to be used for MIDI, rather than audio, data. As soon as you click the MIDI button, the track is automatically connected to Qtractor's Master MIDI input and output buses. These can be changed by using the drop-down lists and the ellipsis button in the **Input / Output** area of the Track Properties window, but there is no reason to do so right now. You can replace the default name "Track 1" with, for instance, "Piano comp."

You can leave the MIDI **Channel** at the default "1" for now, as this channel designation is mainly for output. That is to say, when the track is played back and its MIDI data is sent to somewhere outside Qtractor (in this case, a hardware tone generator) that data will be sent on the MIDI channel specified here (there are 16 possible channels). However, you're going to record now, so the eventual output MIDI channel doesn't matter at the moment - it can easily be changed later before you play your MIDI tracks back.

What *does* matter however is the **Omni** checkbox. If enabled, Qtractor will record MIDI information received on *any* MIDI channel (omni means all). If disabled, it will record only information it receives on the channel number specified in the **Channel** setting, to the right of the Omni checkbox. Your MIDI keyboard might be set to transmit MIDI data on only one channel (e.g., MIDI channel 1), so you need to make sure that each of the MIDI tracks you plan to record on will receive data either from the correct channel, or from all channels. Click the **OK** button and a new empty track appears in Qtractor's main window.

3.5. Recording and Playback

In order to record, you must arm the track by clicking on the “Record” button for that track. This is the “R” button that is present on every track strip, besides the other “M” and “S” buttons (respectively for “Mute” and “Solo” track state settings). You can also do this via the menu item **Track / State / Record**. Either way, the “R” button will turn red (or become highlighted, depending on your GUI settings) and the track is then set armed and ready for recording.

Next, turn on session recording mode by clicking on the Record button in the main transport toolbar (big red circle). You can also do this via the menu item **Transport / Record**.

Now you’re ready to record your first MIDI track. Take a little breath and… press “Play” (blue triangle next to the transport’s Record button), the space bar or choose the **Transport / Play** menu item: you’re rolling. Hit the MIDI keyboard and see a brand new MIDI clip taking shape while you’re performing. When done, press “Play” again to stop.

As previously mentioned, when playing back a MIDI track its data is sent on the channel number specified in the **Channel** setting of the **Track Properties** window. In order for your hardware tone generator to play back the MIDI data from Qtractor as actual sound, you’ll need to make sure that this channel tallies with your hardware set-up.

3.6. Next Steps

This chapter has been merely an introduction to what is a fairly far-reaching subject. More detailed information on the above topics, plus many other aspects of Qtractor, can be found in the next chapter. Specifically, see here for more information on working with MIDI and here for audio. A sample MIDI workflow tutorial using a soft synth can be found in this How To.

Manual - 4 Qtractor–An Overview

Manual - Table of Contents

4. Qtractor–An Overview

4.1. Routing–Connections, Ports, Tracks and Buses

4.1.1. Routing–General Concepts and Information

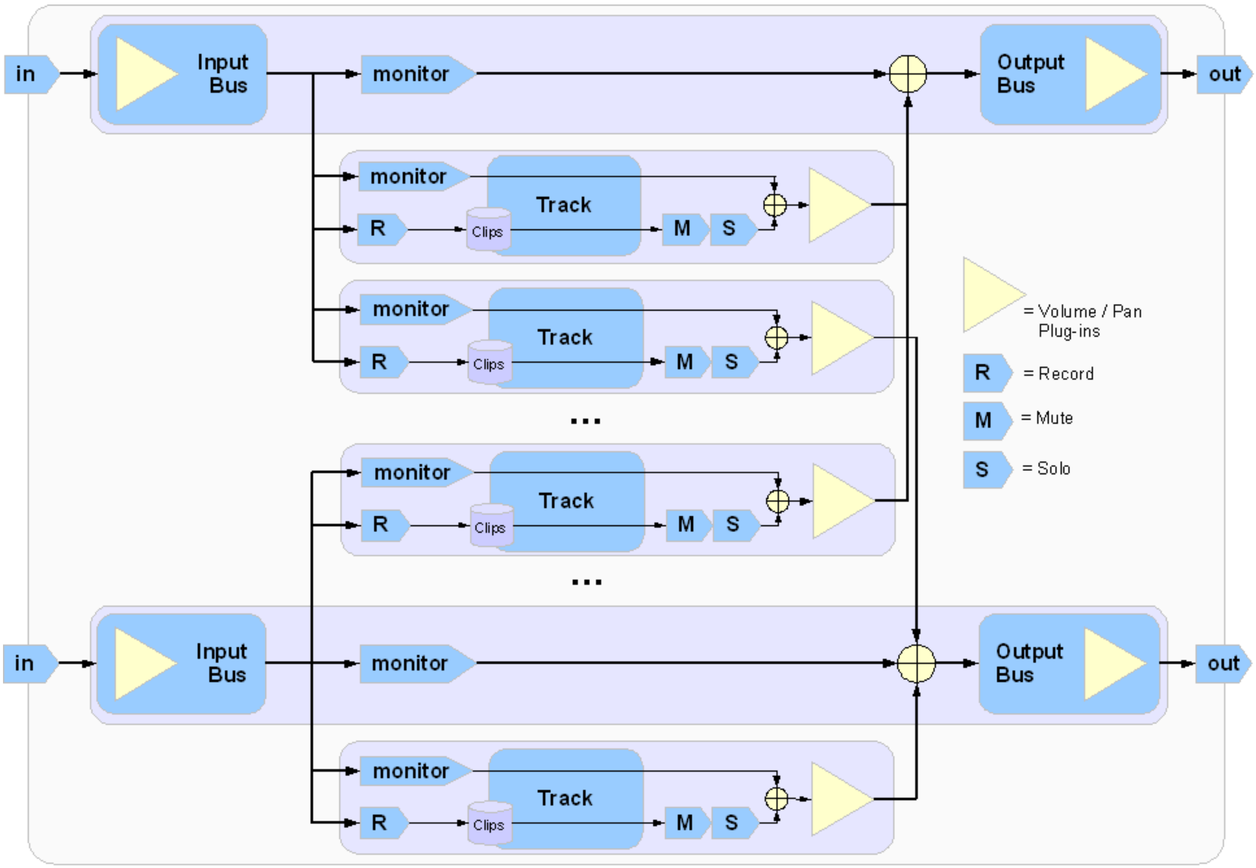
Qtractor can record and play digital data, specifically audio and MIDI data. To record, Qtractor must *get* the data from somewhere; to play the data, Qtractor must *send* it to somewhere that can understand the data stream and produce sound. The process of directing such data to and from – and through – software and hardware is called *routing*. In Qtractor, not much routing is done automatically. Most of it is left to the user, because it is the user's preferences, desires and goals in any specific project that determine how the audio and MIDI data should be routed.

In order to route data, one must have an understanding of the various connections and pathways that are available. A *bus* is like a pipeline through which one or more streams of data can travel. A *port* is like a valve at one or both ends of a bus. A port is normally closed, not allowing any data to flow through, but once a connection is made to it, it opens to allow data to pass through it from one bus to another. However, simply because a connection is made and the port is open doesn't mean that data is flowing through it: the data flow is initiated by the software or hardware at one or both of the ends of the bus.

A *track* can be thought of as a place where digital audio or MIDI data is deposited, rather than a pathway through which such data moves. Data can be deposited in various ways, such as recording, pasting or importing. Tracks are, however, still part of the signal or data flow, as they can "pluck" a copy of their data and send it to one or more buses (and, through those buses, to other destinations). In concrete terms:

- During recording, a track becomes the *destination* for audio or MIDI data entering Qtractor's input bus(es)
- During playback, a track becomes the *source* of audio or MIDI data, which is then sent through a Qtractor output bus to a destination, such as a sound card and its attached speakers

Below is a representation of the connections (ports), buses and routes in Qtractor through which audio or MIDI data can flow.



Audio & MIDI data flow, in, through and out of Qtractor

Routing in Qtractor is very flexible and therefore signal/data flow can become very complex. Data streams can merge and flow through the same bus, or split and go through separate buses to different destinations. They can likewise flow through different buses to the same destination, whilst passing through other intermediate software or hardware in which the data is possibly manipulated in some way. Because of this complexity it is beneficial to develop a solid understanding of the issues related to routing, the many possible routes that data can take and the many possible connections that can be made.

4.1.2. Routing in Qtractor

Routing can be accomplished in several ways, including:

- Using the Connections window to connect Qtractor's input and output buses to outside *sources* (for "reading" and, possibly, recording data) and *destinations* (for "writing" data to, for example, a sound card)

- Using the Track Properties window to assign a track's input and output buses

- Setting the state of a track, from among Record, Mute, Solo and Monitor (some of which are mutually exclusive)

- Adding Plugins, Aux Sends or Inserts to tracks or buses. **Aux Sends** are used to route audio to *internal* destinations, i.e. Qtractor buses; **Inserts** are used for *external* destinations, i.e. applications outside of Qtractor

4.1.3. Routing–Technical Notes

Qtractor is a fairly massive multi-threaded application. For instance, each audio clip has a dedicated disk I/O executive thread, which synchronizes with the master engine, and to the central JACK real-time audio processing cycle, through a lock-free ring-buffer. These audio file ring-buffers are recycled (filled/emptied) at a one second threshold and have a maximum streaming capacity of 4-5 seconds of audio sample data. Smaller clips are permanently cached in a RAM buffer.

Audio thread scheduling is mastered and mandated through the JACK callback API model. MIDI clip events are queued in anticipation through one MIDI output thread, which feeds an ALSA sequencer queue, synchronized on one second periods to the JACK process cycle. A single thread is responsible for listening (polling) for MIDI input, and multiplexes all incoming events through record-armed MIDI tracks. Time stamping is done through the ALSA sequencer facility.

Looping is made possible through the audio file buffering layer, right at the disk I/O thread context. The same consideration is adopted for MIDI output queuing. Audio frame relocation is accounted from successive JACK client process cycles (i.e. buffer-period resolution).

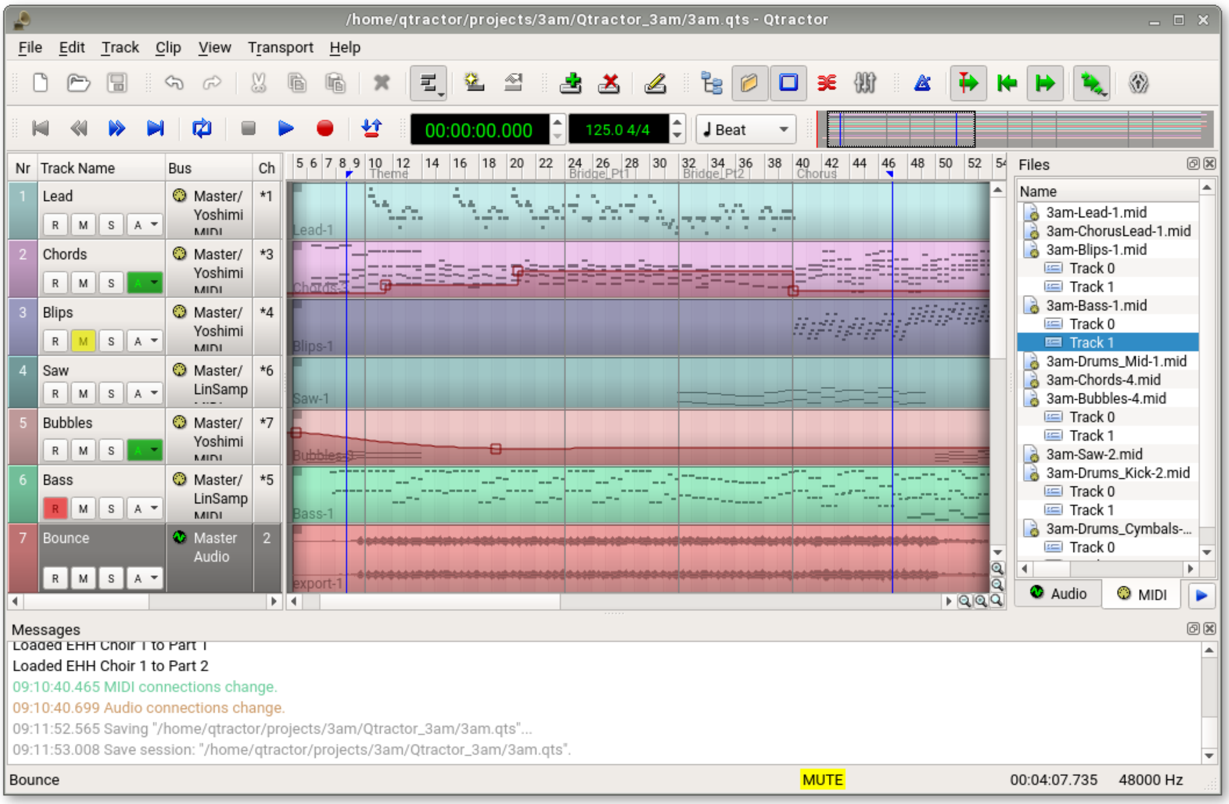
In this particular design, JACK and ALSA sequencer ports are logically aggregated as buses with respect to the audio and MIDI signal routing paths, functioning as fundamental device interfaces. Input buses, through exposing their respective input ports, are the inlets responsible for capture and recording. Output buses are the main signal outlets and are responsible for providing playback and, more importantly, mix-down connections.

Buses are independently assigned to tracks. Each track is assigned to one input bus for recording, and to one output bus for playback and mix-down. The assigned bus determines the number of channels the track supports. Clips bound to disparate multi-channel audio files, for which the number of channels does not match the track's, are automatically resolved on mix-down. The illustration in the above section shows one typical signal flow block diagram.

4.2. Qtractor's Main Window and Work Area

Qtractor's graphical user interface follows a standard design of most modern digital audio and MIDI workstations. The interface is easy and intuitive enough to easily interact with in order to discover the potential of the underlying inner core of the application where its functionality is implemented.

The following illustration shows an overall view of the GUI with an example session loaded into the workspace.



Nr	Track Name	Bus	Ch
1	Lead	Master/ Yoshimi MIDI	*1
2	Chords	Master/ Yoshimi MIDI	*3
3	Blips	Master/ Yoshimi MIDI	*4
4	Saw	Master/ LinSamp	*6
5	Bubbles	Master/ Yoshimi MIDI	*7
6	Bass	Master/ LinSamp MIDI	*5
7	Bounce	Master Audio	2

Messages

```
Loaded EHH Choir 1 to Part 1
Loaded EHH Choir 1 to Part 2
09:10:40.465 MIDI connections change.
09:10:40.699 Audio connections change.
09:11:52.565 Saving "/home/qtractor/projects/3am/Qtractor_3am/3am.qts"...
09:11:53.008 Save session: "/home/qtractor/projects/3am/Qtractor_3am/3am.qts".
```

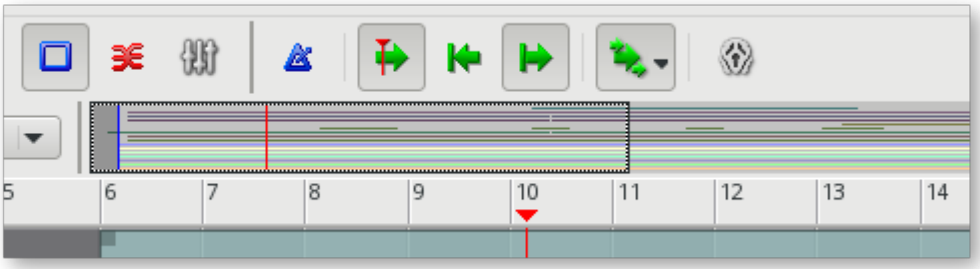
Bounce MUTE 00:04:07.735 48000 Hz

Qtractor's main window and work area

The main Qtractor window is initially laid out in this fashion:

- **Top** - Menus and toolbar icons, with navigation window in bottom-right
- **Left** - Tracks area, showing name, state, buses etc.
- **Middle** - Audio and MIDI clips area, with timeline above
- **Right** - Files pane
- **Bottom** - Messages window

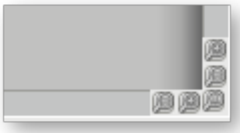
To move around the work area use the horizontal/vertical scroll bars at the bottom/right of the windows, or the mouse wheel to move up/down and `Shift`+mouse wheel to move left/right. You can also click on the navigation window (above the timeline) to jump to specific areas of the session.



The navigation window, situated between the main toolbar icons and timeline

To zoom in/out horizontally and vertically use the magnifying glass buttons in the bottom-right of the main view. `Shift`-clicking these allows you to zoom in greater increments and `Ctrl`-

clicking will zoom to the maximum/minimum. You can also zoom in/out horizontally using `Ctrl`+mouse wheel.

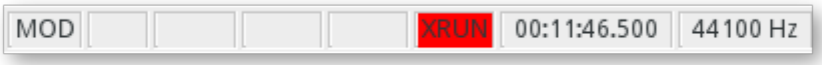


"Magnifying glass" zoom buttons, situated in the bottom-right of the main view

The clips area in the middle is where most of the action takes place. It contains visual displays and representations of audio waveforms or MIDI data. This section is used for editing Clip objects (portions or all of any particular audio or MIDI file, recorded or imported) and for navigation within the project or "session."

The very bottom-right of the screen shows the *Session total time* and *Session sample rate*. To the left of these is a panel which provides information on the session's current state. From left to right, the indicators and their meanings are:

- **MOD** - session state has been modified since last save
- **REC** - session is set to record when transport next starts
- **MUTE** - at least one track is muted
- **SOLO** - at least one track is soloed
- **LOOP** - session is set to loop when transport next starts
- **XRUN** - an x-run has occurred



MOD | | | | | XRUN | 00:11:46.500 | 44100 Hz

Session state, total time and sample rate information

Qtractor also has other useful windows, such as the Mixer window and the Connections (patch-bay) window. These can be opened via the **View** / **Windows** menu, or by pressing `F8` for Connections or `F9` for the Mixer. Three utility windows are additionally featured: the *Messages* window, for general information and debugging purposes; the *Files* pane, where audio and MIDI files are organised and selected; and the *File System* browser, which provides access to your file system in a manner similar to a file manager (such as Dolphin, GNOME Files, Thunar and so on).

Dialog windows for editing *session*, *track* and *clip* properties are also accessible in their proper context, which will be discussed in their respective sections.

Finally, session and application configuration options are assisted through respective customizing dialogs: *Buses*, *Instruments* and *Options…*, available from the **View** menu in the main menu bar.

4.3. Understanding a Qtractor Session (recording or editing)

A Qtractor session project contains all the information about your Clips and their placement, Mixer set-up, plugins, tempo, time signature and Connections patch-bay. When creating or saving a project, all this and any related settings are saved on your hard disk within this session project file.

To create a new session, simply start Qtractor and begin working. If you already have a session loaded and want to start a new one, choose **File** / **New** or click the *New session* icon (blank piece of paper) in the upper-left of the screen. To open a session, either use the **File** / **Open…** menu command or drag your session file (**.qtr** or **.qts**, which are essentially the same) into the left or middle pane.

4.3.1. Session Audio Sample Rate

It is important to note that Qtractor sessions are locked to a session project sample rate. This is dependent on the sample rate of the JACK server running at the time the session is created.

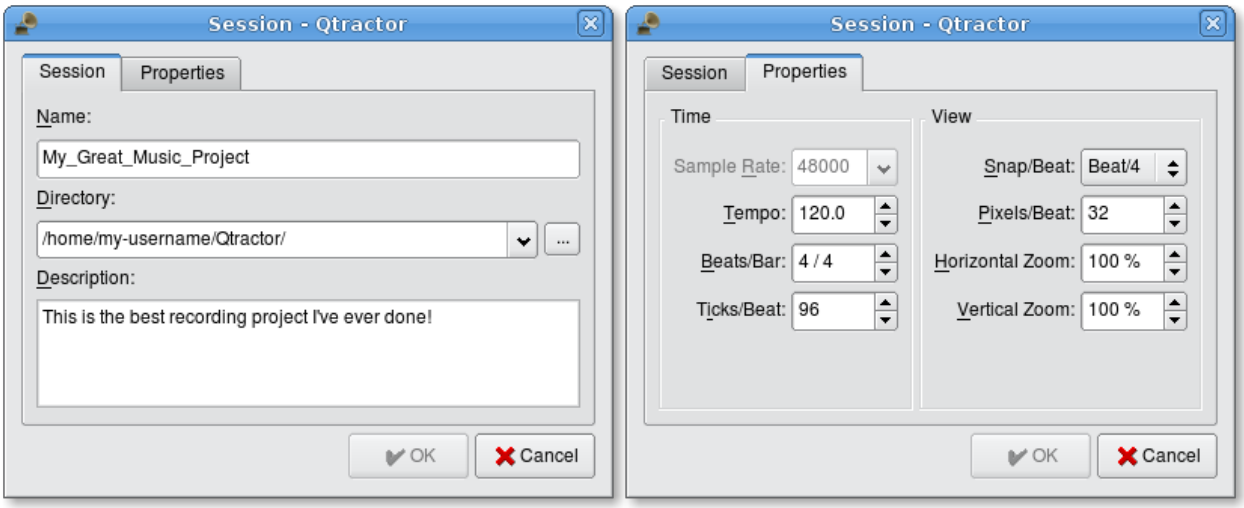
Any attempt to convert non-matching sample-rate sessions will result in a recommendation warning message.

However, individual audio clip files are automatically converted on playback in real-time to the host sample-rate (via libsamplerate). This method, while it works very well, is not the recommended method due to possible errors in the real-time sample rate conversion. Real-time sample rate conversion is also going to use quite a bit more valuable CPU resources.

Rui Nuno Capela is working toward eliminating this shortcoming by taking control of JACK from within Qtractor, and restarting it using the session's project parameters. This will ultimately reconnect any plugins, set the proper sample rate, etc. Until this feature is available, please follow the recommendations listed above.

4.3.2. Session Properties

To access the session properties, choose the **File** / **Properties** menu item. Here, you can name your Qtractor session, set the tempo, time signature (how many beats per bar) and decide how many "ticks" (the smallest time unit in a session) will be within the time span of each beat. If you choose to rename your session, Qtractor will update any midi and audio clip file names to match this.



Name: My_Great_Music_Project

Directory: /home/my-username/Qtractor/

Description: This is the best recording project I've ever done!

Sample Rate: 48000

Tempo: 120.0

Beats/Bar: 4 / 4

Ticks/Beat: 96

Snap/Beat: Beat/4

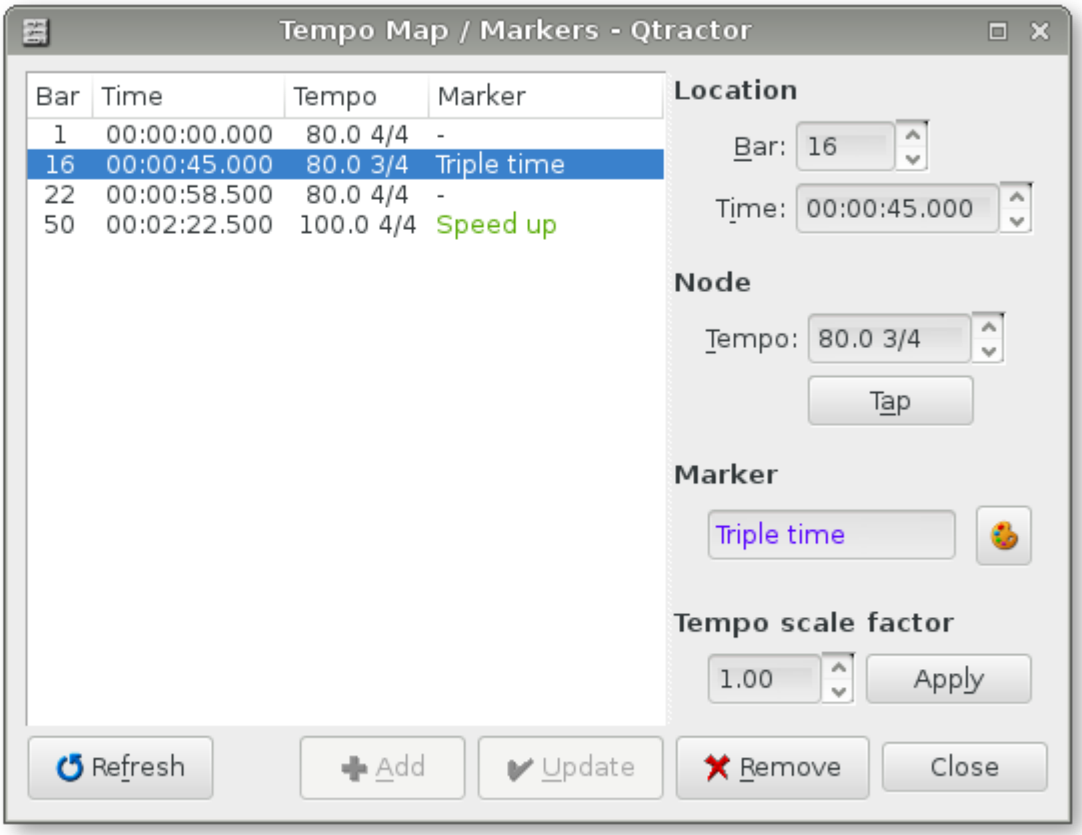
Pixels/Beat: 32

Horizontal Zoom: 100 %

Vertical Zoom: 100 %

Session properties window showing both Session and Properties tabs

Tempo and time signature can be changed during the course of your composition using the **Tempo Map / Markers** window. This is accessed via the **View** menu or by double-clicking the region directly above the tracks which displays the bar numbers. Determine where you wish your tempo/time signature change to occur in **Location**, input the details of the change in **Node**, then click either **Add** to add a new node, or **Update** to make changes to a previously created one. If desired, you can also add descriptive text in **Marker**, plus a colour, by clicking the palette icon next to the text field. Markers may be assigned to tempo/time signature changes, or independent of them. Once you're happy with the content of your Marker, click **Add** to add it to the map. The other option in this window is **Tempo scale factor**, which allows you to scale the entire Tempo Map (speeding up/slowing down the whole composition) by an arbitrary factor.



Bar	Time	Tempo	Marker
1	00:00:00.000	80.0 4/4	-
16	00:00:45.000	80.0 3/4	Triple time
22	00:00:58.500	80.0 4/4	-
50	00:02:22.500	100.0 4/4	Speed up

Bar: 16

Time: 00:00:45.000

Tempo: 80.0 3/4

Triple time

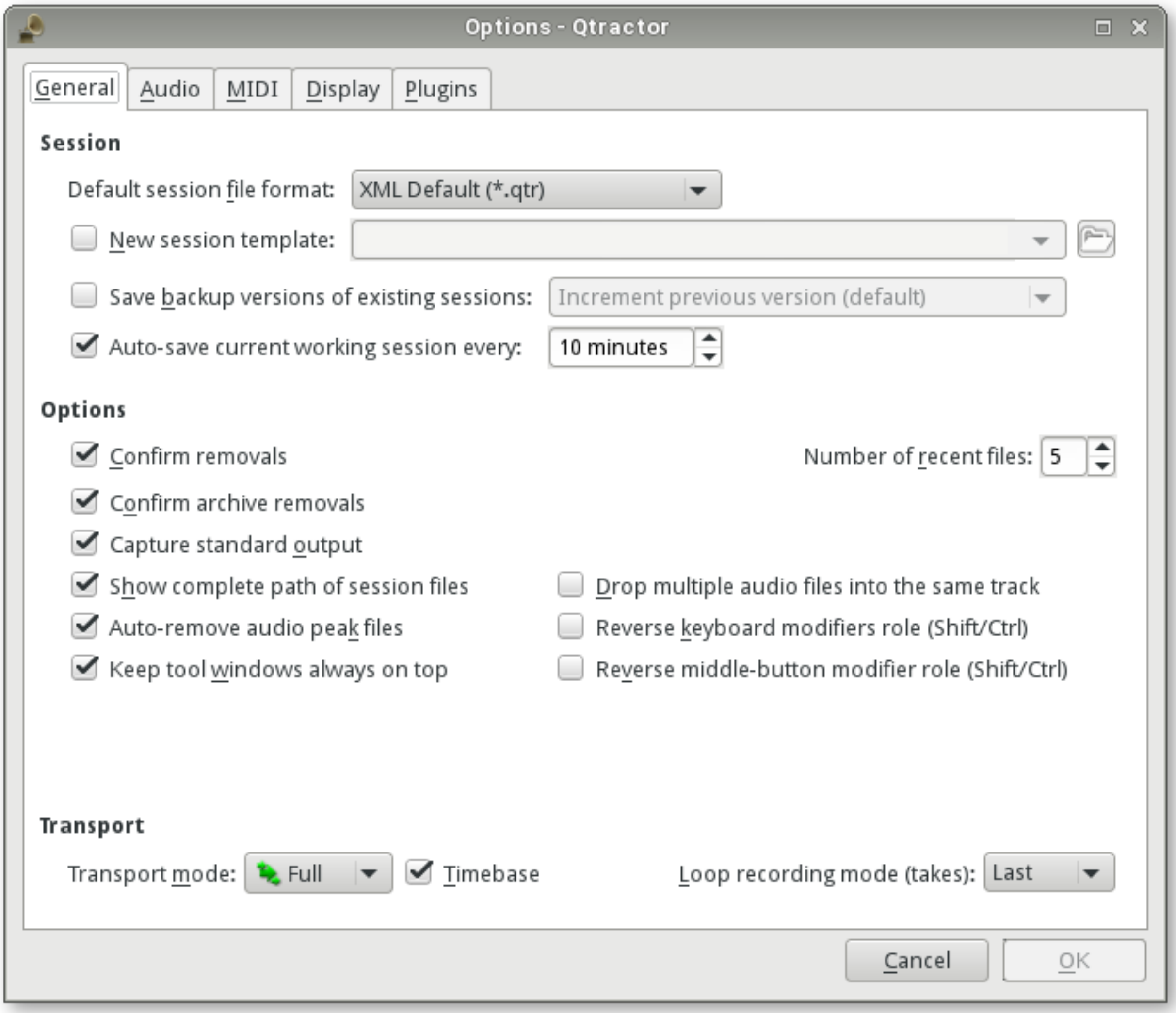
1.00

Tempo Map / Markers window showing tempo/time signature changes and markers

There are two limitations to be aware of with regard to tempo and time signature changes. Firstly, changes must occur at the start of a bar - there is no facility to add changes within one. If you, for example, need to insert a change into the middle of a 4/4 bar, create two 2/4 bars and add the change to the start of the second of these. Secondly, there is no facility to "ramp" between two different tempos (gradually increase/decrease tempo over a defined period). Hopefully, such a feature will be added in the future, though whether this will happen or not is currently unclear.

4.3.3. Session Options

Access the Options windows via the **View** / **Options…** menu item. This window allows you to control the global parameters of Qtractor. These are global settings which are not saved within your session file.



Session

Default session file format: XML Default (*.qtr)

□ New session template:

□ Save backup versions of existing sessions: Increment previous version (default)

☑ Auto-save current working session every: 10 minutes

Options

☑ Confirm removals

Number of recent files: 5

☑ Confirm archive removals

☑ Capture standard output

☑ Show complete path of session files

□ Drop multiple audio files into the same track

☑ Auto-remove audio peak files

□ Reverse keyboard modifiers role (Shift/Ctrl)

☑ Keep tool windows always on top

□ Reverse middle-button modifier role (Shift/Ctrl)

Transport

Transport mode: Full

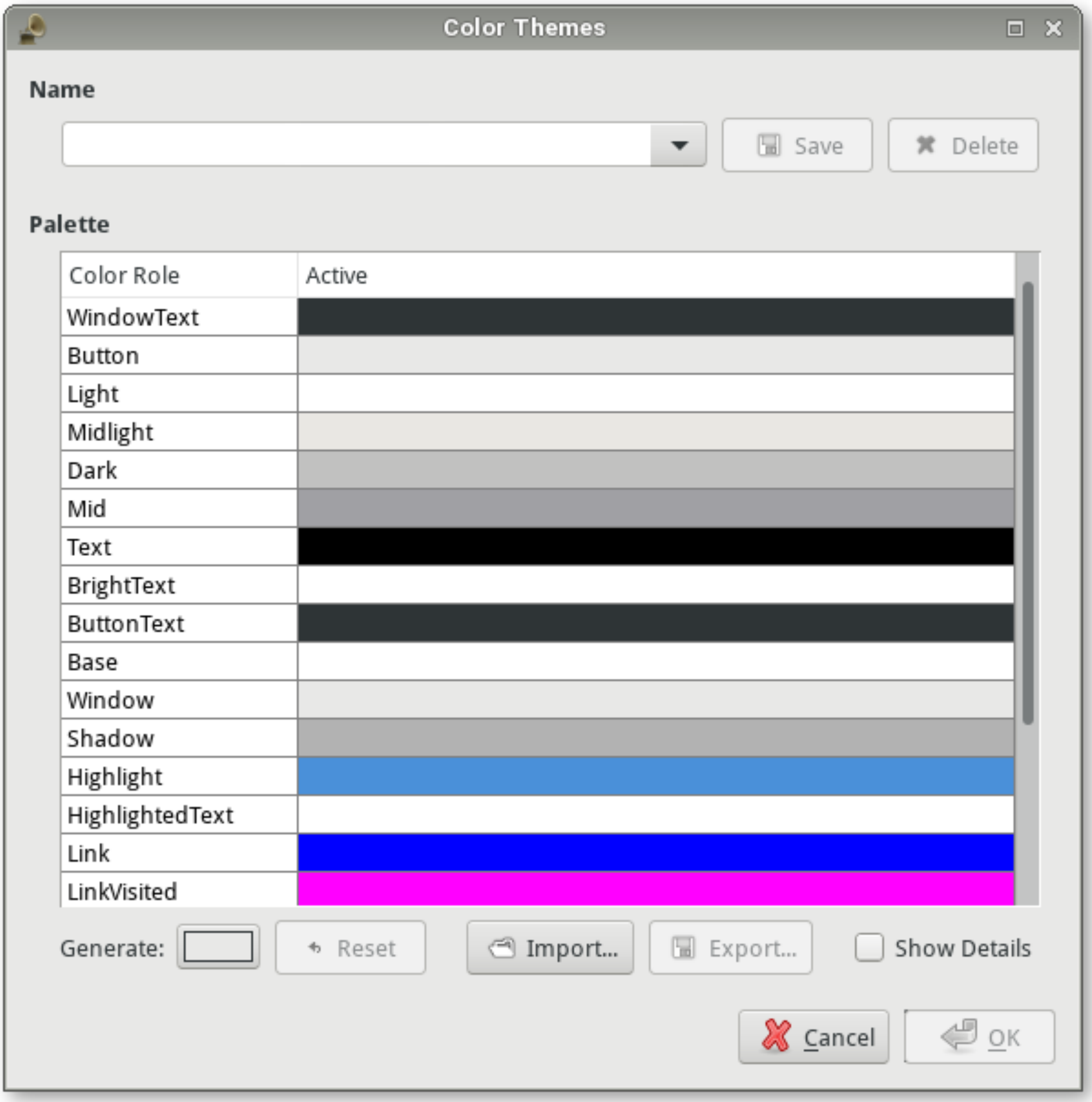
☑ Timebase

Loop recording mode (takes): Last

Options window, showing General tab

The window is divided into five tabs: **General**, **Audio**, **MIDI**, **Display** and **Plugins**. Most of the settings are self-explanatory, and there are tool tips to guide you, but some which require more explanation are covered elsewhere in this chapter. The default options should largely be fine to begin with, but if you're having trouble seeing parts of the GUI you may wish to adjust the options in the **Display** tab. Qtractor takes its look-and-feel from your system settings by default, but on this tab you can disable *Use desktop environment native dialogs* (which should be off by default) and choose your preferred *Color theme* and *Style theme*. You may also wish to run `qtconfig` to perform configuration for all Qt applications (of which Qtractor is one).

If you need to make detailed colour-related changes, click the **…** button to the right of *Color theme* to open the **Color Themes** window. Here, you can fine-tune a variety of settings, plus import them from/export them to external files via the **Import** and **Export** buttons at the bottom of the window.



☐ Show Details

Color Role	Active
WindowText	⬛
Button	⬜
Light	⬜
Midlight	⬜
Dark	▨
Mid	▨
Text	⬛
BrightText	⬜
ButtonText	⬛
Base	⬜
Window	⬜
Shadow	▨
Highlight	🟦
HighlightedText	⬜
Link	🟦
LinkVisited	🟪

Color Themes window, with the Show Details option enabled

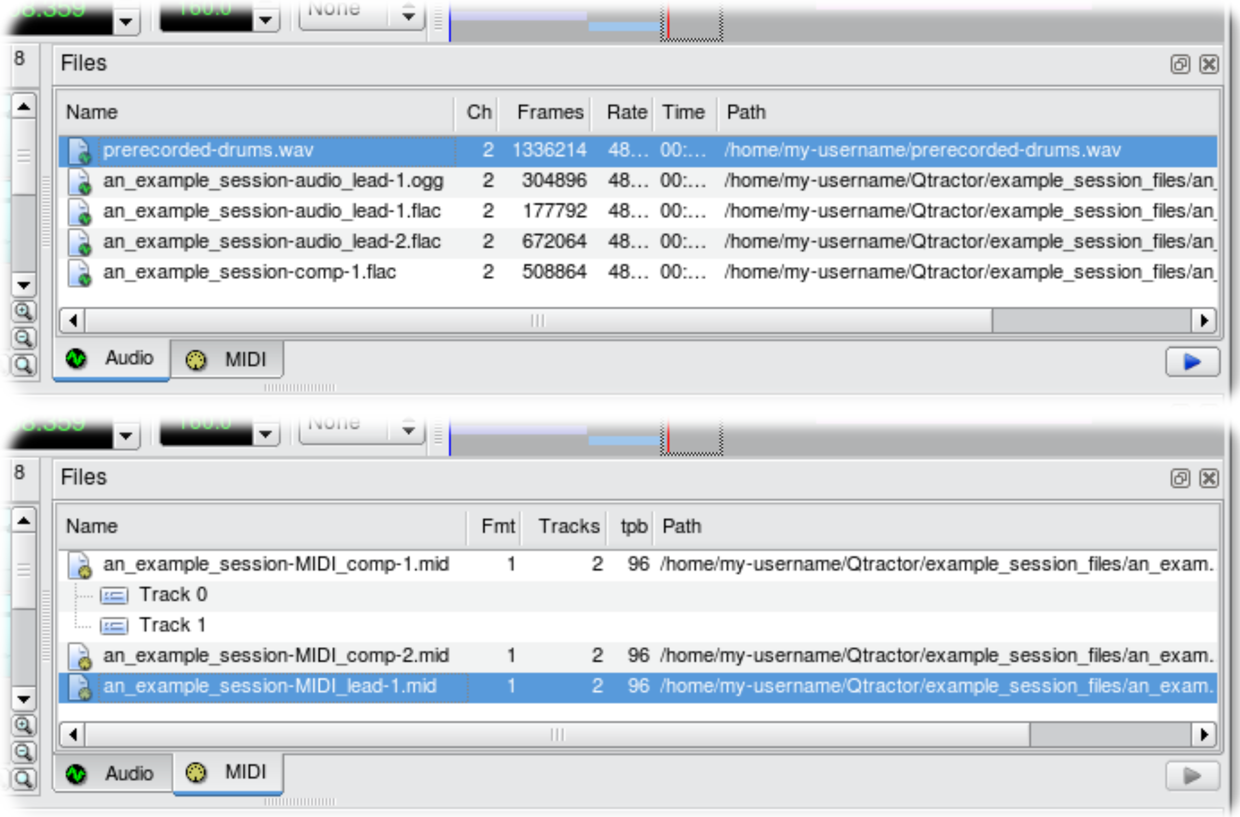
Finally, if you experience problems with Qtractor when using a desktop environment based on GTK+ (GNOME, Xfce etc.), try a Qt-based theme for *Style theme* and make sure that *Use desktop environment native dialogs* is disabled.

4.4. Files

Sound file selection is made available through a tabbed mini-organizer, known as the *Files* pane, which lists all the files contained in the current session. Audio and MIDI file lists are kept separate on their respective tabs. Audio file format support is provided by libsndfile, enabling the use of wav, aiff, flac, au, etc. The optional libraries libvorbis and libmad provide support for ogg and mp3 formats respectively. MIDI file support covers the usual SMF formats 0 and 1, through a native, home-brew implementation.

Files can be imported using the menu command **Track** / **Import Tracks** / **Audio…** or **MIDI…**, or by right-clicking in the relevant tab of the Files pane and choosing **Add Files…**. If the Files pane is not visible, you can enable it via **View** / **Windows** / **Files**. The default position for the

Files pane is to the right of the clips area, but it can be detached and docked elsewhere (such as to the left of the Tracks pane) if needed. You can preview files by double-clicking them, or by selecting them and either clicking the play button on the lower right hand side of the pane or choosing **Play** from the right-click menu.



Files

Name	Ch	Frames	Rate	Time	Path
prerecorded-drums.wav	2	1336214	48…	00:…	/home/my-username/prerecorded-drums.wav
an_example_session-audio_lead-1.ogg	2	304896	48…	00:…	/home/my-username/Qtractor/example_session_files/an_
an_example_session-audio_lead-1.flac	2	177792	48…	00:…	/home/my-username/Qtractor/example_session_files/an_
an_example_session-audio_lead-2.flac	2	672064	48…	00:…	/home/my-username/Qtractor/example_session_files/an_
an_example_session-comp-1.flac	2	508864	48…	00:…	/home/my-username/Qtractor/example_session_files/an_

Files

Name	Fmt	Tracks	tpb	Path
an_example_session-MIDI_comp-1.mid	1	2	96	/home/my-username/Qtractor/example_session_files/an_exam.
Track 0				
Track 1				
an_example_session-MIDI_comp-2.mid	1	2	96	/home/my-username/Qtractor/example_session_files/an_exam.
an_example_session-MIDI_lead-1.mid	1	2	96	/home/my-username/Qtractor/example_session_files/an_exam.

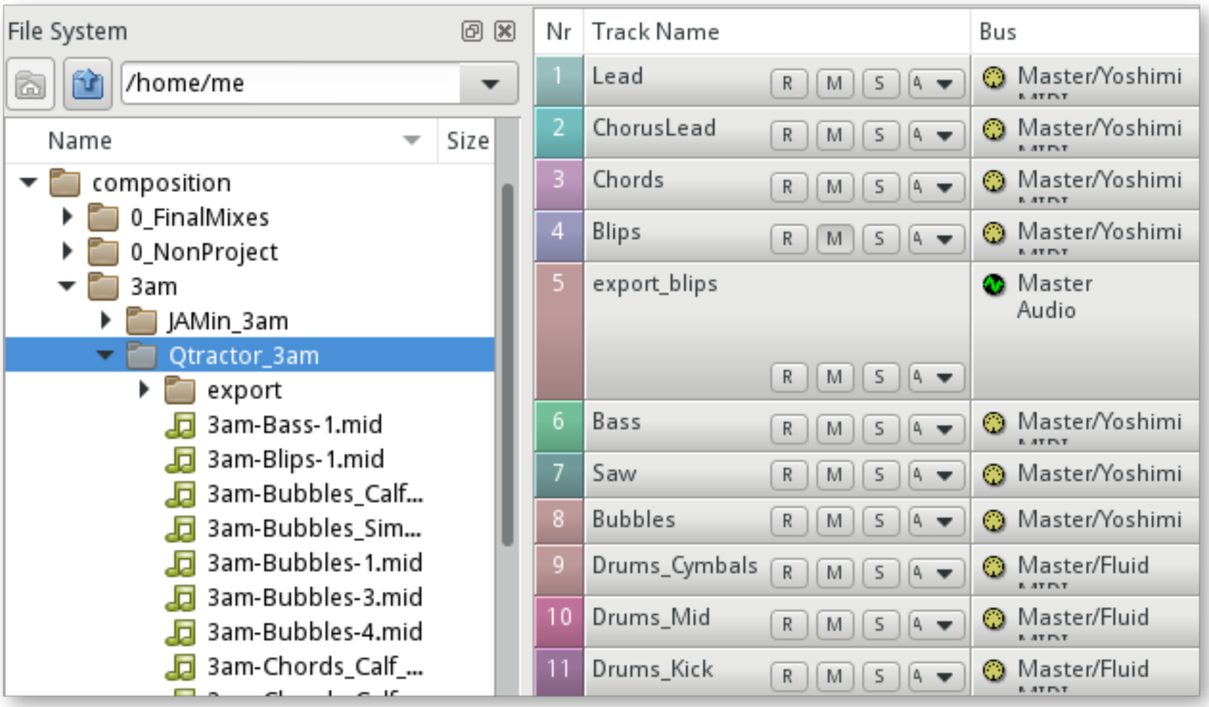
The Files pane of Qtractor's main workspace. Both Audio and MIDI tabs are shown

When importing a MIDI file, note that the file's tempo map will be reproduced in Qtractor *only* if it is the first MIDI content to be loaded into the current session. If the current session already has one or more MIDI tracks (even if the tracks contain no actual clips), the tempo map from the imported MIDI file will be discarded. This also means that if a **New session template** containing tracks has been enabled via the **View** / **Options…** / **General** menu, the tempo map of *any* imported MIDI file will be automatically discarded.

Individual and multiple files can be drag-and-dropped from the desktop environment and within the file list. Dropping a file into a blank region of the tracks area will create a new track and add the clip (which represents the file) to it. Dropping a file into an existing track will create a new clip within it. When dropping an audio file into an existing track you can drag the file itself, but with a MIDI clip you must drag your desired Track/Channel (displayed when clicking the file name in the Files pane), rather than the file itself. Finally, dragging a session file (**.qtr** or **.qts**, which are essentially the same) into the left or middle pane will load that session.

After working on a project for a while you'll probably find that the Files pane contains files which you no longer need. To remove anything not currently referenced in the tracks area, right-click the Files pane and choose **Cleanup**. Note that this merely cleans up the *list* of files - it does not delete the files themselves from your hard drive. If you wish also to remove non-needed files from your session directory, save the session as a **.qtz** archive and use this in place of your previous session. See here for details on archives.

For convenience, a File System browser with tree-view is also available. This can be enabled via **View** / **Windows** / **File System**. Its default position is to the left of the Tracks pane, but like the Files pane it can be detached and docked elsewhere. For instance, you may choose to have both windows on one side of the screen, with one above the other.



- composition
  - 0_FinalMixes
  - 0_NonProject
  - 3am
    - JAMin_3am
    - Qtractor_3am
      - export
      - 3am-Bass-1.mid
      - 3am-Blips-1.mid
      - 3am-Bubbles_Calf...
      - 3am-Bubbles_Sim...
      - 3am-Bubbles-1.mid
      - 3am-Bubbles-3.mid
      - 3am-Bubbles-4.mid
      - 3am-Chords_Calf_...

Nr	Track Name	Bus
1	Lead	Master/Yoshimi
2	ChorusLead	Master/Yoshimi
3	Chords	Master/Yoshimi
4	Blips	Master/Yoshimi
5	export_blips	Master Audio
6	Bass	Master/Yoshimi
7	Saw	Master/Yoshimi
8	Bubbles	Master/Yoshimi
9	Drums_Cymbals	Master/Fluid
10	Drums_Mid	Master/Fluid
11	Drums_Kick	Master/Fluid

File System browser, docked to the left of the Tracks pane

4.5. Clips

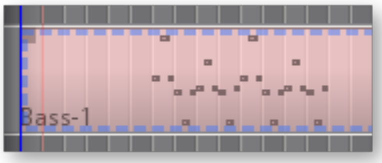
4.5.1. Clips Summary

Clip objects are the elemental items of a session arrangement, and can contain either audio or MIDI data. A clip represents either a region of, or the entirety of, an actual audio or MIDI file. MIDI clip objects are representations of a sequence of events of one single MIDI channel, as extracted from an SMF format 0 file, or of one single track from an SMF format 1 file. Clips can be copied, truncated, lengthened/shortened (by dragging their edges) and time-stretched as if they were actual audio/MIDI data; editing a clip object in these ways is a non-destructive process.

Clips are placed on tracks either by importing audio or MIDI files as new tracks or by dragging and dropping files into the main track window. Empty clips may also be created by right-clicking on a track and choosing **Edit** / **Clip** / **New…**.

4.5.2. Copying, Cutting, Splitting and Merging

To *copy* or *cut* a clip, select the clip, then right-click and choose **Copy** or **Cut** from the menu. Next, choose **Paste** from the right-click menu, position the box outline where you want to paste the clip and left-click. Note that when copying a MIDI clip within the same track, the two clips become *linked*, meaning that any change to one of them will automatically affect the other. If you select a clip which is currently linked to another, any such linked clips will be highlighted with a dashed outline, enabling you to see which other clips will be affected by changes made to the selected one. To unlink a clip, thus disabling automatic changes, select it and choose **Clip** / **Unlink** from the right-click menu.



A linked clip highlighted with a dashed outline

To *split* a clip, select the clip, then position the playhead (red vertical line) where you want to split it and choose **Clip** / **Edit** / **Split** from the right-click menu. If you use this operation often, it is far more convenient to assign a keyboard shortcut for it. Since Qtractor never affects the original file it is referencing, any split you make to a clip is reversible. You can undo a split via the **Edit** / **Undo** menu, or by clicking and dragging the edge of the clip then extending it to its original length. Note that when dragging audio files to hide/reveal portions of them, the waveform may appear to "move" slightly. This is just a visual artefact - the position of the actual audio in the track is not affected.

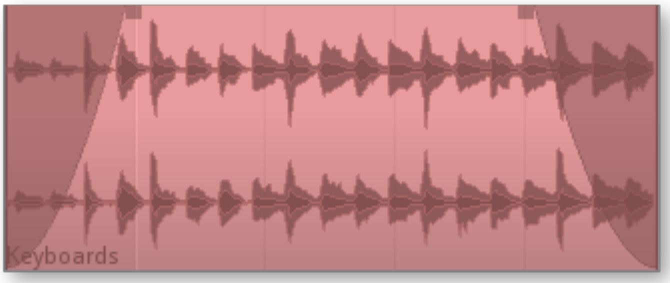
To *merge* clips and so consolidate a number of them into one file on your hard drive:

1. Select the clips within a track that you want to merge into one
2. Choose **Clip** / **Merge**, either via the main toolbar or the right-click menu
3. In the **Merge/Export** window, enter a new name for the merged file and save it to a logical location on your hard drive (preferably within your session folder)
4. The merged clip immediately replaces the old clips in your project

4.5.3. Fades and Cross-Fades

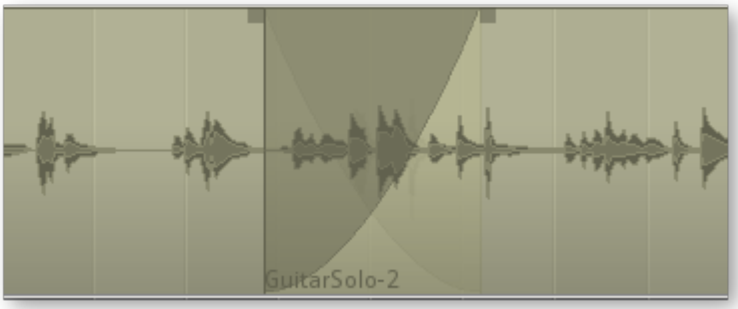
Fading a clip in or out is a common enough operation that there is a quick shortcut to achieve the effect. Notice that in the top corner of any audio or MIDI clip there is a semi-transparent square node. Clicking and dragging this node further into the clip will create a fade-in (if done at the beginning of the clip) or a fade-out (if done at the end).

The actual method of audio volume (gain) and MIDI velocity change can be one of linear, quadratic or cubic types. You can set your desired type via the clip's **Properties** window (see next section).



An audio clip, with left and right upper corner fade handles drawn inward to create a fade-in and fade-out

When two clips overlap you may wish to cross-fade them, simultaneously fading-out the first and fading-in the second. You can do this manually, by dragging the fade handles to the desired lengths (they can still be manipulated even if clips overlap). Alternatively, you can select either clip and choose **Clip** / **Cross Fade** - Qtractor will then add a cross-fade, determined by the length of the overlap. Once you have your cross-fade you can fine-tune its length and type if needed, as per the information above. If you have trouble clicking on a particular clip because it's obscured by another, try selecting it from the Files pane.



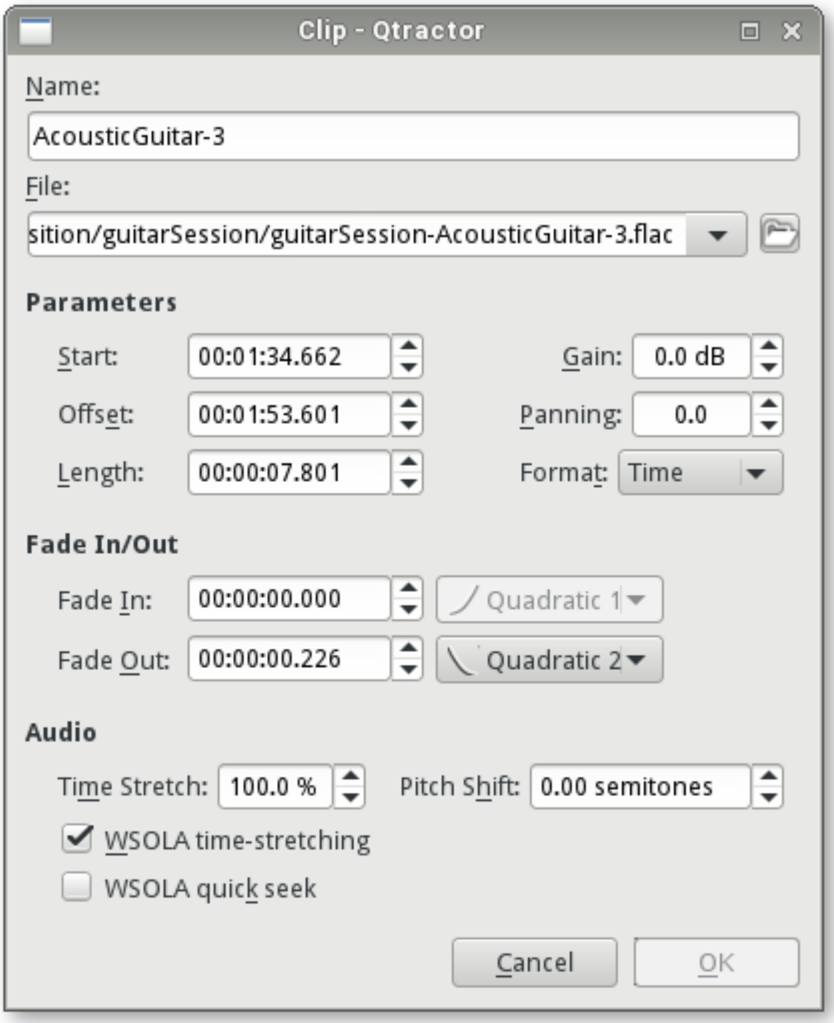
Two audio clips cross-faded together

4.5.4. Clip Properties

A clip's properties can be accessed by double-clicking the clip (for audio clips), or by opening the clip and choosing **File** / **Properties** (for MIDI).

Audio clip properties include its Name (the label the clip is known by within the session); File path; Start time (location), Offset and Length; Gain; Panning; Format (Frames, Time or BBT); Fade-in/out length and type; Time Stretch percentage (can be set within a range of 10%-1000%) and options; and Pitch Shift amount (can be set within a range of -/+40 semitones). Time-stretching is applied in real-time at the buffering level, as a custom WSOLA algorithm based on and derived from the SoundTouch library.

MIDI clip properties include its Name; File path; Track/Channel; Start time, Offset and Length; Volume; Format (Frames, Time or BBT); and Fade-in/out length and type.



Clip - Qtractor

Name: AcousticGuitar-3

File: sition/guitarSession/guitarSession-AcousticGuitar-3.flac

Parameters

Start: 00:01:34.662

Offset: 00:01:53.601

Length: 00:00:07.801

Gain: 0.0 dB

Panning: 0.0

Format: Time

Fade In/Out

Fade In: 00:00:00.000 Quadratic 1

Fade Out: 00:00:00.226 Quadratic 2

Audio

Time Stretch: 100.0 %

Pitch Shift: 0.00 semitones

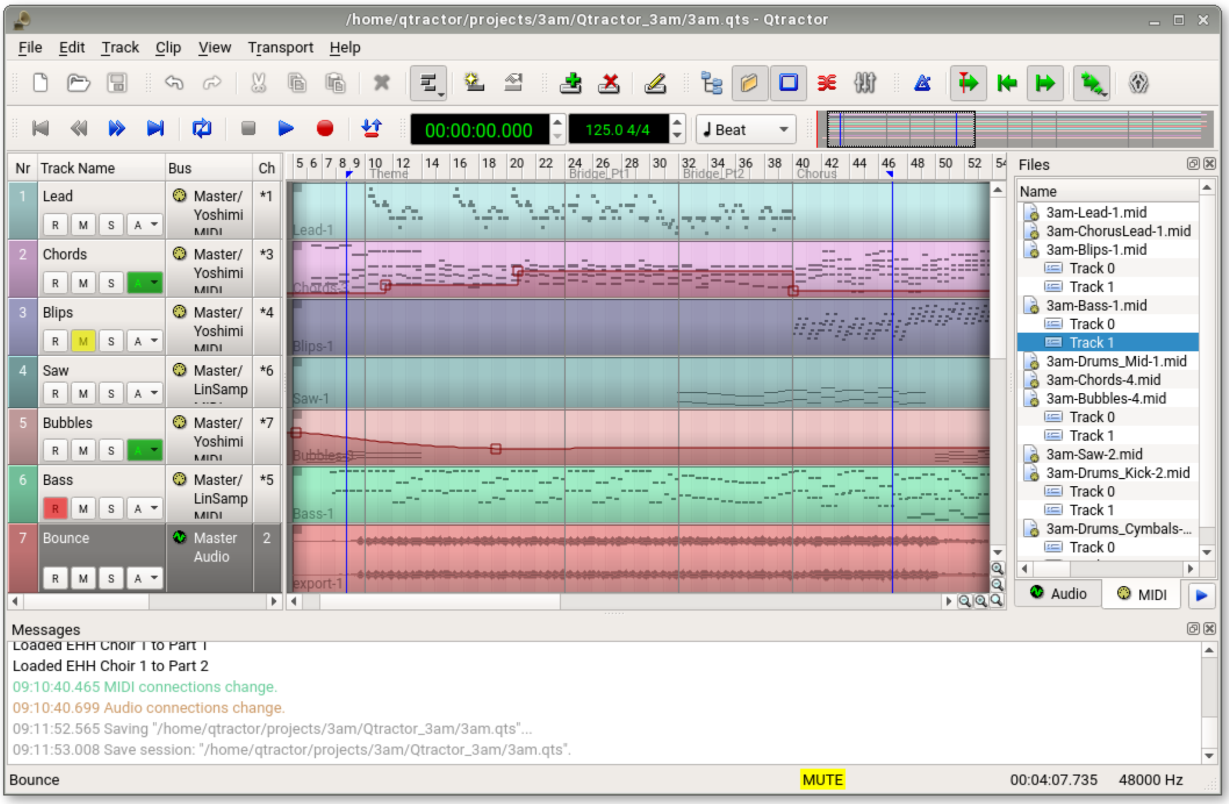
☑ WSOLA time-stretching

☐ WSOLA quick seek

Audio clip properties window

4.6. Qtractor Main Workspace–Tracks Area

4.6.1. Tracks Summary

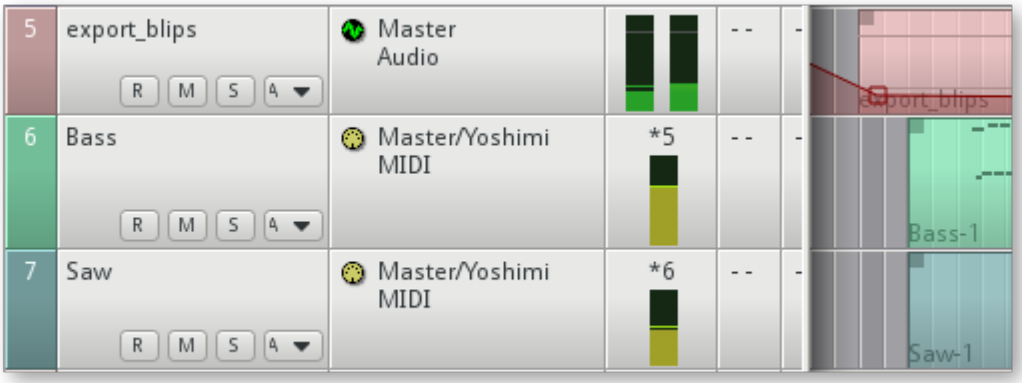


```
Loaded EHH Choir 1 to Part 1
Loaded EHH Choir 1 to Part 2
09:10:40.465 MIDI connections change.
09:10:40.699 Audio connections change.
09:11:52.565 Saving "/home/qtractor/projects/3am/Qtractor_3am/3am.qts"...
09:11:53.008 Save session: "/home/qtractor/projects/3am/Qtractor_3am/3am.qts".
```

MIDI and audio tracks in the main workspace of Qtractor

Tracks are arranged as a sequence of one or more clips of the same file type, either audio or MIDI. The tracks window is the main application workspace, serving as a virtual canvas of a multi-track composition arranger. Most of the editing operations are performed here.

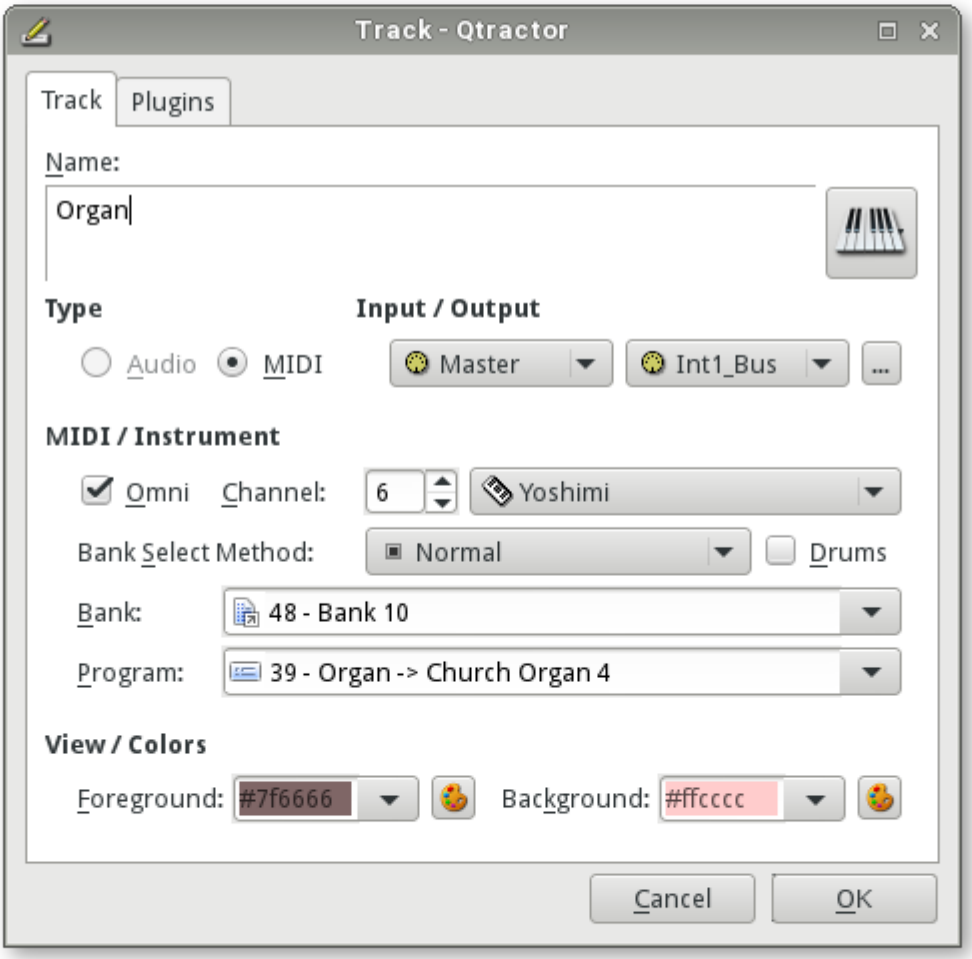
The tracks area has two panes: the left one displays the list of tracks with their respective properties and the middle pane is the audio and MIDI clips area, where multi-track composition and arranging is performed. Column widths in the left pane can be adjusted to allow for long track names etc. by dragging their edges; double-clicking resets them to their defaults. Track height can be adjusted by dragging up/down in the **Nr** column of the left pane, or by right-clicking and using the **Height** menu.



Audio and MIDI tracks with meters enabled

Note that information regarding global UI settings, such as column widths and Mixer set-up, is stored in `~/.config/rncbc.org/Qtractor.conf` (and so restored upon re-loading Qtractor), while track height is stored in the session file (and so restored upon re-loading a session).

To create a new track either right-click in the left pane and choose **Add Track**, or right-click in the right pane and choose **Track** / **Add Track**. In the **Track** window which appears, choose a name in the **Name:** box, choose a **Type** (*Audio* or *MIDI*), adjust any other options desired, such as the track's colour and icon, and whether it is for drums (which will show MIDI Note On events as diamonds instead of rectangles) or not, then click **OK**.



Name: Organ

Type: ○ Audio ● MIDI

Input / Output: Master | Int1_Bus | ...

MIDI / Instrument

☑ Omni Channel: 6 | Yoshimi

Bank Select Method: Normal □ Drums

Bank: 48 - Bank 10

Program: 39 - Organ -> Church Organ 4

View / Colors

Foreground: #7f6666 Background: #ffcccc

Track window for a MIDI track

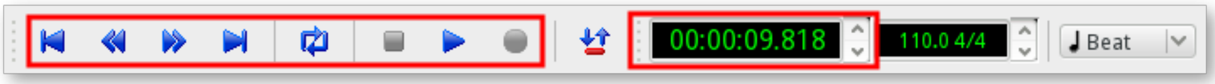
You can also duplicate an already-existing track by right-clicking the track in the left pane and choosing **Duplicate Track**.

As is usual, tracks are stacked on horizontal strips and clips are layered on a bi-dimensional grid, in time sequence for each track strip. Time is modeled on the horizontal axis and pictured by a bar-beat scale ruler at the top of the track view. Clips may be conveniently aligned to discrete time positions, depending on the current *Snap* mode setting, which is set via either its icon on the toolbar or the **View** / **Snap** menu. When set to "None", snapping is carried out to MIDI resolution, quantized to ticks per quarter note granularity.

Each track has its own user assignable colors for better visual identification. Audio clips are displayed with approximate waveform graphic, with peak and RMS signal envelopes as read from the respective audio file segment. MIDI clips are shown as a *piano roll*-like graphic, with note events shown as small rectangles, depicting pitch, time and duration.

All session, track and clip editing operations are undo/redo-able. Discrete view zooming is also available.

The red vertical line in the tracks area is the *playhead*. This shows the point at which playback/ recording will begin when you start the transport rolling and its time corresponds to the clock towards the top-left of the screen. You can control the position of the playhead via the transport controls (to the left of the clock), by dragging the top of the playhead, or by `Ctrl`/`Shift`-clicking in the horizontal bar above the tracks which houses the bar numbers or in an area of the Tracks pane which contains no clip (`Ctrl`/`Shift`-clicking on a clip will select the clip). Alternatively, you can `Ctrl`/`Shift`-middle click on either the horizontal bar or anywhere in the Tracks pane (with or without a clip).



Transport controls (left box) and clock (right box) showing current position of playhead

The blue vertical lines are *Edit Markers*. See below for information on these.

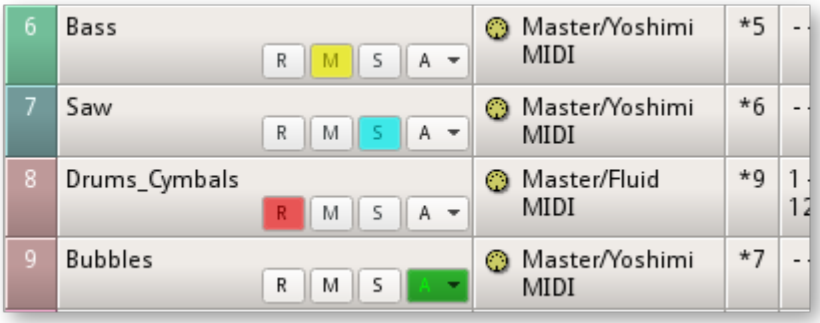
4.6.2. Track States

A track's state is controlled via the **R**, **M**, **S** and **A** buttons in the left pane. Their meanings are as follows:

- **R**ecord - arm the track for recording, ready for creating new audio or MIDI clips with captured material
- **M**ute - silence the track so that it produces no sound
- **S**olo - set only this track to produce sound, silencing all the others (though those already solo-ed will too produce sound)
- **A**utomation - automatically control various parameters of a track or its plugins. See the Automation section for details

When muting/solo-ing a track, holding down `Shift` or `Ctrl` while clicking the **M** or **S** buttons produces the following effects:

- `Shift`-click - Enable/disable state on all tracks. If the button clicked is currently disabled, all tracks will be enabled; if it is currently enabled, all tracks will be disabled
- `Ctrl`-click - Enable/disable state on this track only. For example, `Ctrl`-solo-ing a track with other tracks already solo-ed will un-solo those other tracks, leaving just the target track solo-ed



Mute, Solo, Record & Automation buttons for individual tracks

It is possible to select more than one track at a time in order to mute/solo/monitor them all simultaneously, by using the following commands:

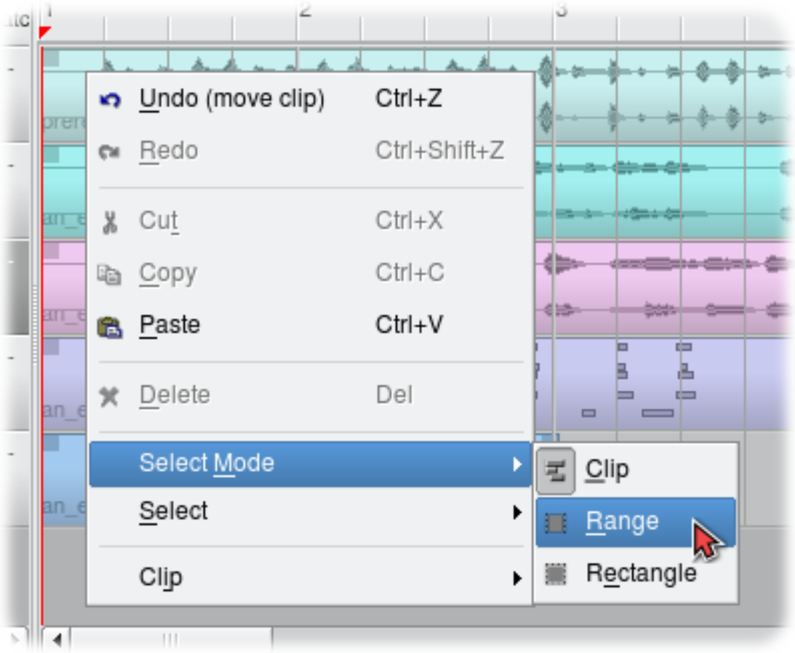
- Add tracks - `Shift`+left-click/mouse drag; `Shift`+Home/End keys; `Shift`/`Ctrl`+up/down arrow keys
- Toggle tracks - `Ctrl`+left-click/mouse drag; `Ctrl`+Home/End keys

In Qtractor parlance, though there is only ever one *current* track at any one time for editing purposes, there may be one or more *selected* tracks, invoked by the above commands, on which an operation can be performed. The *current* track is always displayed with a slightly darker background, whilst the colour of *selected* tracks is determined by the current palette scheme or theme (the default being that of the desktop environment). Once you have your desired tracks selected, you can mute/solo/monitor them using the relevant icons, menus or keyboard shortcuts. It is also possible to `Shift`/`Ctrl`-click the **M** or **S** buttons (as discussed above) with multiple tracks selected in order to apply these operations to the entire "set" of selected tracks.

Tracks can also be muted and solo-ed on mix-down, which will then apply when exporting, enabling you to export specific tracks only.

4.6.3. Making Selections - Select Modes

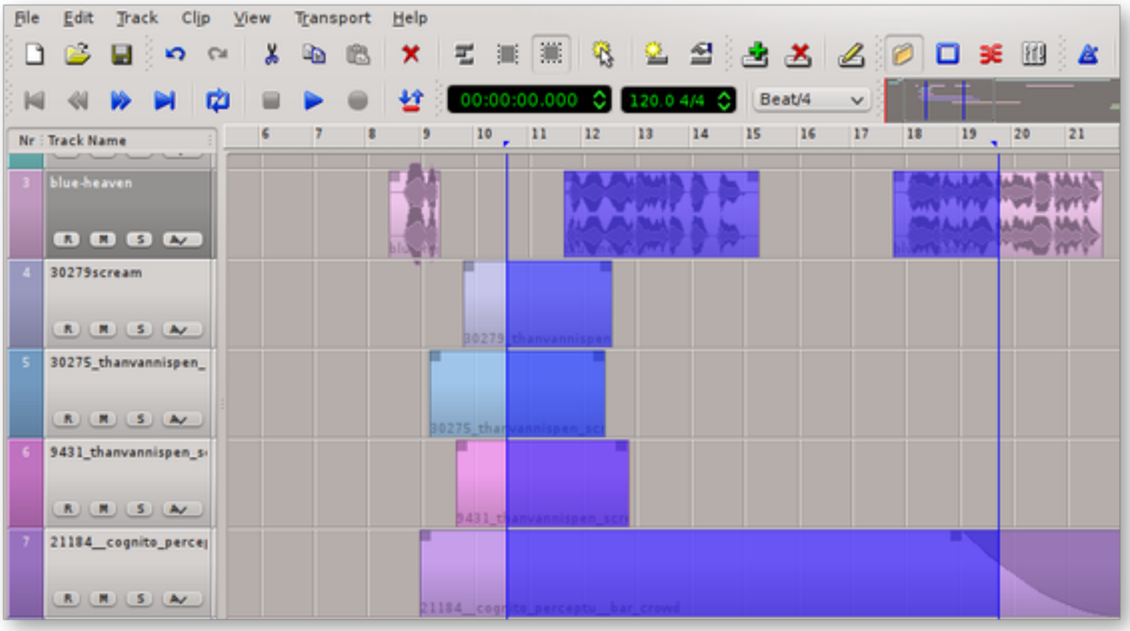
The mouse cursor has four *select modes*: *Clip*, *Range*, *Rectangle* and *Automation*. Choose your desired mode via **Edit** / **Select Mode** from the main toolbar, or by right-clicking in the track area and choosing **Select Mode**.



Select Mode context menu (right-click menu)

- **Clip** - allows you to select whole clips, click and drag them within and between tracks, and extend/shorten their lengths. The precision with which you perform these operations is determined by the current *Snap* mode setting
- **Range** - spans all tracks in your workspace and allows you arbitrarily to select any portion of clips, irrespective of their beginning/end. You can, for example, select a range in the very middle of a clip and all other clips above and below it, because you are selecting a block of time, rather than a particular clip
- **Rectangle** - is similar to *Range*, but is track-specific. This allows you to draw an arbitrary "box" area over one or more tracks, the extent of which becomes your selection

Once you've made your selection, you can then perform the usual cut, copy, paste etc. operations.

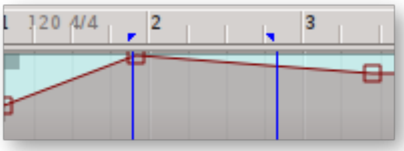


Making a selection using the Range* select mode*

For the final select mode, *Automation*, see its section.

4.6.4. Making Selections - Edit Markers

In addition to using select modes, you can select a range in your workspace by using the *Edit-head* and *Edit-tail* markers. These are the two blue vertical lines which are present in the tracks area and which are independent of the playhead.



The Edit-head and Edit-tail markers

To set the *in-point* of your selection, left-click in the bar/tempo area of the workspace, where the top of the blue lines are situated. To set the *out-point*, right-click in the same area. You can also drag the lines. Middle-clicking will bring both lines together at that point.

Once you've set your markers, you can use them to make a selection. To do so, from the main toolbar, choose **Edit** / **Select** and one of **Track** or **Track Range**. You can also right-click in the track area to access the same menu.

4.6.5. Loops

Looping a section is a common operation, often used to enable a musician to practise a particular passage, or a mix engineer to focus on a specific area. To play a section of your composition in a loop:

1. Select a range, using one of the methods described above

2. From the **Transport** menu, select **Loop Set**. Alternatively, click the *Loop* button (pair of blue, circular arrows) on the main toolbar

3. Move the transport into the loop range, or somewhere before it, then press the *Play* button or the space bar. Once the transport reaches the end of the loop range, it will return to the in-point and seamlessly continue playing until stopped

It is also possible to record using a loop, but such an operation would more commonly be handled by Qtractor's *Takes* feature. See its section for details.

4.6.6. Paste Repeat

In electronic music especially, there's often a need to record one or two bars and then loop them to create standardised bass lines or drum beats. For drum tracks, you may prefer to handle them by using a specialist drum machine, such as Hydrogen, which would play the same data (either pure MIDI data or audio samples) for as many bars as you program it to. However, the same effect can be achieved in Qtractor by using a *Paste Repeat*. This allows you to copy an audio or MIDI clip and then paste it back-to-back for as many repetitions as you wish.

There are two ways to perform a Paste Repeat - one quick and one slightly more detailed. The quick method is to `Ctrl`-click and drag a clip's left or right edge. This will replicate the clip as many times as fits into the desired area. The more detailed method is as follows:

1. Check your snapping settings in the **View** / **Snap** menu (when pasting clips for a loop, the user generally wants precision snapping to the clips' edges)

2. Select the clip you want to copy using the *Clip* select mode (or use *Range* mode, *Rectangle* mode, or the Edit Markers to select a portion of a clip) and use **Edit** / **Copy** to copy it

3. Click **Edit** and choose **Paste Repeat**

4. In the **Paste Repeat** window, enter the desired number of iterations in **Count** and, if needed, the duration (in frames, time or BBT) between the start of each iteration in **Period**. Click the **OK** button to confirm

5. Position the box outline where you want to paste the clips and left-click to anchor them in their track

Note that any clips copied in this way will become linked to the original, as covered in the Clips section.

4.6.7. Punch In/Out

If one or two bars of a track don't quite measure up to your standards, you might want to re-record just those bars without having to record the entire track over again. This process is called *punching in/out*.

To perform a punch in/out:

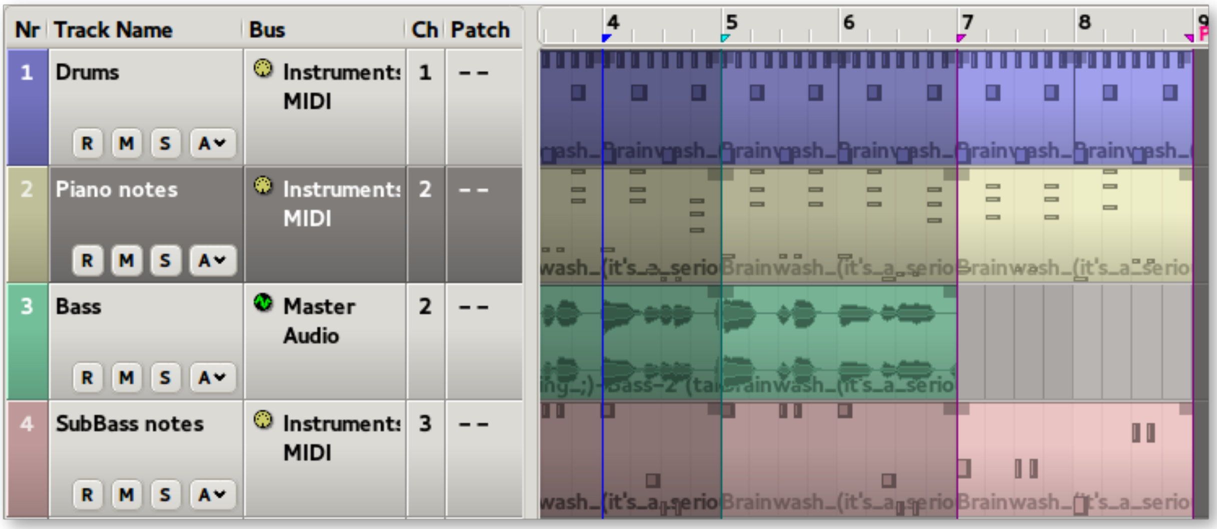
- Use the *Range* or *Rectangle* selection mode, or the Edit Markers, to define the region of your timeline that you wish to re-record

- Once your range is set, click the *Punch in/out* button in the main toolbar (pair of up/down arrows), or use the **Transport** menu and select **Punch Set**

- Arm the recording destination track by clicking its "R" button in the track properties pane, on the left hand side of the main workspace

- Press the *Record* button in the main toolbar, position the playhead somewhere before your selected region, then press the *Play* button or the space bar. Qtractor will play back your piece and you can play along, but nothing will actually be recorded unless it is within the *punch* range

- To stop playback, click the *Stop* button in the main toolbar, or use the space bar



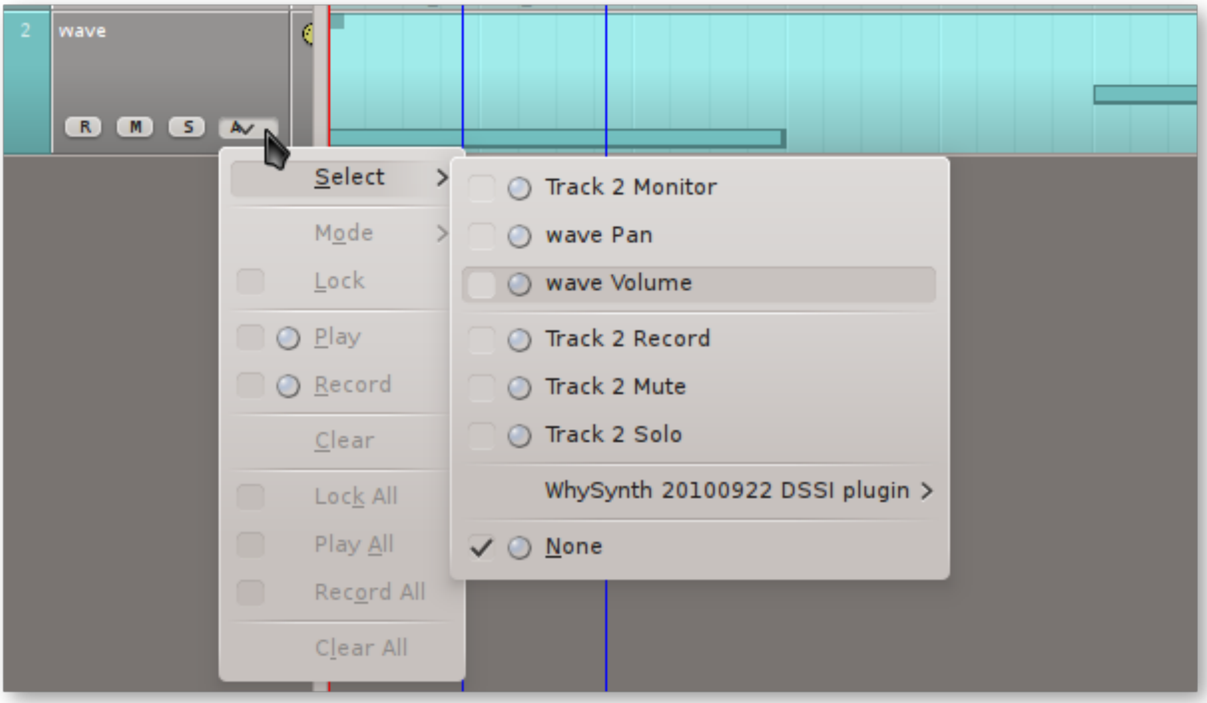
Nr	Track Name	Bus	Ch	Patch
1	Drums	Instruments MIDI	1	- -
2	Piano notes	Instruments MIDI	2	- -
3	Bass	Master Audio	2	- -
4	SubBass notes	Instruments MIDI	3	- -

Punch in/out range (purple markers) inside a loop range (green markers)

4.6.8. Automation

Qtractor features automation of nearly any aspect of any plugin, as well as basic track functions like volume and panning. It is only possible to automate tracks; buses cannot be automated.

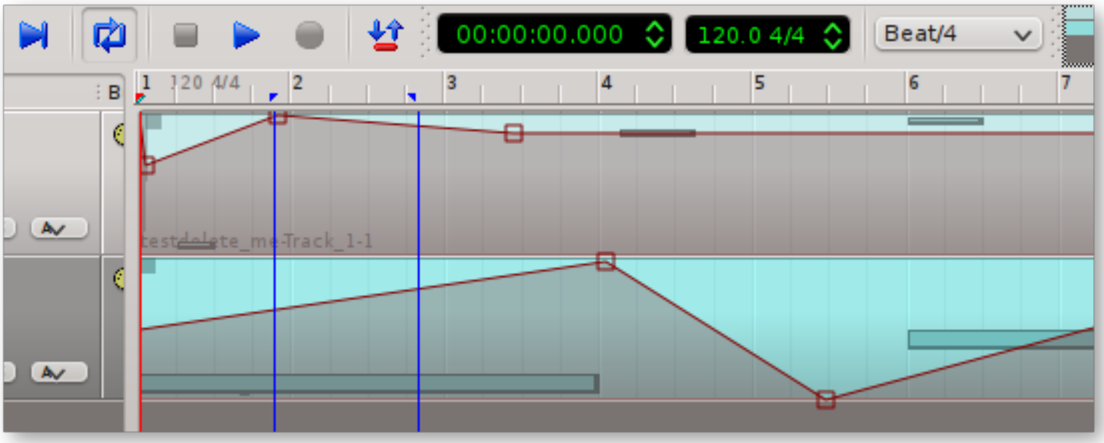
- In the track you wish to automate, click the *Automation* button in the track list. From the pop-up menu, select the attribute of the track that you wish to automate first (here, we're automating "wave Volume"). An overlay is placed on the track.



Automation pop-up menu

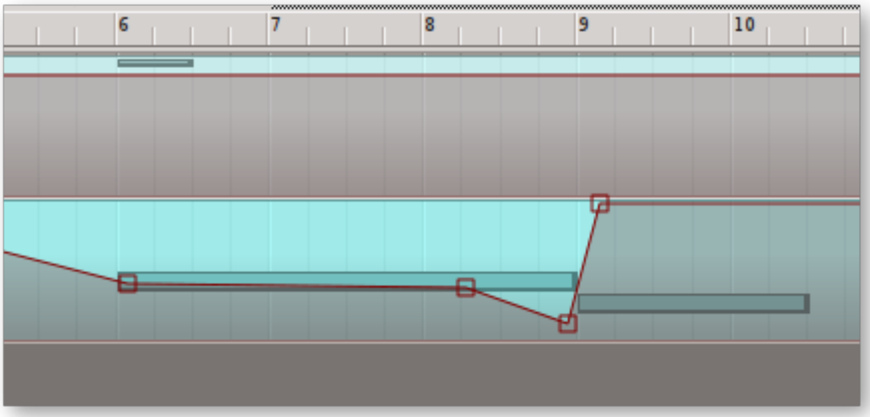
- Click the **Edit** menu to go into the **Select Mode** category, and choose the **Automation** selection tool. Alternately, click the *Automation Selection Tool* icon from the top menu bar

- Armed with your automation selection tool, click on the automation overlay on your track to create a node. Node placement is constrained by the *Snap/beat* setting, so set this to *None* if you wish to place a node at an arbitrary point
- Adjust the volume of this node to be the initial volume. You can either drag the node or double-click it and enter a precise value
- Click again on the automation overlay to create a second node, and move it to another volume level. For the space between the two nodes, the volume will change from the starting node level to the ending node level
- You can drag multiple nodes left/right, maintaining their values (vertical positions), by first selecting the nodes then dragging them whilst holding Ctrl or Shift. It is not currently possible to drag multiple nodes up/down



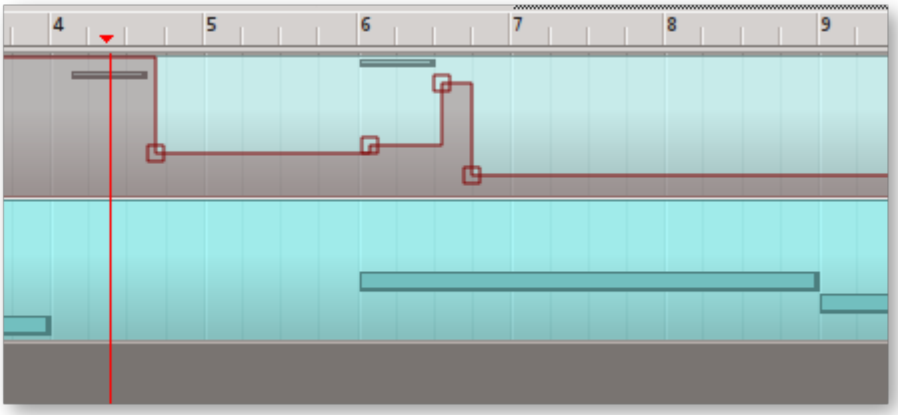
Adding automation nodes

If you need a less gradual change, move the nodes closer together, or use more nodes to shape different movements.



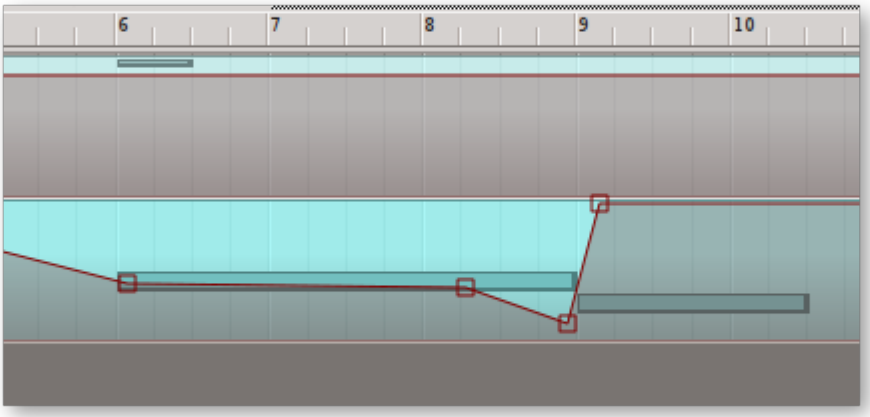
Adjusting automation nodes

The type of transition used between automation nodes can be customized via the track's **Automation** button, within the **Mode** category. **Hold** places the automation nodes into absolute values until otherwise changed.



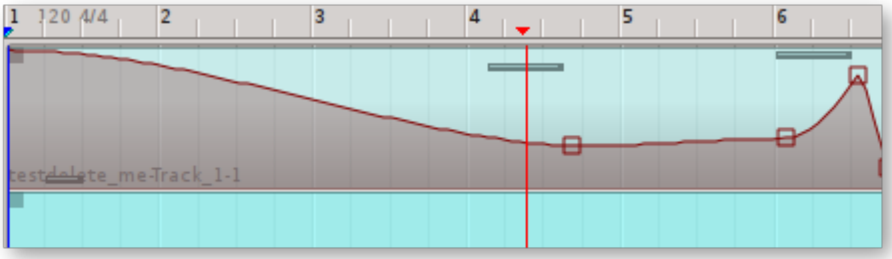
"Hold" automation mode

- **Linear** - effects an even, linear progression from one node to the next. This is the default automation mode.



"Linear" automation mode

- **Spline** - same as Linear, except with B-splines instead of a completely linear progression between nodes.
- **Logarithmic** - same as Spline, except that the degree of the Bézier curves changes proportionately to the difference between nodes.



"Spline" automation mode

- **Color** - when you automate volume, panning and effects all on one track it can get confusing. Use **Color** to change the shade of the automation overlay so that you can distinguish one set of controls from another. **Lock** prevents accidental changes to automation; this is especially useful when working on a different automation function on the same track.

If you are a skilled live mixer and want to record your performance in real-time, you can override existing automation decisions, or record new ones, in two different ways, each available in the Automation menu:

- **Play** - with this enabled, you may make real-time changes to existing automation, and upon release of the slider or knob that you are automating, the value returns to the pre-existing value. This may be combined with the Record option.

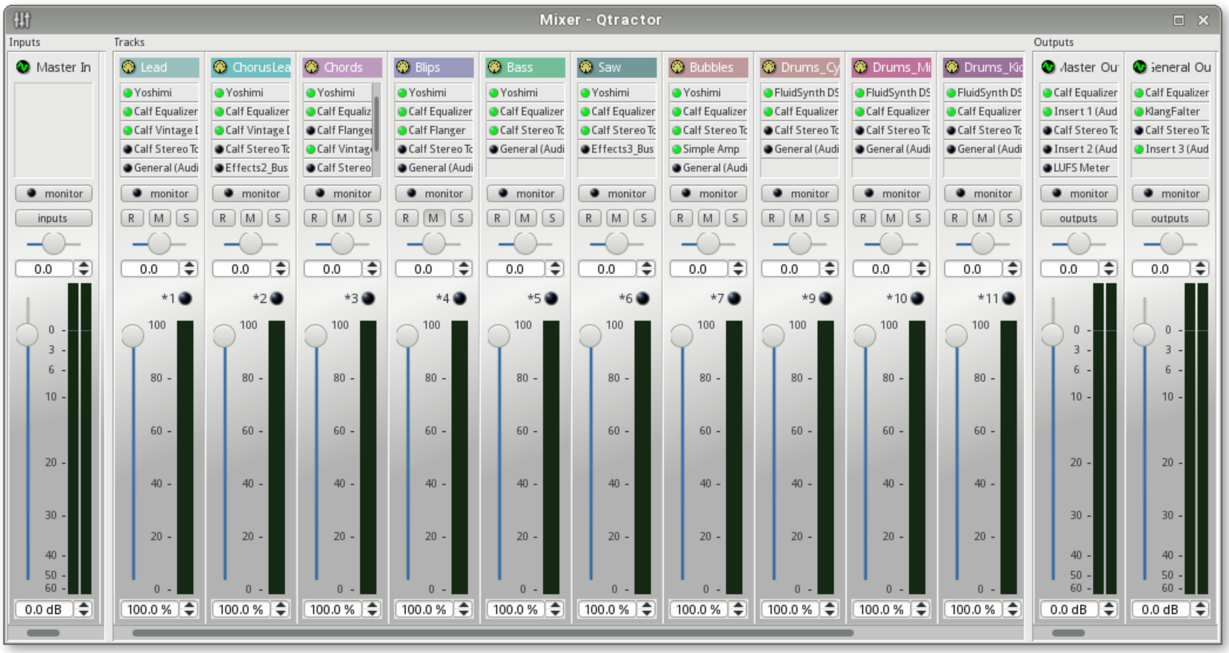
- **Record** - with this enabled, you may make real-time decisions for automation and every move will be recorded, in real-time, as a node in your track. They can be manually changed later, if needed.

4.7. Mixer

4.7.1. Mixer Summary

The Mixer window serves for session control, monitoring, recording and assistance in mix-down operations. It is divided into three panes: the left accommodates all input buses; the centre, individual track strips (either MIDI, with a yellow “plug” icon, or audio, with a green “waveform” icon); and the right, output buses. These are marked as Inputs, Tracks and Outputs respectively. Right-clicking in this area allows you to toggle the Inputs/Outputs’ displays on and off; you can also left-click and drag & drop them both within and outside of the Mixer.

The Mixer window can be accessed via the **View** / **Windows** / **Mixer** menu. It opens as a separate window, making it a prime candidate for occupying a second screen if you have a multiple-head set-up.



The Mixer window

Each vertical mixer strip offers a volume slider (MIDI tracks) or gain control (audio tracks and buses), panning slider, Mute/Solo/Record state buttons (tracks only), monitor button, connections button (buses only) and output event activity LED (MIDI tracks only).

Right-clicking in the panel directly underneath each strip's name allows you to perform a variety of operations. For instance, you can add plugins, or set up dedicated outputs via **Audio / Dedicated**. A dedicated output is an audio output specifically for that track, which can be connected via JACK (in Qtractor's **Connections** window, or QJackCtl, it will appear in the **Audio** tab, under *Qtractor*). You can also choose whether to **Auto-connect** the dedicated output via the same right-click menu - doing so means that it will automatically be connected to *system:playback_1* and *2*.

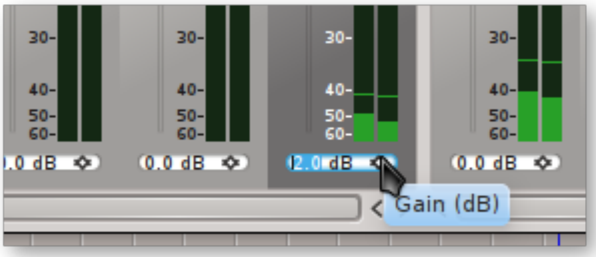
At the extremities of the Mixer window are the *Master In* and *Master Out* buses. These are independent of any one track and control everything coming into or going out of the Mixer. Here, you can adjust the master input/output levels and place any effects processors you want to apply globally (such as a compressor to prevent peaks on outgoing audio).

Monitoring is presented in the form of peak level meters for audio and note event velocity for MIDI, both with fall-off eye-candy. Audio volume is presented on a dBFS scale (IEC 268-10) and pan is applied in approximated equal-power effect (trigonometric weighting). For MIDI tracks, volume control is implemented through the respective channel's Controller 7 and system-exclusive master volume for output buses. MIDI pan control is only available for track strips and is implemented through the channel's Controller 10. MIDI input buses have volume and pan controls disabled.

4.7.2. Setting Levels

When monitoring the levels of your tracks, make sure that the sound does not enter the red zone on the volumeter. In audio production, any sound that goes above 0dB hits its peak and can cause distortion. It's not uncommon to flirt with 0dB to truly maximize the impact of important sounds, but leaving sounds above 0dB in your final mix can ruin your project.

Using the volume level slider, adjust your track volumes to safe settings while listening to your project. Then, listen to your project again to refine the levels in relation to one another. If the sound source is too low (or too loud), you can add (or subtract) gain at the input point with the Gain adjustment at the very bottom of the Mixer window. This adds or subtracts gain level at the point of the sound's entry, meaning that you are changing the amount of sound reaching any effects processors. Generally, gain adjustment is avoided and other tools, like a Compressor or Equalizer, are favoured.



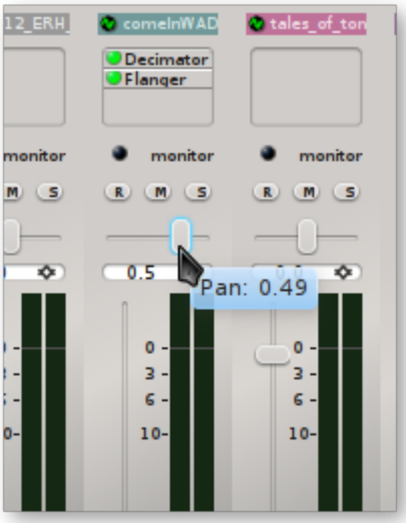
The Mixer's gain adjuster

Remember that Track levels don't exist in a vacuum; the levels are cumulative. So, if you have two tracks with near-maximum levels that produce very loud sounds at the same time, the *Master Out* level will **peak** even if each track was "safe" independently.

There is no authoritative way of achieving a balanced mix. Some professional mix engineers favour an additive mix, in which they start all levels at a reasonable and safe level and then accentuate important parts by adding volume to tracks. Others favour a subtractive method, starting everything at the highest safe (or unsafe) level and then bringing volumes down so that everything falls into place. Predictably, still others prefer a combination of these methods.

4.7.3. Panning

By default, tracks are set to play from the center point of the stereo space, meaning that the sound will feel as if it's situated directly in front of the listener, or spaced evenly between his/her ears. To diversify the sound, perhaps in order to emulate a live set-up, to suggest that an instrument is just next to the listener, or simply to broaden the scope and impact of the piece, you can *pan* tracks between the left and right speakers using the panning slider.



The Mixer's panning slider

When panning a track, "less" is often "more" and an even spread is usually best. In other words, instead of arbitrarily assigning tracks in stereo space just to give a spread to your sound, try to consider their overall balance. Avoid moving sounds to the extreme stereo positions (also known as *hard left/right*), since, in real life, sounds are rarely only heard in one ear and not the other. The sound of extreme stereo separation was common in early stereo mixes from the 1960s, in which the drums might only be heard in the right ear and the guitar only in the left; the effect was revolutionary at the time but is largely considered utterly unnatural now.

Generally, important tracks (vocals, lead etc.) and low frequency/groove tracks (bass, kick drum etc.) are positioned in the middle, while other parts are panned to the sides in various degrees. The extreme edges are often used for atmosphere, emphasis, or effect.

4.8. Signal Flow

4.8.1. Signal Flow Summary

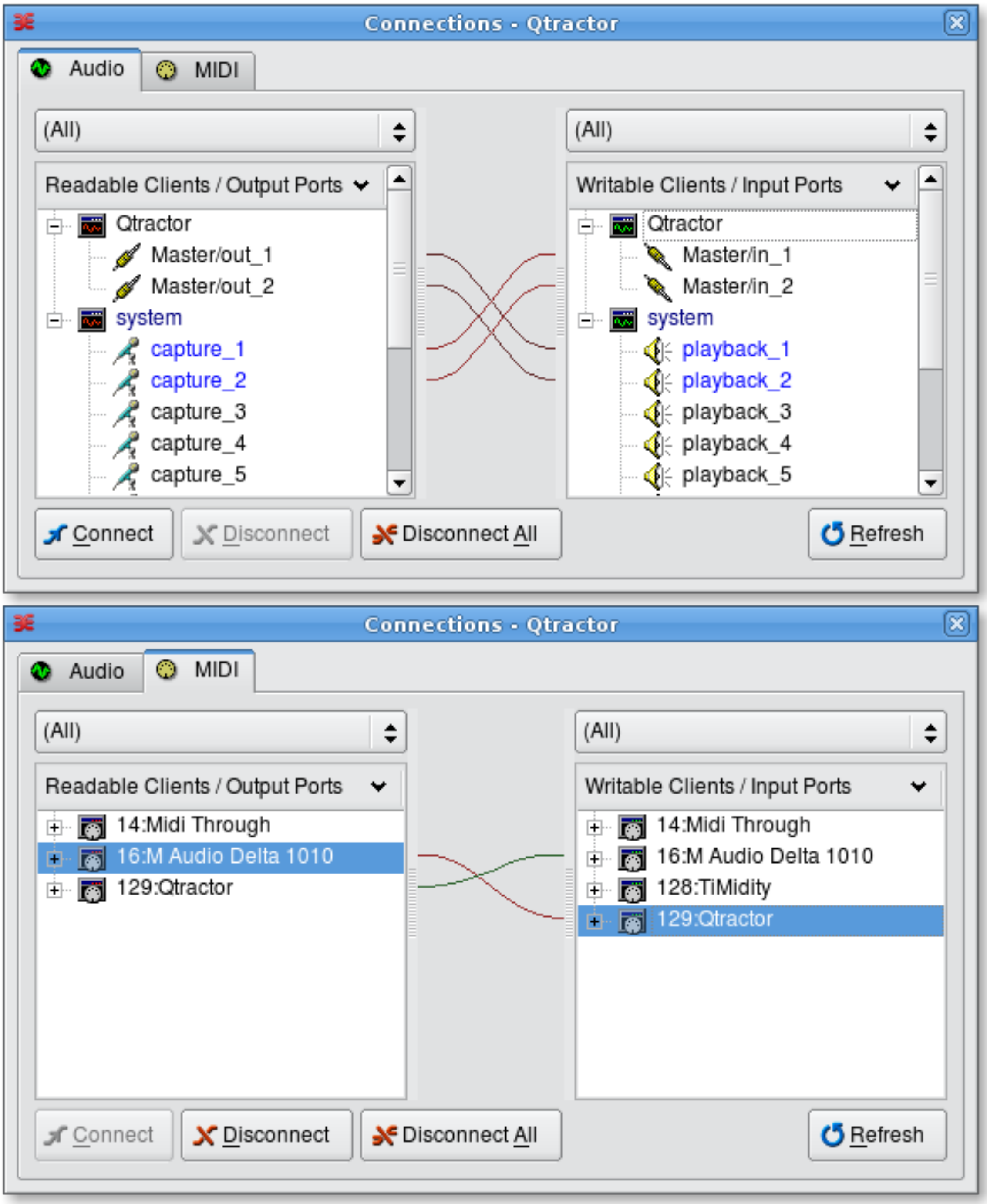
For anything other than the simplest of projects, you will likely want to manipulate the flow of MIDI and audio data, routing it into, out of and through Qtractor in a variety of ways. You may want to drive an external sound module, accept input from a MIDI controller to record performance data, group together similar tracks to share the same plugin, or perform some other kind of processing before final export. In Qtractor, **Connections**, **Buses**, **Aux Sends** and **Inserts** are all used to control “what goes where”. There are many different ways to approach the process of routing and exactly what you’ll need will depend on your set-up. With this in mind, this section will give a general overview of the options available so that you can decide what best fits your requirements.

4.8.2. Connections

The Connections window serves to establish the audio and MIDI port connections between the internal core layer input and output buses (ports), and the external devices or client applications. It can also be used to connect external client application ports - either JACK clients for audio or ALSA sequencer clients for MIDI. All active connections on the existing input and output buses are saved and restored upon session recall. To open the Connections window, select **View** / **Windows** / **Connections**, or click on its icon.

Connections icon

A window similar to the below will then open. Its features and usage are almost identical to those of QjackCtl.



Qtractor's Connections window, showing both the Audio and MIDI tabs. "Readable" ports are sources of data (where audio or MIDI data can come from) while "writable" ports are places that data can be routed to (sent to)

Once you've decided how to approach the routing for a particular project, you can then make the necessary connections between Qtractor and whatever other programs or external devices you're using. You can connect a Readable/Output port (left pane) to a Writable/Input port (right pane) in several ways. One method is to click a port in the Output pane and, with the mouse button held down, drag the cursor to a port in the Input pane until the port is highlighted, then release the mouse button. Another method is to left-click a port in the Output pane, left-click a port in the Input pane, then either left-click the **Connect** button or right-click elsewhere in the window and select *Connect* from the pop-up menu. With this method you can also `Ctrl`-click or `Shift`-

click to select multiple ports. Regardless of the method you use, a line representing a "virtual cable" will appear between the connected ports in the middle section of the window.

The *System* input ports *playback_1/2* represent your default left/right stereo output (e.g. your first-configured sound card). If you wish to use alternative destinations for playback, monitoring etc., you can connect them up here.

4.8.3. Buses

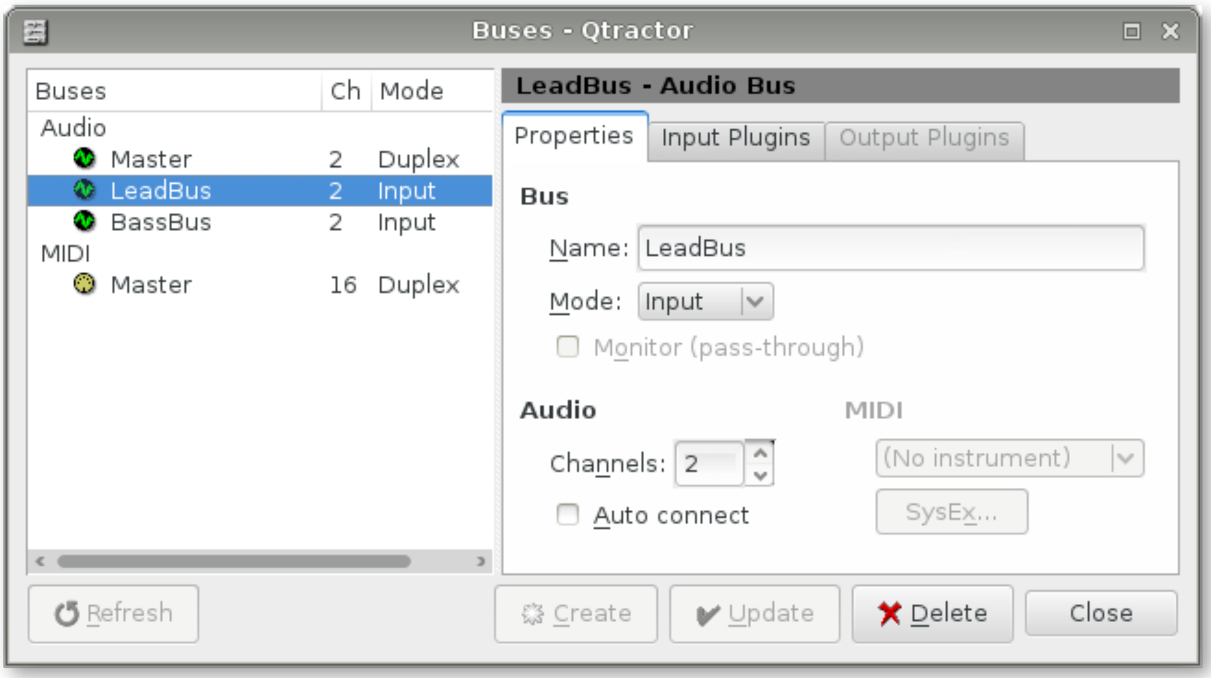
As explained at the start of this chapter, a bus is like a pipeline through which streams of data can travel. Qtractor has **Input**, **Output** and **Duplex** buses for both audio and MIDI. Input buses receive data streams from particular sources (inside Qtractor, or from other software or hardware), Output buses send streams to other destinations and Duplex buses both receive and send streams.

How many and what kind of buses you need will depend on what you want to achieve, but the basic steps to follow are:

1. Create a bus

- Assign it as an Input or Output for a track
- Connect it via the Connections window

To open the Buses window, click on the **View** / **Buses** menu.



Buses	Ch	Mode
Audio		
Master	2	Duplex
LeadBus	2	Input
BassBus	2	Input
MIDI		
Master	16	Duplex

LeadBus - Audio Bus

Bus

Name: LeadBus

Mode: Input

□ Monitor (pass-through)

Audio

Channels: 2

□ Auto connect

MIDI

(No instrument)

The Buses window, showing one MIDI and three audio buses

The left pane contains a list of the session's buses; clicking on one of these will display its information in the right pane. The right pane has three tabs:

- **Properties** - Name, Mode (Input, Output or Duplex), Monitor (Duplex buses only), Channels and Auto-connect (Audio buses only) and Instrument and SysEx (MIDI buses only)

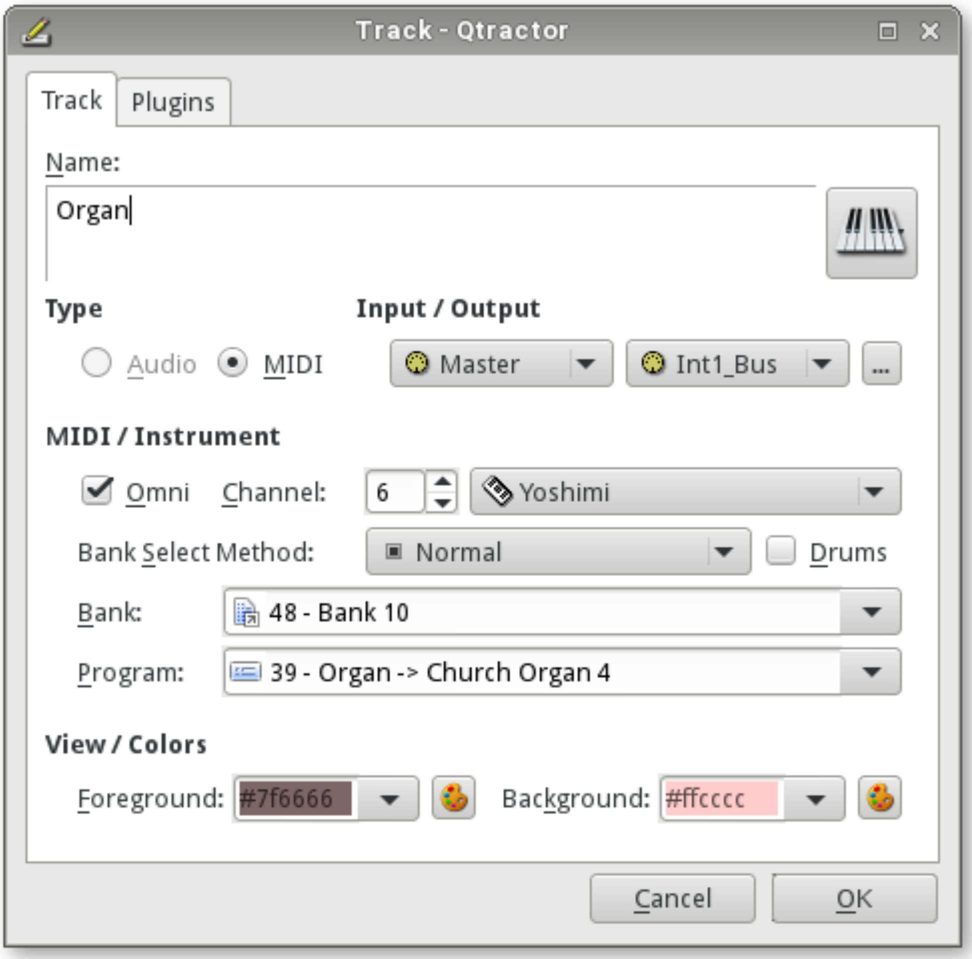
- **Input Plugins** - plugins processed on input (Input and Duplex buses only)
- **Output Plugins** - plugins processed on output (Output and Duplex buses only)

The method of creating a bus may not be immediately apparent. To create a new MIDI or audio bus, select any bus of the desired type in the left pane (if you're working on a clean session, the only choices will be MIDI and Audio Master) in order to display its properties in the right pane. Next, change the text in the **Name** field; this will enable the **Create** and **Update** buttons. To create a new bus, click **Create**.

Before clicking **Close**, you may wish to change other options in the right pane. For audio buses, a commonly used option is the **Channels** setting. Its default is **2**, which, for an Input bus, will give you a stereo recording. However, you may want to set this to **1** to create a mono input bus in order to record, for example, an external source via a single microphone. With a bus' **Channels** set to **1** and the bus assigned as the Input to a track, any recording on that track will then be in mono.

If you've made any changes, click **Update**, then **Close**. (If you don't click **Update** a reminder dialogue box will appear in any case.)

Once you've created your bus, you'll need to assign it to the relevant track(s). To do so, select a track and choose the **Track** / **Track Properties…** menu, or double-click the track in the left pane of the main view. This will open the **Track** window. In the **Input / Output** area, assign your bus via the drop-down box (whether it appears in Input, Output or both will depend on what type of bus it is).



Name: Organ

Type: ○ Audio ● MIDI

Input / Output: Master | Int1_Bus

MIDI / Instrument

☑ Omni Channel: 6 Yoshimi

Bank Select Method: Normal □ Drums

Bank: 48 - Bank 10

Program: 39 - Organ -> Church Organ 4

View / Colors

Foreground: #7f6666 Background: #ffcccc

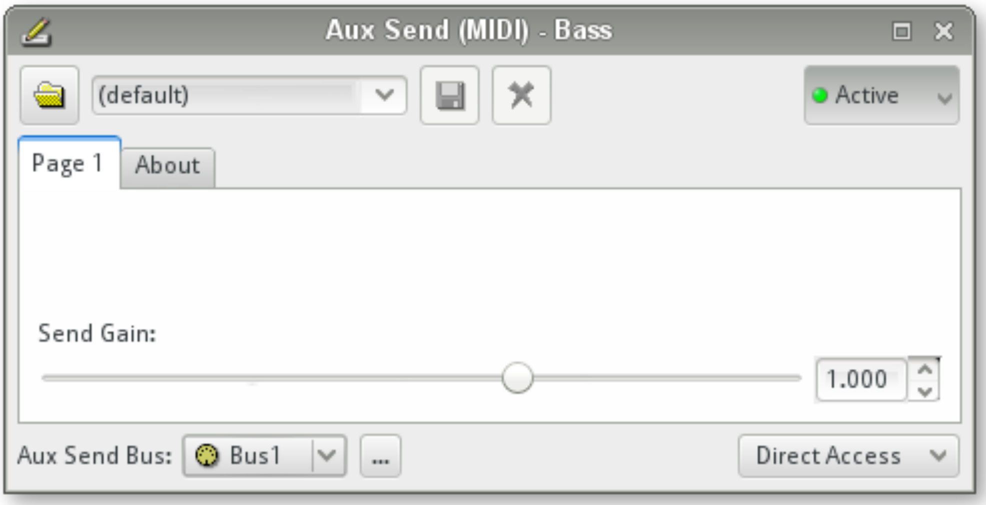
Track window for a MIDI track

The final step is to connect the bus up to its destination - a task which is achieved via the Connections window. Again, exactly what you need to do will depend on your set-up, but if you wanted, for example, to send a particular MIDI track to an external software or hardware synth, you would connect its output bus' Output connection in the left pane to your synth's Input in the right pane. One thing you should *not* do is route the output of one Qtractor bus to the input of another, as this will give unpredictable results.

Buses are an extremely valuable tool and their use is almost essential for any reasonably sized project and when using Qtractor with external inputs and outputs. They enable you to use the full 16 MIDI channels through each bus, giving you much more flexibility than if you were using the Master MIDI bus only. You can, for example, have a dedicated MIDI bus routed to a multi-instrument synthesizer, sound module or soft synth, giving you 16 separate channels to control different instruments within the synth. Buses can also be used to route the audio output of external synths into Qtractor audio tracks for recording. For more information on this topic, see this How To.

4.8.4. Aux Sends

Aux Sends are used to route audio or MIDI from a track or input bus to another destination *within* Qtractor - specifically, to a bus. To create one, open the Mixer, right-click under the name of your track or bus, then choose **Inserts** / **Audio** or **MIDI** / **Add Aux Send**. In the window which appears, set the Aux Send to "Active" (top right button), choose your destination bus in the **Aux Send Bus** combo box (bottom left) and choose the amount you want to send via the **Send Gain** slider.


Aux Send (MIDI) - Bass

(default)

Active

Send Gain: 1.000

Aux Send Bus: Bus1

Direct Access

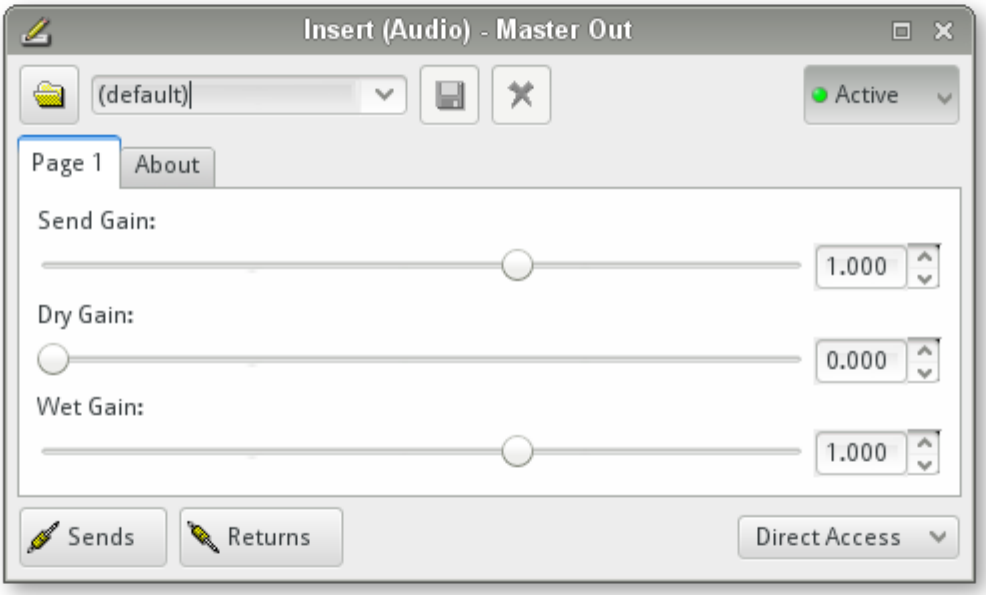
The Aux Send window for a MIDI track

What value to set **Send Gain** to will depend on the purpose of your destination bus. For example, you may want to send several tracks to the same bus in order to share that bus' plugin(s). This is a reasonably common requirement, particularly for reverb, where you would have an "overall" setting in the destination bus' plugin, then adjust the amount you send to this via each track's Send Gain slider. In this way, you can control the amount of reverb applied, in order, for example, to mimic an instrument's placement on the stage.

4.8.5. Inserts

Inserts are similar to Aux Sends, but are used to route audio or MIDI *externally* of Qtractor. They can be added to tracks, or to input or output buses. You can even route your entire mix to an external destination, such as JAMin, before routing it back into Qtractor for export.

To create an Insert, first, launch your external application. Then, open Qtractor's Mixer, right-click under the name of your track or bus and choose **Inserts** / **Audio** or **MIDI** / **Add Insert**. In the window which appears, set the Insert to "Active" (top right button) and adjust **Send Gain**, **Dry Gain** and **Wet Gain** as necessary.



Send Gain: 1.000

Dry Gain: 0.000

Wet Gain: 1.000

The Insert window for an audio track

Next, use the **Sends** button to connect the Insert to your external application and the **Returns** button to connect the application back to the Insert (these buttons open up the same **Connections** window discussed above). The signal will now flow from your track/bus out to the external application, then return to the same track/bus and continue on from the point below the Insert.

4.8.6. Connection Persistence

When working with purely internal sources and destinations (plugin soft synths, Aux Sends etc.) Qtractor will automatically restore any connections you have made upon re-loading the session. With external applications, however, things are slightly different: Qtractor will only restore such connections *if they are present when the session is saved*. For example, if you save and close your session with everything correctly connected, upon restarting both Qtractor and your external application, the connections will be correctly restored. However, if you close the external application, then make a change in Qtractor and save (i.e. with Qtractor and the application in a "non-connected" state), upon restarting, the connection will be lost and you'll need to make it again.

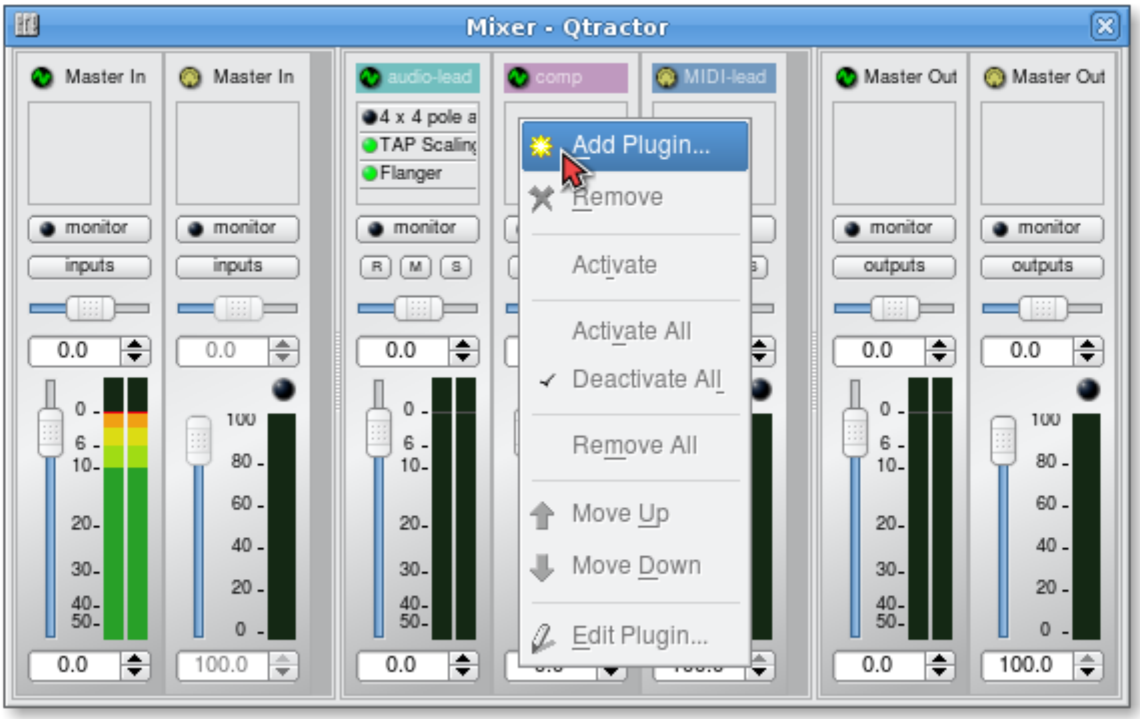
This behaviour can be avoided by always ensuring that you have your external connections in order before closing Qtractor (easier said than done when using many external applications). It can also be mitigated somewhat by using QjackCtl's *Patchbay* feature to make the connections persistent; however, this too is rather unwieldy as it means adding each item manually to the

Patchbay. A third approach would be to use a session manager, such as QJackCtl's *Session* feature (which employs "JACK session"), or NSM.

4.9. Plugins

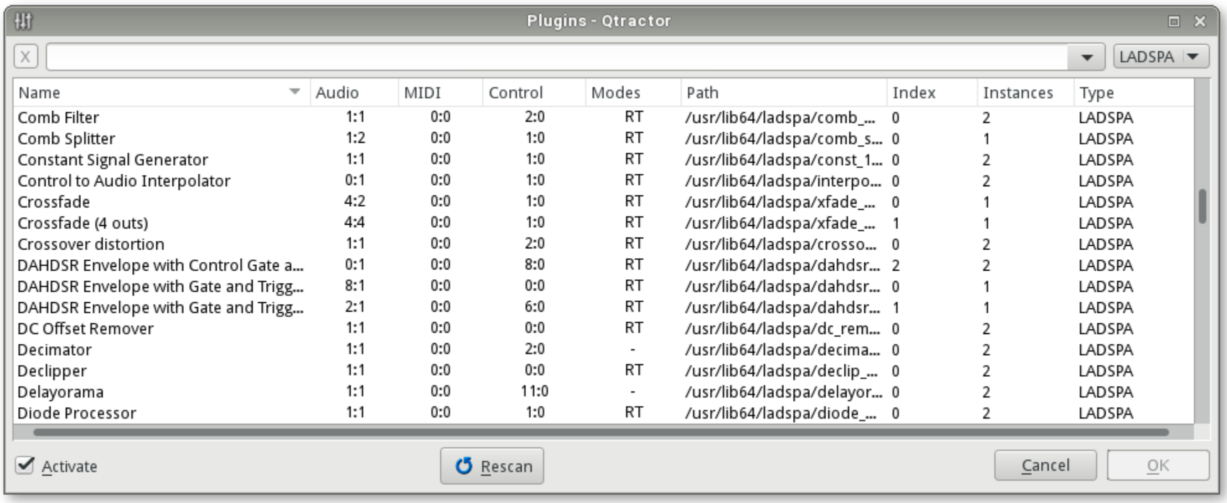
4.9.1. Plugins Summary

There are four types of plugins supported within Qtractor: LADSPA, DSSI, VST and LV2. They can be added to both buses and tracks and are aggregated seamlessly as one single instance on a multi-channel context. To add a plugin, open the Mixer, right-click in the panel below the name of the desired track and select **Add Plugin**.



Mixer window showing plugin area at top. Plugins are added, removed, activated/deactivated, moved in the processing chain and their settings edited using a right-click context menu

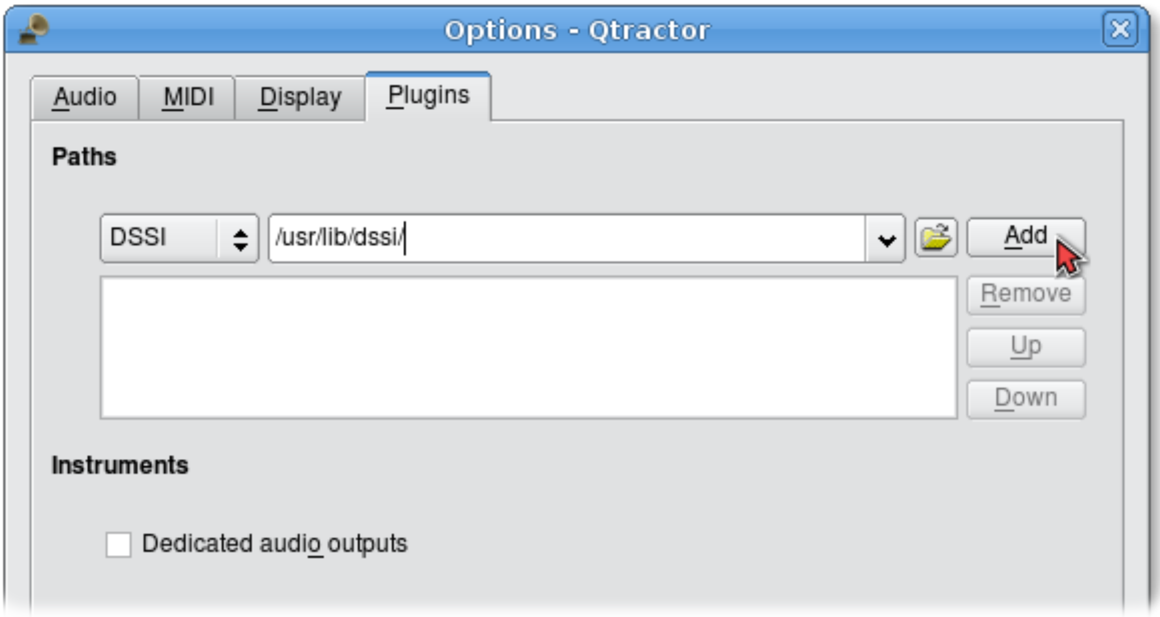
In the **Plugins** window which then appears, select the plugin type - **(Any)**, **LADSPA**, **DSSI**, **VST**, or **LV2** (see the following sections for information on the different types) - in the top-right box. Next, select your plugin from among those installed and, if desired, tick **Activate** in the bottom-left corner to enable the plugin as soon as it's added to the mixer strip. If you have a long list of plugins to choose from, you can use the top-middle box to search for what you want.



Name	Audio	MIDI	Control	Modes	Path	Index	Instances	Type
Comb Filter	1:1	0:0	2:0	RT	/usr/lib64/ladspa/comb_...	0	2	LADSPA
Comb Splitter	1:2	0:0	1:0	RT	/usr/lib64/ladspa/comb_s...	0	1	LADSPA
Constant Signal Generator	1:1	0:0	1:0	RT	/usr/lib64/ladspa/const_1...	0	2	LADSPA
Control to Audio Interpolator	0:1	0:0	1:0	RT	/usr/lib64/ladspa/interpo...	0	2	LADSPA
Crossfade	4:2	0:0	1:0	RT	/usr/lib64/ladspa/xfade_...	0	1	LADSPA
Crossfade (4 outs)	4:4	0:0	1:0	RT	/usr/lib64/ladspa/xfade_...	1	1	LADSPA
Crossover distortion	1:1	0:0	2:0	RT	/usr/lib64/ladspa/crosso...	0	2	LADSPA
DAHDSR Envelope with Control Gate a...	0:1	0:0	8:0	RT	/usr/lib64/ladspa/dahdsr...	2	2	LADSPA
DAHDSR Envelope with Gate and Trigg...	8:1	0:0	0:0	RT	/usr/lib64/ladspa/dahdsr...	0	1	LADSPA
DAHDSR Envelope with Gate and Trigg...	2:1	0:0	6:0	RT	/usr/lib64/ladspa/dahdsr...	1	1	LADSPA
DC Offset Remover	1:1	0:0	0:0	RT	/usr/lib64/ladspa/dc_rem...	0	2	LADSPA
Decimator	1:1	0:0	2:0	-	/usr/lib64/ladspa/decima...	0	2	LADSPA
Declipper	1:1	0:0	0:0	RT	/usr/lib64/ladspa/declip_...	0	2	LADSPA
Delayorama	1:1	0:0	11:0	-	/usr/lib64/ladspa/delayor...	0	2	LADSPA
Diode Processor	1:1	0:0	1:0	RT	/usr/lib64/ladspa/diode_...	0	2	LADSPA

Plugins window, showing LADSPA plugins currently installed

Qtractor will look in the most likely installation directories for plugins and display any found in the above window. If, however, you have plugins installed in non-standard directories, you should specify their paths in the **Plugins** tab of the **View** / **Options…** window. Once you've made your selection, click the **OK** button to add the plugin.



DSSI /usr/lib/dssi/

□ Dedicated audio outputs

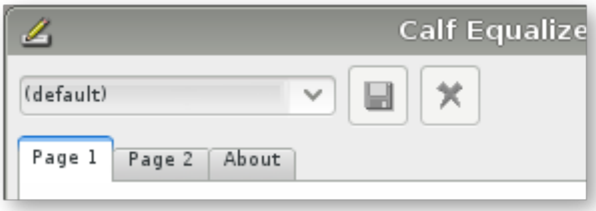
*Specifying the locations (paths) of plugins in the **View** / **Options…** window*

Plugins are processed in order from top to bottom and can be individually selected, activated and moved (by dragging/dropping) within the plugin chain order. They can also be copied or moved to other mixer strips by dragging/dropping and choosing your desired option from the pop-up menu which appears; a faster way to accomplish this is `Ctrl`+drag to copy and `Shift`+drag to move. There are a couple of things to note about plugins, which may affect your workflow.

- All plugins are *pre-fader* - that is, they are processed before the Mixer's Gain fader. If you want them to behave as though they were post-fader you can insert any plugin which controls gain at the start of the chain (i.e. before your pseudo post-fader plugins) and use that, rather than the Mixer fader, to control the track's gain

- There is no plugin delay compensation, so if a plugin adds latency to the signal chain this won't be corrected for. If you do need to correct for this, the only option is to do so manually by adjusting a clip's timing via the **Offset** value in its Properties window

Individual plugin control parameters can be modified in real-time through provided dialog windows and maintained as named presets for re-usability. To create a preset, once you have your desired settings ready, enter a name in the box at the top left of the generic plugin GUI (it will say "(default)" to begin with), then click the save icon next to it. You can select from among your saved presets by clicking the arrow next to the preset name or, when in the Mixer, by right-clicking the plugin and choosing **Preset**.



Generic plugin GUI, showing preset box and save and delete icons

Note that certain plugin-specific GUIs, such as Calf LV2, may not display the presets information even though the functionality itself is there. In such cases, you can access the generic GUI by right-clicking the plugin in the Mixer and choosing **Properties…**, then create presets from there. When more than one GUI exists for a plugin, you can also opt to choose which you want by enabling *Select plugin's editor (GUI) if more than one are available* in the **View** / **Options…** / **Plugins** menu.

4.9.2. LADSPA

LADSPA has been a Linux audio plugin standard for many years. There are literally hundreds of LADSPA plugins available for Linux, providing a huge variety of processing effects, such as equalization, filtering, reverb, chorus, amp and speaker simulation and so on. Well-known LADSPA plugins include the swh package by Steve Harris and the tap package by Tom Szilagyi. These may already be present in your distro's repositories.

4.9.3. DSSI

DSSI plugin support is available for instrument and effects plugins. You must have the core DSSI subsystem installed in order for this type of plugin to function. Well-known DSSI plugins include the fluidsynth soft synth for working with soundfonts (`.sf2`/`.sf3`) and Sean Bolton's WhySynth. Again, these may already be present in your distro's repositories.

4.9.4. VST (Linux Native)

Native Linux VST plugins are supported, but there are only a few native Linux VSTs available. Please see section 2.4.2 for complete information on building Qtractor with native VST support.

Note: Native Linux VST support does NOT include running of Windows® VST plugins. Please use the DSSI-VST wrapper when attempting to use this type of plugin, and make sure your Windows® VST plugins are located within your DSSI path environment variable (the variable name is **VST_PATH**).

4.9.5. LV2

LV2 is the extensible successor to LADSPA and offers many improvements over it. A wide variety of plugins are available, such as the synthv1 synth, by Rui Nuno Capela (author of Qtractor), and the Calf set of effects. If you have the choice of using an LV2 plugin over a LADSPA or DSSI one, it's generally better to use the LV2 one. Doing so will ensure that when saving a Qtractor archive file (**.qtz**), all the relevant information is correctly included in the archive.

4.10. Working with MIDI

4.10.1. MIDI Summary

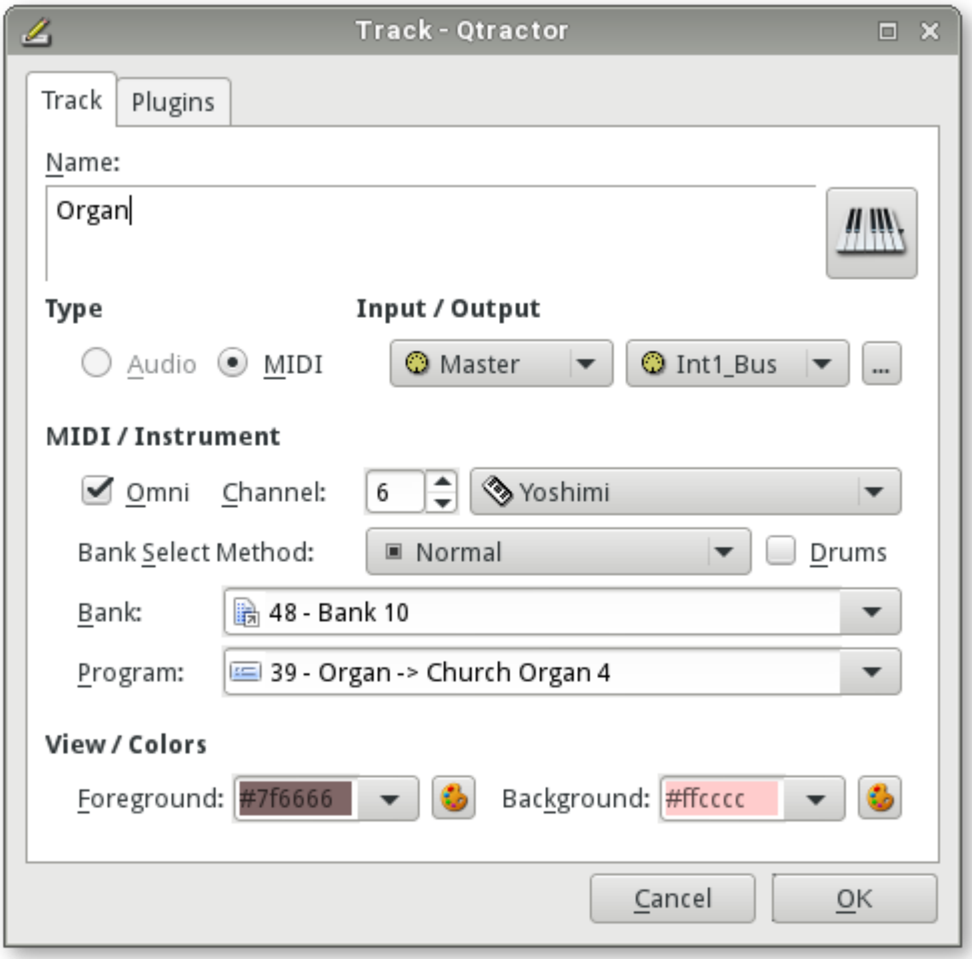
MIDI is a versatile format for triggering sounds. It can be used to trigger hardware synths external to your computer, software synth applications on your computer, samples, loops, automation functions and more. Some music programs bundle your MIDI data into your project file, which can make it difficult to share one MIDI loop between projects. However, Qtractor saves all MIDI tracks as independent files, meaning that you can easily re-use them.

There are basically three ways to get MIDI data into your Qtractor session and they are each covered in the sections below:

- Import MIDI data from an existing file
- Play in MIDI data via a USB MIDI keyboard controller
- Enter MIDI data into the MIDI Editor (sometimes called a “piano roll” or “matrix editor”)

Before beginning with MIDI, unless you're using Qtractor to drive an external hardware synth you should make sure that you have some software synthesizers installed on your system. If you've just installed soft synths now you should relaunch Qtractor so that it will detect the newly available plugins.

Note that with each successive MIDI track you add, you must increment the MIDI Channel used (Qtractor will do this by default), otherwise key presses on your MIDI controller, or playback from Qtractor, will use the same MIDI channel to trigger different sounds. In other words, if Track 1 and Track 2 are both using MIDI Channel 1, then a note played on Track 2 will also trigger the instrument assigned to Track 1. To change a track's Channel, double-click on the track in the left hand pane of the main view then, in the **Track** window which appears, select the **Channel** number underneath **MIDI / Instrument**. If you run out of channels, you can use multiple buses.



Name: Organ

Type: ○ Audio ● MIDI

Input / Output: Master, Int1_Bus

MIDI / Instrument

☑ Omni Channel: 6 Yoshimi

Bank Select Method: Normal □ Drums

Bank: 48 - Bank 10

Program: 39 - Organ -> Church Organ 4

View / Colors

Foreground: #7f6666 Background: #ffcccc

Track window for a MIDI track, where the MIDI Channel can be selected

4.10.2. Importing MIDI Data

Since MIDI data consists of nothing more than digital signals to activate and deactivate sounds at certain times, MIDI files are small and easily distributed online. A simple internet search for MIDI files will result in thousands of files for nearly any song you can name. Taken alone, MIDI files are useless; imported into Qtractor and assigned instruments, however, the files come to life like a player piano. The MIDI file you import into Qtractor could be from the internet, one you created yourself in another application, or even a data dump from a MIDI-capable hardware synth. In this example, we will use one from the internet:

1. Download a MIDI file, such as JS Bach “Invention in C minor, BWV 773 (Canon)” and save it to your hard drive - preferably in your Qtractor’s session directory, to keep your project self-contained

2. In Qtractor, go to the **Track** menu and select **Import Tracks** / **MIDI**. Alternatively, you may open the **Files** pane, click on the **MIDI** tab and use the right-click menu to import the MIDI file without automatically adding it to a track (if you do this, you’ll need to drag the file into the workspace manually). *Please also see the note in the Files section regarding importing MIDI files*

3. The MIDI tracks will appear in your Qtractor session. Imported tracks will often have embedded General MIDI instrument assignments, as is the case with this example. General

MIDI was an effort to standardize a set of 128 instruments to provide similar playback results across all systems

If you have a soft synth installed and assigned to the relevant track(s) you can press the *Play* button in the top toolbar of Qtractor to hear the MIDI file. If you're using a General MIDI-compliant soft synth bank you should get a good approximation of how the original author of the file wanted the tracks heard.

4.10.3. Creating MIDI Data with a MIDI Controller

A MIDI controller is a piece of hardware to help you play the software-based synthesizers in your computer. Typically, it is a piano keyboard with a few octaves and no built-in sounds of its own. It usually plugs into the USB port of your computer and, once configured, will pass on any key press to Qtractor, allowing you to play a software synth in real-time as well as record the key presses themselves.

MIDI and USB are two technologies that, fortunately, have been kept universal enough that you almost don't need to question whether any given USB MIDI controller will work with your system. There are perfectly capable USB MIDI controllers being offered from vendors like Akai, Roland/Edirol, Digidesign/M-Audio and many others. If you require realistic, weighted keys with high resolution touch-sensitivity then look at the upper price-range of controllers and go to a music store to audition them. However, if all you require are a few octaves of piano keys so that you can get a tune into the computer quickly and easily, the basic controllers will be enough.

To use your USB controller to input MIDI data:

1. Plug the USB MIDI controller into your computer

2. Launch QJackCtl and start it

3. Launch Qtractor and click on the **View** menu, then select **Connections** from the **Windows** category

4. In the **Connections** window, click on the **MIDI** tab. Find your USB controller device from those listed on the *Readable Clients/Output Ports* column on the left hand side. Open its category to see the available outputs

5. Connect the output of your MIDI device to Qtractor in the *Writable Clients/Input Ports* column

6. Create a new track, then double-click the track in the left pane of the main view. In the **Track** window which appears, make sure that the **Channel** value (underneath **MIDI / Instrument**) corresponds to the channel your MIDI device is transmitting on. Alternatively enable **Omni** alongside this to tell Qtractor to receive input on all channels

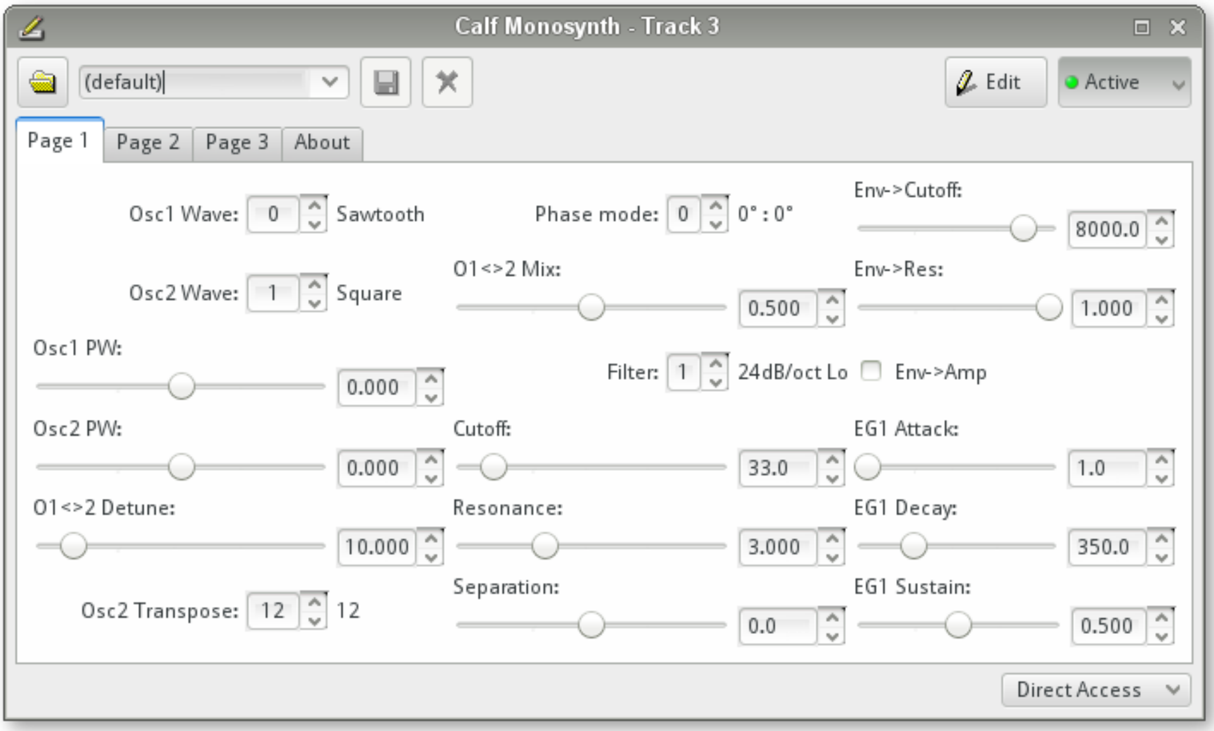
7. In the Mixer, enable the track's **Monitor** button (so that it turns green)

In Step 4, if you do not see your USB controller listed in the left column, check to ensure that your controller is powered on (some require external power; others are powered by the USB port) and check the cable connection. If you still cannot see the device in the left column make sure your computer sees the controller by checking `dmesg | tail`*. If your computer registers the device but you are not seeing it in* ***Connections*** *click* Refresh* in the bottom right of the window. If it still doesn't appear try restarting both QJackCtl and Qtractor.*

Presuming you have a soft synth installed and assigned to the track, pressing the keys on your USB controller should now trigger the soft synth in Qtractor, generating sound. To record MIDI data, arm your track by clicking its “R” button in the left pane of the main view, click the *Record* button in the top menu, then click the *Play* button to start the transport recording - every key press you make on your USB controller should now be recorded in real-time. Depending on the feature set of your controller, velocity, pitch and other data may also be recorded

If you re-record over a clip, Qtractor will create a new, overlapping clip (which can, if desired, be merged with the original clip using **Clip** / **Merge**). If you’d rather *overdub* the original clip, adding extra notes directly to it instead of creating a separate clip, select the desired clip and choose **Clip** / **Record** from the main menu, or **File** / **Record** from the MIDI Editor. Once selected, the clip will change colour and any data recorded in its region of the timeline will be added to it.

You may wish to control particular settings of a plugin by sending MIDI controller messages to it from your USB controller. Although assigning MIDI controllers to LV2 plugins is not permitted by the LV2 specification, Qtractor offers another way of handling this which works for all plugin types. To assign a controller, first open the plugin’s generic GUI, by right-clicking on its name in the Mixer and selecting **Properties…**. A window such as the below should appear, the content of which will be defined by the plugin’s exposed settings.

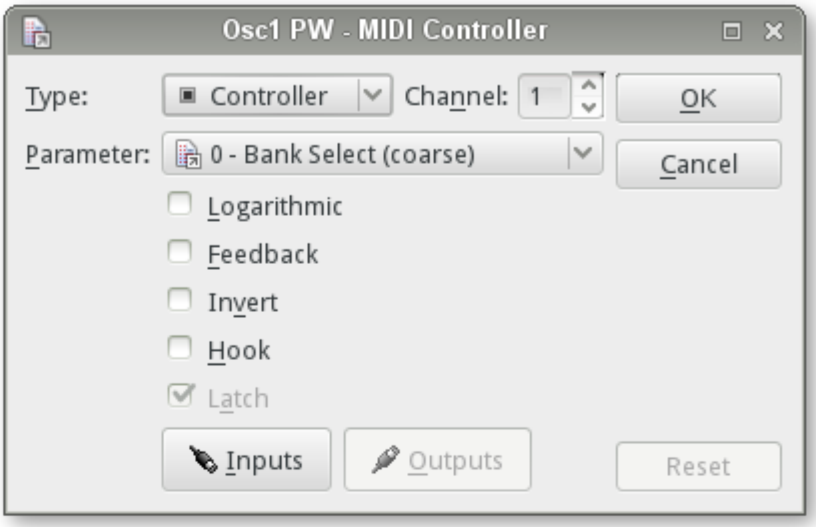


(default)

Osc1 Wave: 0 Sawtooth
Osc2 Wave: 1 Square
Osc1 PW: 0.000
Osc2 PW: 0.000
O1<>2 Detune: 10.000
Osc2 Transpose: 12 12
Phase mode: 0 0° : 0°
O1<>2 Mix: 0.500
Filter: 1 24dB/oct Lo
Cutoff: 33.0
Resonance: 3.000
Separation: 0.0
Env->Cutoff: 8000.0
Env->Res: 1.000
☐ Env->Amp
EG1 Attack: 1.0
EG1 Decay: 350.0
EG1 Sustain: 0.500

A plugin’s generic GUI

Next, right-click on the name of the setting you want to control and, in the dialogue box which appears, choose **MIDI Controller**. The below window should then appear.



Osc1 PW - MIDI Controller

Type: Controller

Channel: 1

Parameter: 0 - Bank Select (coarse)

□ Logarithmic

□ Feedback

□ Invert

□ Hook

☑ Latch

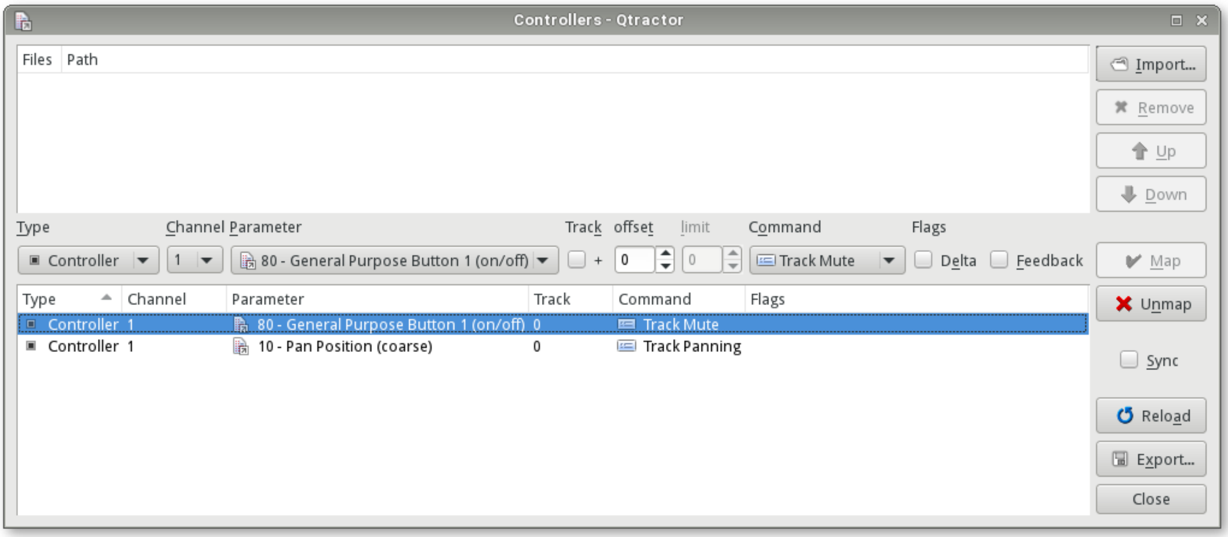
MIDI Controller window, where controller messages can be assigned to plugin settings

If you have not already done so, connect your hardware MIDI controller up, using the **Inputs** button. Then, choose which controller message to use via the **Type** and **Parameter** options, together with your desired MIDI channel in the **Channel** box. There are also several other options below these:

- **Logarithmic** - the linear scale of the MIDI controller (rotary knobs, sliders, etc.) is mapped to a logarithmic scale value on the target controlled subject. This is usually applicable to audio volume, dB scales or even frequency spectrum ranges
- **Feedback** - causes any change in the state value of the controlled subject to be sent/fed-back to the MIDI controller, to allow the latter to reflect and sync its state accordingly. This is mostly applicable to MIDI controllers with visual (e.g. LEDs) and/or mechanical (e.g. motorized faders) feedback
- **Invert** - the minimum-maximum range of the MIDI controller is inversely mapped to the maximum-minimum scale values on the target controlled subject (and vice-versa)
- **Hook** - the MIDI controller and its controlled subject state/value are always in sync (hooked to each other); otherwise a deferred sync mode is in effect, with both ends playing catch-up to each other's from either direction until they match their states/values
- **Latch** - this is used only for on/off toggle parameters and relates to *Latch* vs *Momentary* modes of switches. These are the opposite of each other, so latch=OFF is the same as momentary=ON and vice-versa. This option is mostly applicable to foot-switches or pedal controllers. *Latch* (=ON) mode is the default setting for most such controllers, but a controller that operates in *Momentary* mode will usually send an event when you depress the switch (0 -> 1) and another when you release it (1 -> 0). (This is the meaning of "momentary": the switch state is only signalled to be momentarily on while the switch is depressed, returning to its former state upon release.) Setting the **Latch** option to OFF for this latter case will simulate a toggle switch, meaning that the target subject state will only change to ON *or* OFF when the switch is *either* being depressed *or* released, respectively

Note that MIDI controller assignments are applicable only to the session you create them in. If you want to use the same assignment in a different session you'll need to create it again.

You can also create and manage generic MIDI controller mappings, plus import/export these mappings as a **.qtc** file, via the **View** / **Controllers** menu.



Type: Controller
Channel: 1
Parameter: 80 - General Purpose Button 1 (on/off)
Track: □ + offset 0 limit 0
Command: Track Mute
Flags: □ Delta □ Feedback

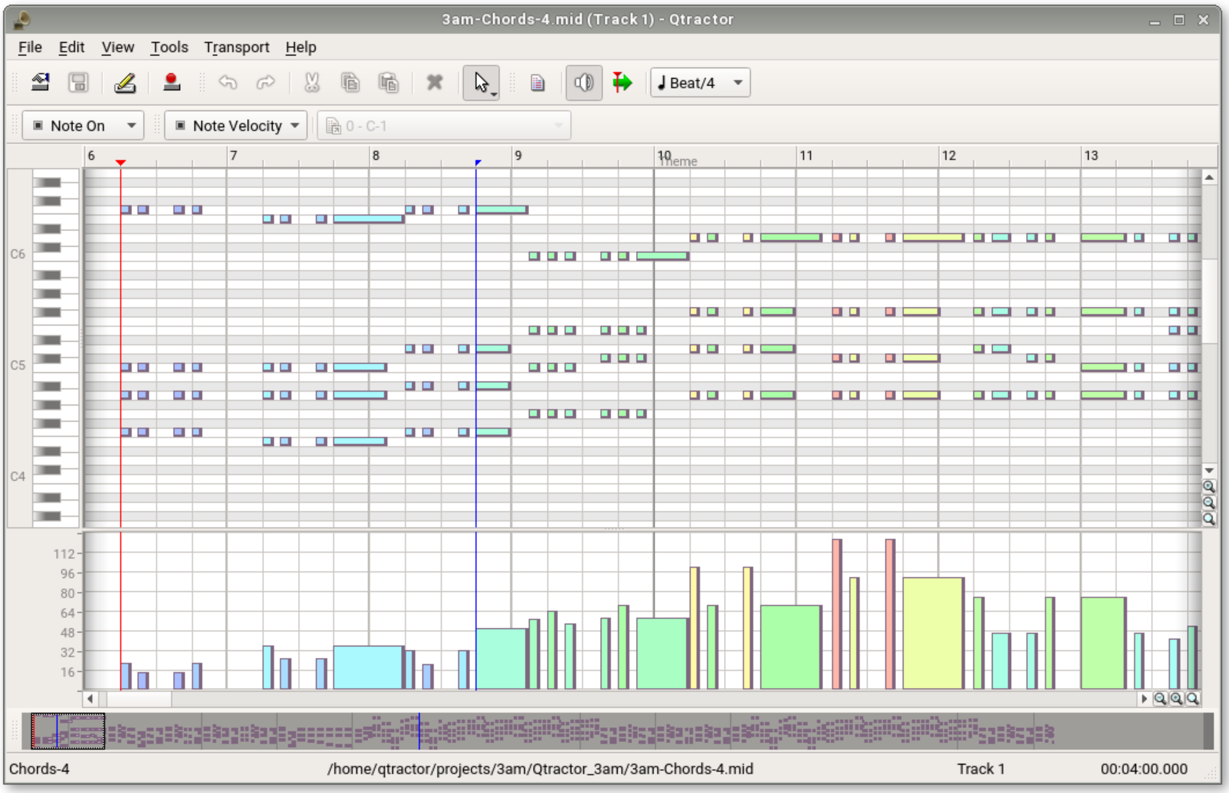
Type	Channel	Parameter	Track	Command	Flags
Controller	1	80 - General Purpose Button 1 (on/off)	0	Track Mute	
Controller	1	10 - Pan Position (coarse)	0	Track Panning	

The Controllers window, where generic MIDI controller mappings can be managed

4.10.4. Creating MIDI Data with the MIDI Editor

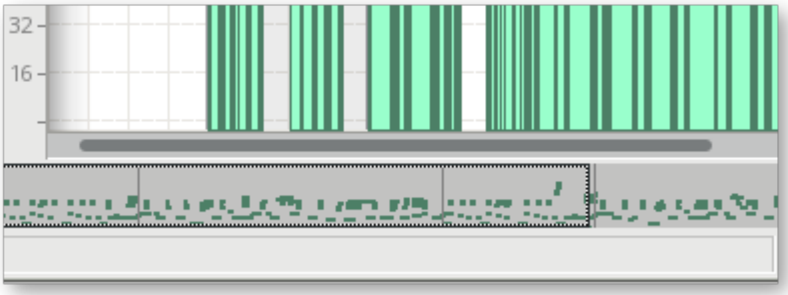
If you have no USB MIDI controller, you can still create MIDI data using Qtractor's *MIDI Editor* (sometimes called a "matrix editor" or "piano roll" in other music programs). Data is edited on a grid through the usual GUI operations, such as selection, drag-and-drop, move, cut, copy, paste and deletion. The MIDI Editor can be invoked either for raw entry or to modify existing MIDI data that you've imported or played into Qtractor.

The MIDI Editor is very intuitive. The main, upper pane is the *Editor Grid*, which represents the notes being played. Vertically, the grid corresponds with **notes** on the chromatic keyboard (pitched as per the keyboard on the left of the window); horizontally, it corresponds with **beats** per bar. The lower pane is the *Value Grid*, which controls extra MIDI data, such as velocity, program changes, pitch bends and so on. The data represented here changes according to the *Value* menu setting in the upper menu bar.



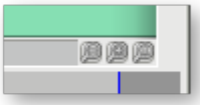
MIDI Editor window, showing notes in a familiar "piano roll" display (above) with their velocities (below)

You move around the MIDI Editor in the same way as the main work area - using the horizontal/ vertical scroll bars at the bottom/right of the window, the mouse wheel/`Shift`+mouse wheel, or the navigation window (at the bottom of the MIDI Editor window itself).



The MIDI Editor's navigation window, situated at the bottom of the MIDI Editor window itself

To zoom in/out, again, as per the main work area use the magnifying glass buttons at the bottom-right of both the Editor Grid and Value Grid (`Shift`-clicking or `Ctrl`-clicking if needed), or `Ctrl`+mouse wheel.



"Magnifying glass" zoom buttons, situated towards the bottom-right of the MIDI Editor window

All MIDI editing operations are available and processed in real-time. Several MIDI Editor instances may be active and open at any one time, provided each one refers to its own clip. All MIDI content may be saved as standard MIDI files (SMF) Format 0 (all tracks reduced to one track, preserving channel assignment data) or Format 1 (multi-track). The format for the SMF files may be set in **View** / **Options…** / **MIDI**.

To create new MIDI data:

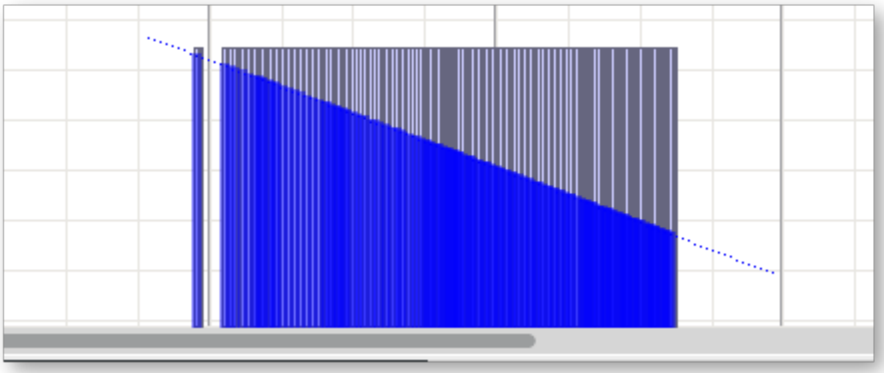
1. Create a destination track, if you don't have one already

- Right-click within the track and choose **Clip** / **New**. The MIDI Editor window will appear
- Click the *Edit on* (pencil) icon in the menu bar and draw notes in the upper pane of the grid. You can also click the *Edit Draw* (pencil with wavy line) icon to draw multiple notes "free-hand" by dragging the mouse
- In the bottom *Value Grid* pane you can adjust a variety of parameters. The default view shows *Note Velocity*, but this can be changed via the *Value type* and *Parameter type* combo boxes just below the pencil icons. Drag the bars up/down to increase/decrease their values, or click in a blank area (with one of the Edit modes enabled) to add new data

Below is a summary of the MIDI Editor's main tools:

- *Arrow* - select/move notes in the Editor Grid (or values in the Value Grid) and increase/decrease their length. To select all the notes of a particular pitch, left-click the desired key of the piano keyboard on the left hand side of the window and drag the cursor right, into the editor area
- *Edit on* (pencil) - draw notes or values, with their spacing determined by the *Snap/beat* setting (see below). You can also hold down Ctrl or Shift in this mode to put the Edit cursor into select mode temporarily - this is useful for quickly switching between inputting and selecting notes/information, as it avoids having manually to switch between the Edit and Arrow (described above) cursor modes
- *Edit Draw* (pencil with wavy line) - dynamically draw multiple notes/values with a click-and-drag of the mouse. This is, again, governed by the *Snap/beat* setting, but the faster you move the cursor the fewer the lines that will be drawn
- *Note Type* - "Note On" indicates that the notes you draw in the grid represent the length of the note. "Key Press" indicates that the notes represent a MIDI event without a defined duration, such as triggering an external MIDI sequence to begin playing
- *Value* - determines the information shown in the Value Grid. This may be note velocity, program changes, pitch bend or other extra MIDI data
- *Parameter type* - only available when certain *Values* are selected, such as *Controller*. Provides further information for the *Value* setting
- *Snap/beat* (musical note) - determines the resolution when drawing and editing notes/values. In a 4/4 session, for instance, a setting of *Beat* would cause a click in the Editor Grid to produce a crotchet (quarter note), *Beat/2* would produce a quaver (eighth note) and so on. Regardless of the default note length, a note can be lengthened/shortened by clicking and dragging one of its edges to increase/decrease its length. The resolution of such changes is also determined by the *Snap/beat* setting
- *Value ramp* - allows you to draw a continuous, linear slope of MIDI events in the Value Grid. This can be useful for operations such as creating crescendos/diminuendos or adjusting pitch bend

To use the value ramp: 1. Choose your desired *Snap/beat* setting 2. Enable *Edit on* and *Edit Draw*, then draw a series of events in the Value Grid (lower pane) 3. Hold down Shift or Ctrl (or switch to *Edit off*) and highlight these events 4. Disable *Edit Draw* (leaving *Edit on* enabled), then hold down the left mouse button and drag the dotted line over the events to set the angle of the slope. You can not only shave the tops off the bars but extend their heights too. You should see something like the below as you drag



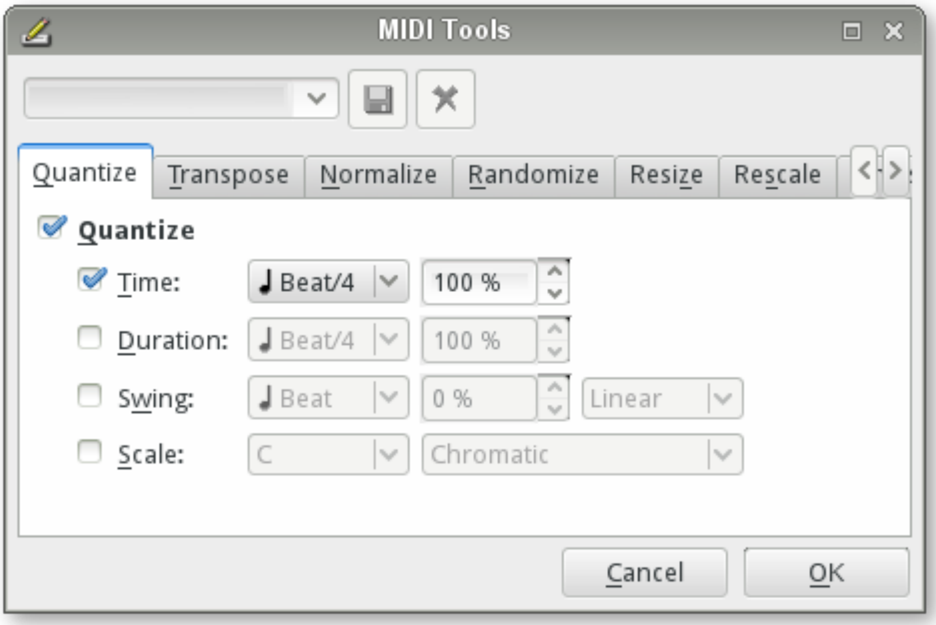
Adding a diminuendo using the value ramp

You can also use the tools detailed in Additional MIDI Editor Features with MIDI events such as these. For example, if you wish to make your diminuendo longer/shorter you can do so easily using the *Rescale* tool. With the MIDI events highlighted, right-click and choose **Tools** / **Rescale**, then adjust the *Time* percentage value accordingly.

4.10.5. Additional MIDI Editor Features

In addition to input and editing tools, the MIDI Editor features functions such as *Quantization*, *Transposition*, *Timeshifting* and more to make it easy to modify your composition. To access these features, open the **Tools** menu via the MIDI Editor toolbar, or by right-clicking within the window. If no notes are selected, the Tools will be unavailable; select a note or block of notes with the *Arrow* tool to make them available. Many of these tools are also very useful for adjusting MIDI event data in the Value Grid.

- *Quantize* - allows you to automatically structure notes in stricter uniformity with your time signature. This is especially helpful if you've played the notes in and were not perfectly on the beat. You can quantize notes such that the notes occur on the beat (or divisions thereof); such that the durations of notes are extended to be on the beat (or divisions thereof); such that notes play with some degree of "swing"; and even such that the notes played match a specific scale, from Minor to Major to the extremely obscure
- *Transpose* - allows you to transpose the pitch of, or shift the timing of, a note/notes, or reverse a sequence of notes
- *Normalize* - adjusts note velocity by either a percentage or an absolute value. A percentage value will adjust the velocity of the notes equally by the percent given in relation to the original velocity, while an absolute value adjusts velocity of each note to that value regardless of original velocity
- *Randomize* - provides random changes to the Note, Time, Duration, or velocity (Value). You may adjust how extreme the changes will be with percent values
- *Resize* - allows you to control the duration of notes to any number of beats (or divisions thereof), or the Velocity to any value
- *Rescale* - changes the selected notes by some percentage. You may alter the time that the notes are triggered, the duration for which they sound and the velocity at which they are played
- *Timeshift* - alters the timing of the selected notes on a curve, such that the acceleration from the beginning note to the ending note is either increased or decreased



☑ Quantize

☑ Time: Beat/4 100 %

□ Duration: Beat/4 100 %

□ Swing: Beat 0 % Linear

□ Scale: C Chromatic

The MIDI Editor's Tools menu

In order for Qtractor to know how to accelerate the timing of the notes, you must define a range of time for the acceleration to occur. Do this with the *Edit-head* and *Edit-tail* markers (the blue markers in the timeline above the MIDI Editor grid). Mark the out point (tail) of your range by right-clicking on the timeline, and then set the in point (head) of the time range by left-clicking somewhere to the left of the original marker. Once the range is defined, select the notes you wish to timeshift with the Arrow tool. Open the *Tools* menu and select *Timeshift*. Use the slider to cause either an acceleration, or a deceleration, in your selected notes over the course of the defined time range and then click **OK** to commit the change.

The **View** menu provides customization for how the MIDI Editor is laid out and how the notes are presented to you. The default layout is clean and pleasingly minimalist, but take a look at some of the other options to see what works best for you:

- *Menubar* - turns off the top menu bar; use `Control+m` to toggle it on or off

- *Statusbar* - turns off the status bar at the bottom of the window

- *Toolbars* - defines what information is visible in the window. Choose from *File* (file opening and saving icons), *Edit* (cut, paste, undo), *View* (preview modes, snapping, quantization), *Transport* (fast forward, rewind etc.), *Scale* (key signature and scale information) and *Thumb* (thumb view at the bottom of the window)

- *Windows* - toggles on or off event information, a panel which provides detailed information on each MIDI event in the MIDI Editor grid

- *Tool Tips* - toggles whether tool tips are visible

- *Note Duration* - defines whether the note's duration is reflected in the length of the velocity bars at the bottom of the grid

- *Note Color* - assigns differing colours to each note; if this is not active, all notes appear as one colour

- *Value Color* - assigns the colours of the corresponding notes to the velocity bars; helpful when you have many overlapping notes but want to adjust their velocities

- *Drum Mode* - displays MIDI Note On events as diamonds instead of rectangles

- *Zoom* - zoom in or out on the grid

- *Snap* - define what division of the beat your notes snap to, or turn snapping off entirely

- *Scale* - change the key signature or type of scale being used

- *Refresh* - causes the MIDI Editor to redraw in the event of latent images

- *Preview Notes* - toggle whether or not you hear the notes of the scale as you input or move notes on the grid

- *Follow Playhead* - define whether the MIDI Editor grid scrolls with the playhead when you are playing the track

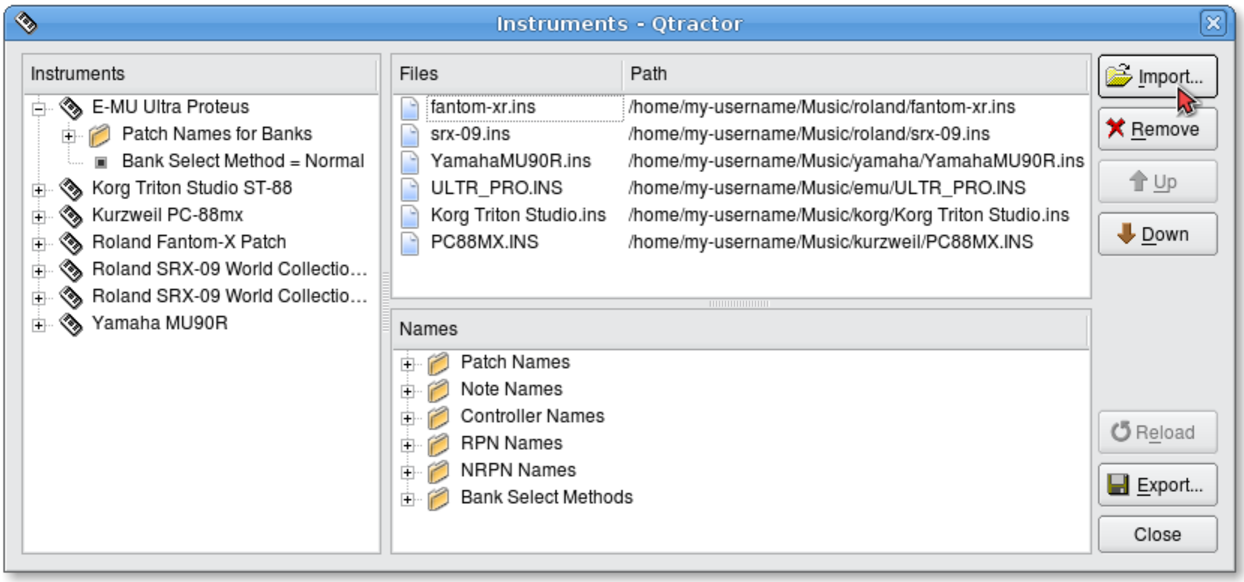
4.10.6. MIDI Instruments

There are many MIDI hardware tone generators available from a variety of manufacturers such as Yamaha, Roland and Korg. In Qtractor, information about the sound patches and sound banks of these tone generators is obtained from “instrument definition” (`.ins`) files. Qtractor supports instrument definition files used by the Cakewalk/Sonar MIDI sequencer software, offering a convenient MIDI bank-select/program-change mapping for existing MIDI instrument patch names and easier, intelligible selection of MIDI track channels. Files for many popular MIDI tone generators and synthesizers can be downloaded from the Cakewalk site.

`.ins` files are imported via the **View** / **Instruments…** window. You can also import MIDI Name XML files (`.midnam`) in the same way. Once you’ve imported a file, select your desired patch via the **Bank** and **Program** options in the **MIDI / Instrument** area of the **Track** window. After clicking **OK**, the information will appear in the main view. Instrument patches can also be assigned via several other menus:

- **Track** / **Instrument** in the main toolbar
- **File** / **Track** / **Instrument** in the MIDI Editor
- Right-clicking a track in the left pane of the main view and selecting **Instrument**
- In the GUI of the plugin itself

When working with soundfonts (for example, via the FluidSynth DSSI, or Calf Fluidsynth LV2, plugins), you can also use the **View** / **Instruments…** window to import their patch names. Simply select the desired `.sf2`/`.sf3` file and its patch names will be imported. For DSSI plugins, changing the patch information in the **Track** window should update the relevant information in the plugin’s GUI automatically and vice-versa. However, this may not be the case for LV2 plugins (due to the LV2 spec, rather than any issue with Qtractor) - if there is a mismatch between the patch set in the plugin and the information in the **Track** window, upon re-loading the session the patch in the track may override that of the plugin. To avoid this behaviour, in the **Track** window set **MIDI / Instrument** to *(No instrument)* and **Bank** and **Program** both to *(None)*, then perform all patch selections directly in the plugin.



Files	Path
fantom-xr.ins	/home/my-username/Music/roland/fantom-xr.ins
srx-09.ins	/home/my-username/Music/roland/srx-09.ins
YamahaMU90R.ins	/home/my-username/Music/yamaha/YamahaMU90R.ins
ULTR_PRO.INS	/home/my-username/Music/emu/ULTR_PRO.INS
Korg Triton Studio.ins	/home/my-username/Music/korg/Korg Triton Studio.ins
PC88MX.INS	/home/my-username/Music/kurzweil/PC88MX.INS

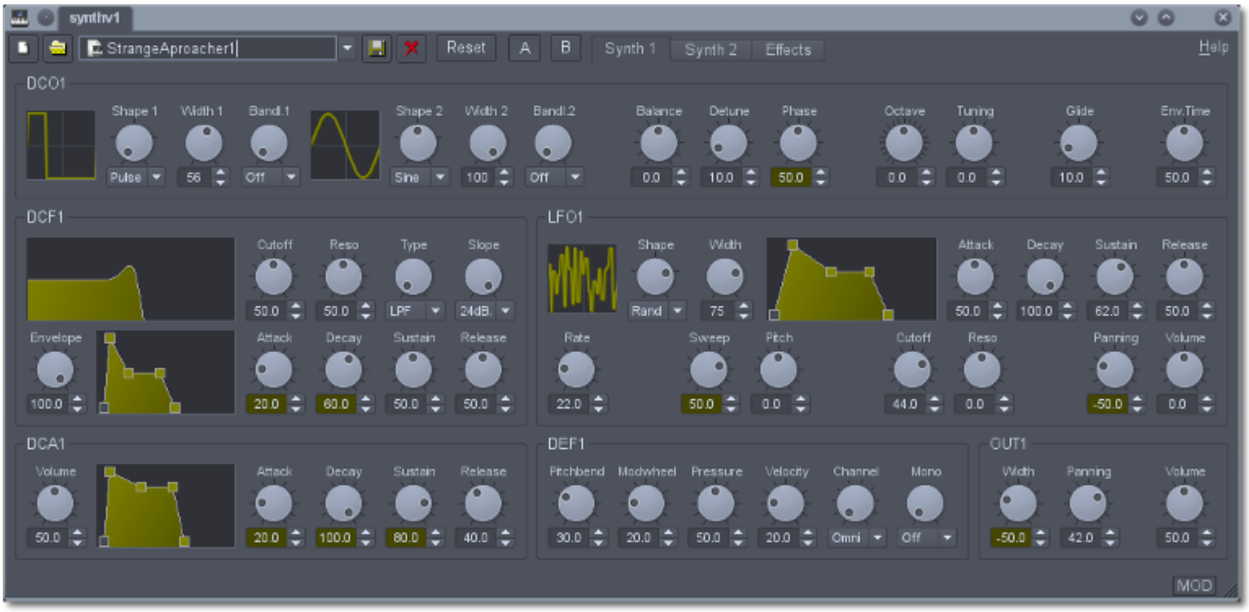
Instruments window, where MIDI instrument definition files, MIDI Name XML files and soundfont patch names can be imported, exported and moved

4.10.7. Soft Synths

Strictly speaking, software synths, or “soft synths”, are not part of Qtractor. However, it is common to use them within Qtractor. Acquiring free soft synths online is a simple matter of locating them via the project’s homepage, your distribution’s repositories, or sites such as sourceforge.net.

Soft synths for GNU/Linux come in a few varieties:

- Standalone synth applications that launch separately from Qtractor, but which plug into JACK so that they can be used as sound sources within Qtractor. One such example is Qsynth, written by the same programmer as QJackCtl and Qtractor
- DSSI Plugins, which are launched and controlled entirely from within Qtractor. For example, Fluidsynth DSSI
- LV2 Plugins, which are launched and controlled entirely from within Qtractor. For example, synthv1
- VST Plugins. The format from Steinberg does in fact support Linux technically, but there are few Linux-native VST synths



synthv1 - a soft synth available in both standalone and LV2 plugin forms

Once your synth of choice is installed, the method of launching it will depend on what form it is. If it’s a standalone application, it can be launched like any other application on your system. If it’s a plugin for Qtractor, launch Qtractor and create a new MIDI track, then choose your soft synth for the MIDI Instrument, as described here.

4.10.8. Soft Synths as Plugins

Using soft synths as LV2 or DSSI plugins is probably the easiest model, as it does not require any additional routing through JACK. The synth is integrated into Qtractor’s interface, the sound is directed to the appropriate track, and the fact that you are using a separate application is almost entirely abstracted away from you. For more information on using plugins, see the Plugins section.

Most soft synths have some level of control over the sounds they produce. To edit a synth’s properties:

1. Choose **View** / **Windows** and open the **Mixer**

2. In the panel towards the top, locate your synth and make sure it’s on (designated by a green lamp on the left of its name). If the lamp is black (synth turned off), click it so that it turns green

3. Double-click on the soft synth to open its settings window and change the desired parameters. The actual information presented will differ depending on the synth

Whether you use LV2, DSSI or VST plugins, the process for using a soft synth (or plugin effect) is the same. In addition, all parameters provided by the interface can be automated, enabling attributes like LFO or resonance to change over time during playback. For more information on this see the Automation section.

4.10.9. Stand-alone Soft Synths

The other kind of software synth you might use is a stand-alone synthesizer, such as Qsynth. To use such a synth, launch QJackCtl and Qtractor, then launch the synth the same as you would any other application on your system.

Once you've launched your synth it will become a source for audio and a destination for MIDI signals in Qtractor. However, you must first verify that your settings are correct:

1. Open the **View** / **Windows** / **Connections** menu

2. The **Audio** tab contains all possible sources of audio. If you've launched QSynth, for example, it will be listed in the left column. Also listed will be Qtractor itself, since it too produces sound. In the right column are the audio destinations available. At the very least, you should see Qtractor's *Master In* and your computer speakers (labelled as *System > playback*). By default, the sound from Qtractor's *Master Out* is patched to *System > playback* so that you can hear everything that Qtractor's Mixer manages

3. This is where the workings of your own studio set-up should be considered. The audio from your synth can logically be patched into either Qtractor's *Master In*, so that its sound can be recorded into an audio track, or *System > playback*, if you have an external mixer or recording device that will be recording it. To patch the sound from the synth to its destination, click the left channel of the synth output (in the left column), then the left destination channel (in the right column), then the **Connect** button. Repeat this for the right channel if stereo is required

This takes care of routing the audio; now you must route the MIDI signals controlling the synth. Here again, there is no "right" way to route the flow of data - you must be somewhat familiar with the equipment and workflow you are implementing and decide for yourself how you want it all to work together. However, a general method of working is described below:

1. Open the **View** / **Windows** / **Connections** menu

2. The **MIDI** tab contains all possible sources of MIDI signals. At the very least, your stand-alone soft synth will be listed in the left column. If you have a USB MIDI controller, it too will be listed. Qtractor itself and a *MIDI Through* channel will also be listed. In the right column are the sound destinations available. At the very least, you should see Qtractor's *Master* and *MIDI Through*, together with any soft synths you are running

3. Connect the MIDI signal from Qtractor (in the left column) into your soft synth (in the right column) so that the data in your MIDI tracks will trigger the synth's sounds

4. If you are using an external USB MIDI keyboard controller, you will also likely want to route its signal into Qtractor's MIDI input so that you can input notes directly into your Qtractor MIDI tracks

4.11. Working with Audio

4.11.1. Audio Summary

There are two typical scenarios for dealing with audio in Qtractor and, indeed, for most digital audio workstations:

- Import existing audio files from your hard drive, a recording device, or a loop collection

- Record audio into Qtractor via your computer's built-in microphone or an external microphone

These methods of acquiring sound for your work can be, and very often are, combined to create a richer audio production.

4.11.2. Importing Audio Files

The easiest way to get sound into Qtractor is to import an audio file from your hard drive. Whether you are using sampled loops to construct a new musical piece, or importing a performance transferred from a recording device, importing audio in this way does not involve recording sound directly into Qtractor.

To bring audio into your project file, right-click in the **Files** pane and select **Add Files** or use `Ctrl+F`. Choose which file or files you wish to import from the file chooser window that appears. To place an audio file in a track, drag and drop it from the Files panel into the workspace. You can add it to an existing track, or drop it directly into an empty workspace area and a new track will be created automatically.

Note that when importing an mp3 file and adding it to the timeline, you may find that the waveform shows extraneous silence at the end. This is a known problem with mp3 files and it is especially prevalent with those encoded in VBR (variable bit rate) mode. To fix this, just delete the file from the timeline, then re-drag it back onto the timeline and the waveform should display correctly.

4.11.3. Recording Audio

Recording audio into Qtractor requires a microphone and at least one audio input channel. Many laptops and webcams have built-in microphones, so, in theory, you could use this as an input source, but for best quality, purchase an external microphone and use it as your audio capture device.

For simultaneous multi-track recording, rarely will the sound card that was bundled with your computer be sufficient. Almost all sound cards embedded on motherboards are set to mix the input signals down to a stereo mix. If you wish to record three separate musicians at the same time, each to a separate track in Qtractor, you will require a sound card with separate, dedicated inputs.

The way that Linux displays available sound inputs and outputs can be daunting at first, until you understand the logic behind it. A Linux system displays sound devices the same way it displays hard drives and available network interfaces: the first device (regardless of actual inputs or outputs available via that device) is labeled `hw0`, the next `hw1`, the next `hw2`, and so on. It is safe to assume that `hw0` would be the built-in sound card on the system; being embedded in the motherboard would certainly qualify it as being the first available sound device.

So, `hw0` represents, in almost every case, your built-in sound card. `hw1` might represent, for instance, a webcam that you keep plugged into your desktop, and `hw2` could represent, perhaps, a USB microphone or a USB interface that you've plugged in. You can usually determine which device is which by looking at the vendor name associated with the `hw` labels; for example, if you have two devices plugged into your system and one is labeled `Blue` and the other `H4`, then you would know from these terms that one is your Blue USB microphone and the other your Zoom Studio H4n.

With regard to outputs, these can sometimes be confusing due to the many possible ways you may wish to output your sound. Typically the stereo mix of your system sound is available as the first two output devices. If you have more than just two speakers and you wish to split your sound to each, then utilize the outputs labeled appropriately (Front, Center and so on). If the sound input and output labels confuse you, take a few minutes to learn them by playing sound on your computer and plugging speakers into each output on your computer. For the inputs, plug a microphone into your different inputs and see where they are received and how they are labeled. It won't take long before you understand the logic behind the labels, and you'll be able to use Qtractor all the more fluidly.

4.11.4. Recording from Line-In

There are many kinds of microphones, each intended for certain kinds of sounds and situations, but on a purely technical level, without the question of aesthetics and microphone design, there are only about three scenarios you will encounter:

- Microphones with an 8th-inch (also called a mini) jack

- Microphones with a quarter-inch or XLR jack

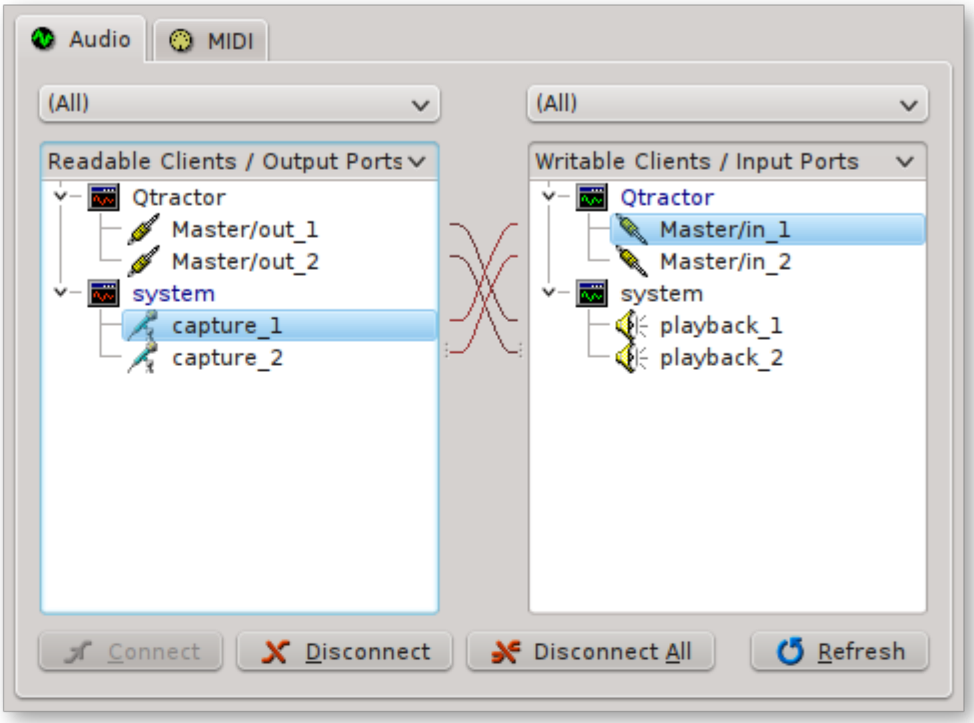
- Microphones with USB connectors

The recording process is different, depending on your input type.

If your microphone has an 8th-inch jack, then you can plug it directly into the line-in of your computer. No external sound interface is required. Your built-in sound card is JACK's default input, so no changes are necessary. If your microphone can easily and cleanly adapt to the standard 8th-inch input with a cable or a simple plug adapter, then you can use this method of recording, as well.

To record from the line-in of your computer into Qtractor:

- Go to the **View** menu and select **Windows** / **Connections** to verify that the *Capture* devices on your System are routed to the *Master/In* of Qtractor and that the *Master/Out* channels are routed to the *Playback* channels of your System



Connections window, set up to capture audio

- Go to the **Track** menu and choose **Add Track** or use `Ctrl+Shift+N`. In the **Track** dialogue box, give the track a name, set the *Type* to *Audio*, and set the *Input/Output* to *Master*. Click the **OK** button to proceed

- Arm the new track for recording by clicking the “R” button in the track listing on the left of the main Qtractor window. This sets the destination for the recorded sound

- Click the *Record* button in the top toolbar

- Click the *Play* button in the top toolbar

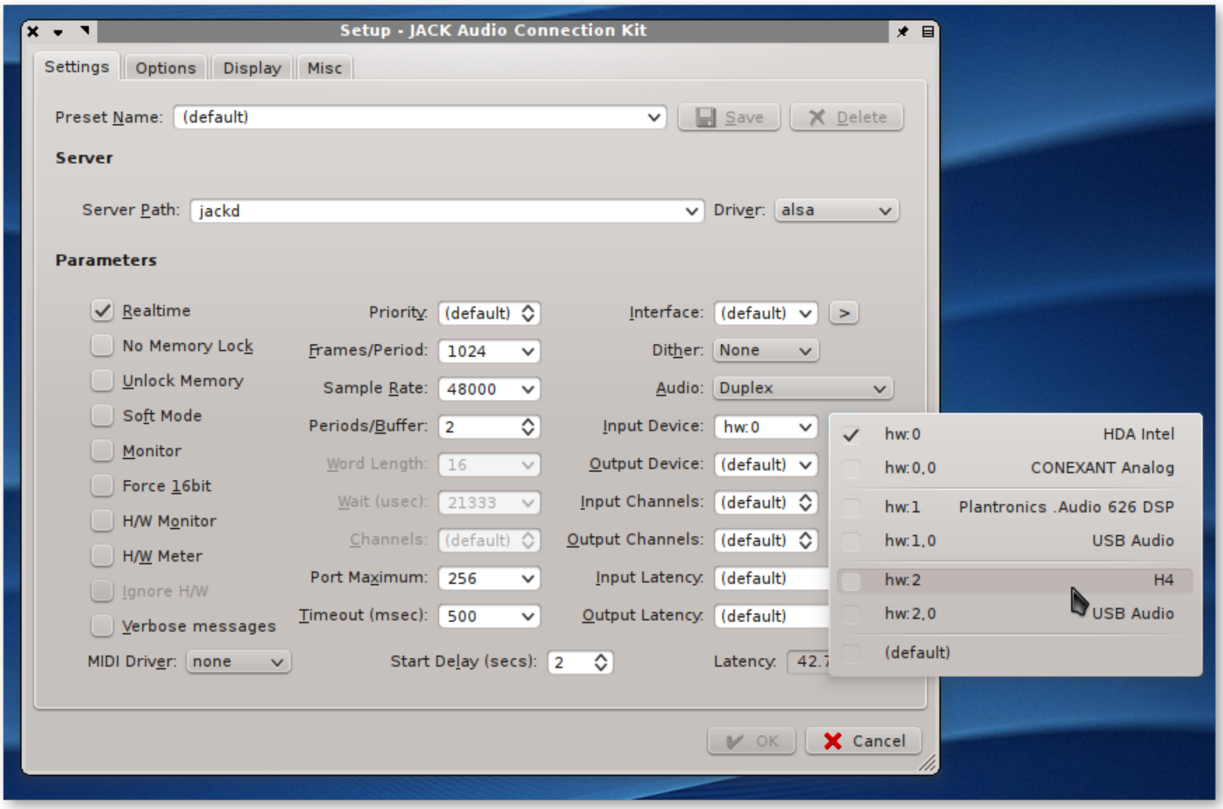
4.11.5. Recording from USB

A USB microphone plugs directly into the USB port of your computer. If you input sound through USB, then (obviously) you are utilizing a different interface than your computer’s built-in sound card. This must be set via QJackCtl for appropriate sound routing to occur:

- In QJackCtl, stop the sound server by clicking the Stop button

- If you have not already plugged in your USB audio interface or USB microphone, then do so. Make sure it’s on

- Click the **Setup** button. In the **Settings** tab, locate the *Input Device* setting and click the “>” button to see your choices



Setup - JACK Audio Connection Kit

Preset Name: (default)

Server

Server Path: jackd

Driver: alsa

Parameters

☑ Realtime
□ No Memory Lock
□ Unlock Memory
□ Soft Mode
□ Monitor
□ Force 16bit
□ H/W Monitor
□ H/W Meter
□ Ignore H/W
□ Verbose messages

MIDI Driver: none

Priority: (default)
Frames/Period: 1024
Sample Rate: 48000
Periods/Buffer: 2
Word Length: 16
Wait (usec): 21333
Channels: (default)
Port Maximum: 256
Timeout (msec): 500
Start Delay (secs): 2

Interface: (default)
Dither: None
Audio: Duplex
Input Device: hw:0
Output Device: (default)
Input Channels: (default)
Output Channels: (default)
Input Latency: (default)
Output Latency: (default)
Latency: 42.7

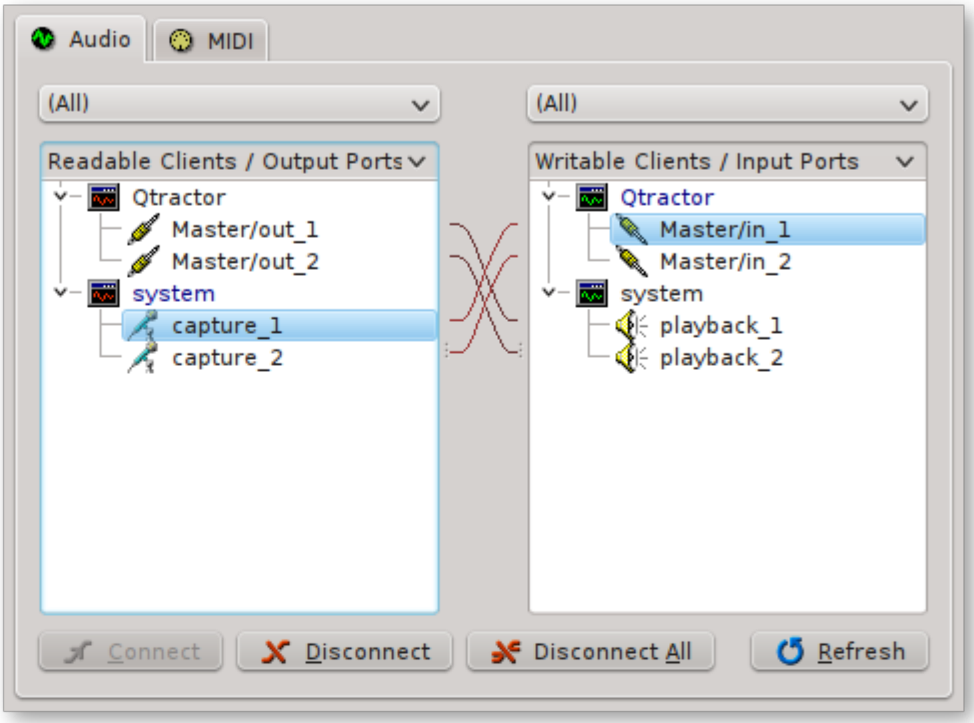
✓	hw:0	HDA Intel
	hw:0,0	CONEXANT Analog
	hw:1	Plantronics .Audio 626 DSP
	hw:1,0	USB Audio
	hw:2	H4
	hw:2,0	USB Audio
	(default)	

QJackCtl settings window showing Input Devices

- Click the **Start** button to activate

Now your USB interface, whether it is a single USB microphone or a 4-channel Audio-to-USB conversion box, is managing your input sources. Plug your XLR microphone into your USB interface, set the interface's input source as your XLR jack, and then create a new track in Qtractor and begin recording:

- Go to the **View** menu and select **Windows** / **Connections** to verify that the *Capture* devices on your System are routed to the *Master/In* of Qtractor and that the *Master/Out* channels are routed to the *Playback* channels of your System.



Connections window, set up to capture audio

- Go to the **Track** menu and choose **Add Track** or use `Ctrl+Shift+N`. In the **Track** dialogue box, give the track a name, set the *Type* to *Audio*, and set the *Input/Output* to *Master*. Click the **OK** button to proceed

- Arm the new track for recording by clicking the “R” button in the track listing on the left of the main Qtractor window. This sets the destination for the recorded sound

- Click the *Record* button in the top toolbar

- Click the *Play* button in the top toolbar

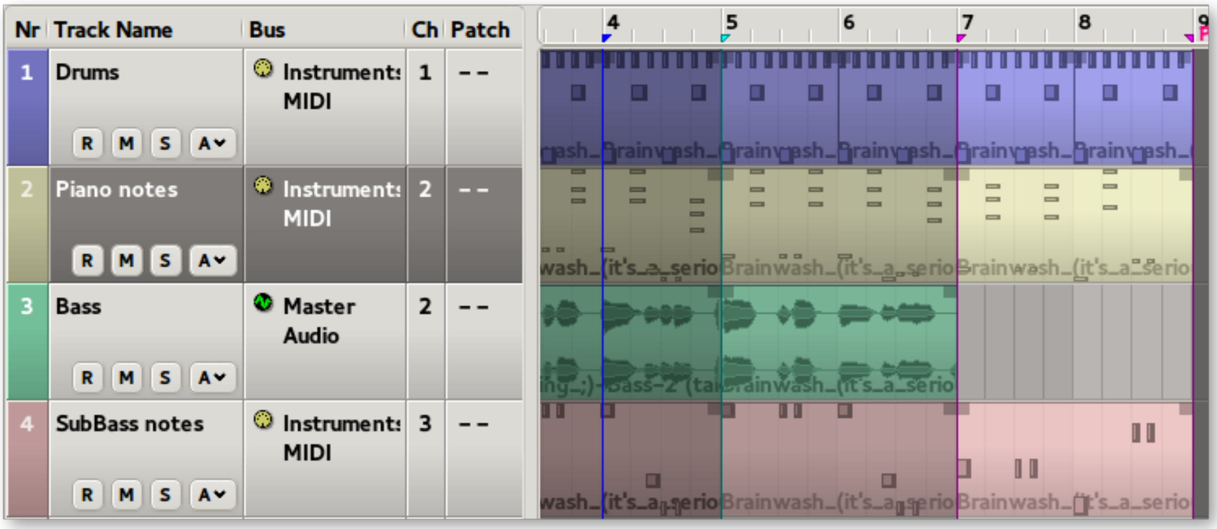
4.11.6. Recording Takes

Aside from one-take recordings, you can create multiple, continuous takes by using the Loop recording mode. To enable the Loop recording mode, go to **View** / **Options…** and, on the **General** tab, set *Loop Recording mode (takes)* to either ‘First’ or ‘Last’. This will control whether Qtractor chooses the first or the last take by default.

After you have enabled the Loop recording mode, select a loop range. To do so, position the Edit markers by left-clicking on the timeline where you want your loop range to start and right-clicking where you want it to end. Now select **Transport** / **Loop** (or press `Ctrl+Shift+L`). This will create a loop range out of your edit range and enable the loop mode as well. Just arm the track you want to record to, press *Record*, then press *Play*. Qtractor will start recording, but when it reaches the end of your loop range it will wrap around and start recording a new take of your loop range. You can record as many takes as you wish and decide later which one you like the most.

After having recorded some takes, select your new clip and go to **Clip** / **Takes** / **Select** to choose between the different takes. You can also press `Shift+T` to iterate through the recorded takes.

It is also possible to combine Punch in/out recording with looped takes recording. Just create a new Loop range as before, then create a Punch in/out range inside your loop range. Recording now starts as soon as the playhead enters your Punch in/out range. Recording stops once the playhead leaves the range, but it will start again, recording another take, as soon as your loop starts over and reaches the punch in point again.



Nr	Track Name	Bus	Ch	Patch
1	Drums	Instrument: MIDI	1	- -
2	Piano notes	Instrument: MIDI	2	- -
3	Bass	Master Audio	2	- -
4	SubBass notes	Instrument: MIDI	3	- -

Punch in/out range (purple markers) inside a loop range (green markers)

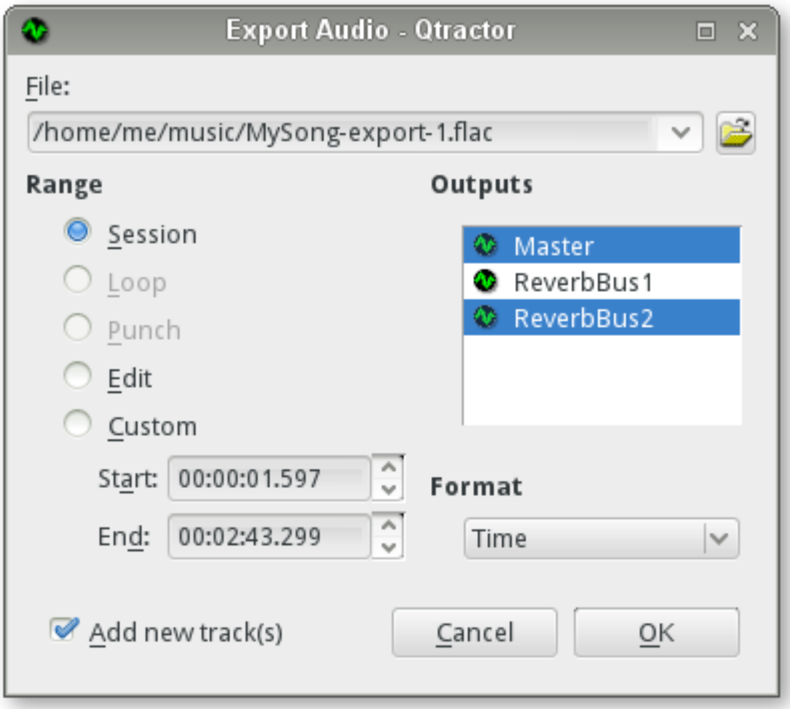
Takes are implemented in such a way that they are appended to each other. In other words, multiple recorded takes will result in a single clip that contains all takes appended to each other. By having activated the takes mode, Qtractor will show you only a section of this clip with the size of the selected loop range. If you go to **Clip** / **Take** / **Select** and choose ‘None’, Qtractor will ‘unfold’ your takes and show the recorded clip at its full length.

Once you know how takes are implemented, you can create them from any clip you like. For example, select a clip and place the edit markers within a fraction of your clip. If you have a clip that is four measures long, place the first edit marker at the beginning of the clip and the last at the beginning of the second measure. Now go to **Clip** / **Take** / **Range**. In the dialog box which appears, choose ‘Edit’. The box on the right of this will show the resulting number of takes - in this case it will create four takes (one for each measure). Just click on the take you want to select. As you can see, takes can be created with a variety of ranges - just use the one that fits your needs best.

Finally, be aware that take state information is kept *only for the original recording track location*. If you change or cut any of the folded clip-takes all take state information will be lost (though such operations can be undone, as normal). If you move a clip to another area of the same track, then select a different take, the clip will “snap” back to its original position. If you move a clip to another track, take state information will be lost. Note that even if information has been lost in whatever way, you can still drag the clip’s edges to access any hidden takes manually.

4.12. Audio / MIDI Export

All or part of the session may be exported to one audio or MIDI file. To export a file, select **Track** / **Export Tracks** / **Audio…** or **MIDI…**, or right-click in the tracks area of the main view to access the same menu. This will open either the **Export Audio** or **Export MIDI** window.



Export Audio - Qtractor

File: /home/me/music/MySong-export-1.flac

Range

● Session

○ Loop

○ Punch

○ Edit

○ Custom

Start: 00:00:01.597

End: 00:02:43.299

Outputs

Master

ReverbBus1

ReverbBus2

Format: Time

☑ Add new track(s)

Export Audio window, where the tracks in a session can be exported as a single audio file in the format previously specified by the user in ***View / Options…***

The **Range** options enable you to specify how much of your session to export. You may export the whole **Session**; any **Loop**, **Punch** or **Edit** range you have previously denoted with its markers; or an arbitrary **Custom** period (the format of which can be expressed in **Frames**, **Time** or **BBT**).

The **Outputs** box allows you to choose which buses to export. `Ctrl`-click to select more than one. Note that when exporting multiple buses, because the buses' outputs are added together, the exported file can become clipped. To address this, you may choose to:

- Lower the level of the output buses, so that their combined output level does not exceed 0dBFS (monitoring this combined output via an external meter if necessary)
- Export to a float sample file format, by selecting one via the **Capture / Export** section of the **View** / **Options…** / **Audio** menu. This way, even if the exported file does become clipped, no data will be lost. You can then simply reduce the level of the file later using an editing tool, such as Audacity

Enabling the **Add new track(s)** option will automatically import a copy of your exported file into a new track below the currently selected one. This can be useful to "freeze" all or part of your session, mixing down several tracks to one new track in order to lighten CPU load or simply to tidy your workspace up.

MIDI export pertains to MIDI material only and results in the merging and concatenation of the selected MIDI tracks and clips into a single file of either SMF Format 0 (all tracks reduced to a single track) or SMF Format 1 (multi-track).

Audio export is implemented through the special JACK *freewheel* mode and is thus faster than real-time, resulting in a complete and exact mix-down of the selected audio material into a designated audio file of the selected format (wav, aiff, flac, au, ogg etc.). It is also possible to export the output of MIDI tracks directly as audio (with standard audio tracks also included in the exported file, if desired) using the same **Export Audio** window, provided that the MIDI tracks' audio is generated by plugin instruments. For MIDI tracks which are used to send data to external instruments driven by JACK-MIDI or ALSA-MIDI (the output of these instruments then being routed back into Qtractor), it is necessary to "bounce" their output to audio tracks within Qtractor before export. The process of bouncing is covered in this How To.

The user-preferred format for exported audio and MIDI files can be set in the Options window (**View** / **Options…**).

4.13. Snapshots, Templates and Archives

4.13.1. Snapshots

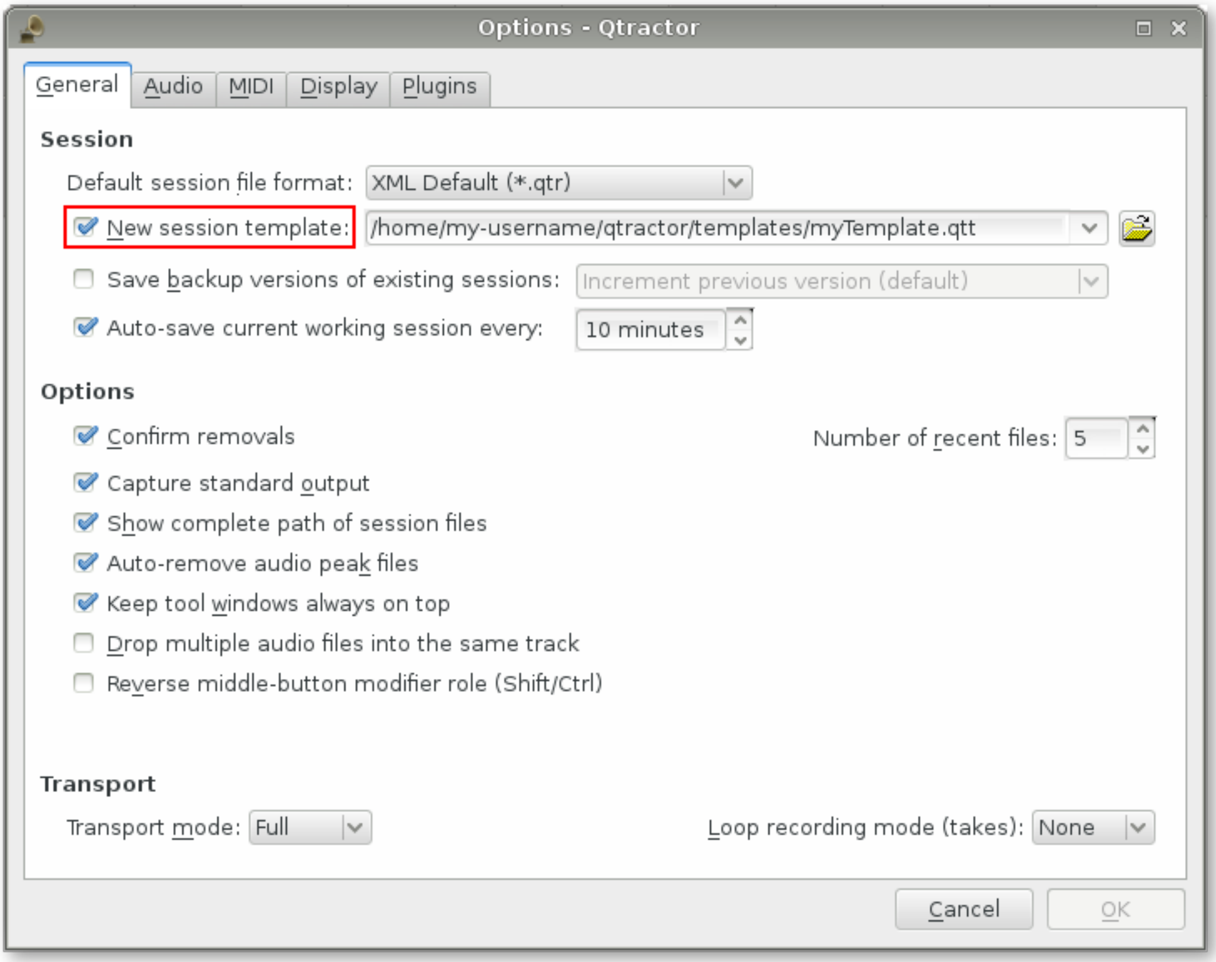
After the initial save of your project, you may find later that you want to save multiple "snapshots" of a session to compare it at different stages. To do this, click **File** / **Save as…** and Qtractor will add an incremental number to the end of your file name. Once saved, you can then switch freely between these snapshots by opening the relevant file.

Be aware that snapshots saved in the standard session file formats (**.qtr** or **.qts**, which are essentially the same) save only *references* to clips, rather than the clips themselves. Because of this, "destructive" edits (such as changing the notes of a MIDI clip) will be reflected in all snapshots. For example, if you have a MIDI clip in snapshot 1, then create snapshot 2 and make changes to the notes in that same clip, when you re-load snapshot 1, the changes you made in snapshot 2 will now be present in snapshot 1 as well. If you wish to avoid this and preserve all of your clips in a separate snapshot, use the **.qtz** archive format, which will safely bundle its own versions of all the clips (see also 4.13.3. Archives below).

4.13.2. Templates

Templates are an extremely useful time-saving feature, which enable you to restore many settings automatically for use in a new project, thereby eliminating the need to re-configure tracks, buses, connections etc. each time. To create a template, once you have your session set up the way you want it, choose **File** / **Save As** and save the file in the **.qtt** format. This file can then be loaded in the same way as a standard session. Template files reproduce all the original session's settings, without any MIDI or audio data, leaving you ready (or at least part of the way ready) to begin a new project. Exactly what you include in your template will depend on your needs, but you'll begin to discover what these are as you use Qtractor. You can save any number of templates to cover whatever kind of projects you need.

If you find that you're using the same template often, you may choose to have it auto-loaded for every new session. To enable this, choose **View** / **Options…** and, in the **General** tab, specify your template in **New session template**.



Session

Default session file format: XML Default (*.qtr)

☑ New session template: /home/my-username/qtractor/templates/myTemplate.qtt

□ Save backup versions of existing sessions: Increment previous version (default)

☑ Auto-save current working session every: 10 minutes

Options

☑ Confirm removals

Number of recent files: 5

☑ Capture standard output

☑ Show complete path of session files

☑ Auto-remove audio peak files

☑ Keep tool windows always on top

□ Drop multiple audio files into the same track

□ Reverse middle-button modifier role (Shift/Ctrl)

Transport

Transport mode: Full

Loop recording mode (takes): None

The Options window

4.13.3. Archives

Qtractor has a zipped archive format, **.qtz**, which collects together all the information needed to reproduce the session elsewhere, such as settings, connections, plugins (LV2 only) and all MIDI and audio files (minus any which have been disassociated with the session via the **Cleanup** option in the **Files** pane). This format can be used to share your Qtractor project with other people, or simply to tidy up a session which contains extraneous files.

Note that even if you save an archive with a different file name the original session's Properties will remain as they were, meaning that any clips subsequently recorded will still bear this old name in their file name. For example, if your original session has "Song_A" as the Name in its Properties and you save an archive of, then work on, the new session as "Song_B", all new clips will continue to be called `Song_A-trackName-1.mid` etc. If you want the new session name to be reflected in the file names of both new and existing clips, open the **File** / **Properties…** / **Session** menu and update the name in the **Name:** box. Qtractor will then update the session's midi and audio clip file names to match this name.

4.14. Other Qtractor Features

4.14.1. Metronome

When recording audio or MIDI data in real-time, a metronome can be very useful for keeping your performance in time. Qtractor has both audio and MIDI metronomes. The audio metronome generates its sound within Qtractor; the MIDI variant requires an external sound source, such as a standalone soft synth. You can even use both metronomes at the same time if desired.

To set up a metronome, open the **View** / **Options…** menu.

- **Audio metronome** - select the **Audio** tab. In **Metronome**, tick *Enable audio metronome* and point Qtractor to the files you wish to use for *Bar* and *Beat*. You can also adjust the *Gain* if necessary, plus create *Dedicated audio metronome outputs* (useful if you have a particular monitoring set-up in mind)
- **MIDI metronome** - select the **MIDI** tab. In **Metronome**, tick *Enable MIDI metronome*, select a *Channel* and specify the *Note (bar)* and *Note (beat)* you wish to use. You can also adjust the *Velocity* and *Duration* of these notes. The MIDI metronome triggers sounds via the *MIDI Master Out* bus by default, but if you'd prefer to use a separate bus you can enable *Dedicated MIDI metronome output* - this will then appear as *Metronome* in the MIDI tab of the Connections window. Once you've decided which bus to use, connect it up to your destination via the Connections window

If you're working with noticeable latency in your system, the *Offset (latency)* option, available for both audio and MIDI metronomes, allows you to compensate for this.

Percussion sounds are often employed for a metronome, though what you use is entirely your choice. For the audio metronome, if you have no suitable files to hand you can search the internet or use the ones from this thread on rncbc.org. For the MIDI metronome, if you want percussion sounds and you're using a General MIDI-compliant bank, set *Channel* to `10`, which is the channel for percussion.

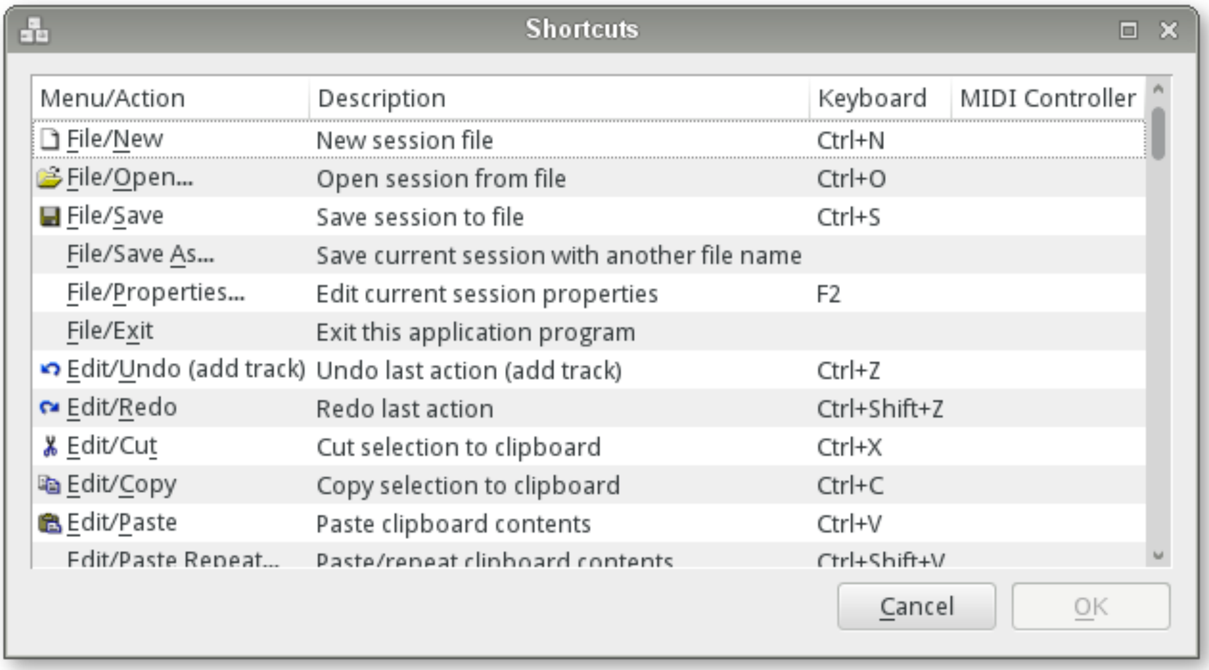
In order to hear either metronome while the transport is rolling you must turn it on. To do so, click the blue triangle icon in the top menu bar, or select **Transport** / **Metronome**.

The Metronome icon

4.14.2. Keyboard Shortcuts Editor

Keyboard shortcuts are useful for the power user, in that they provide a quick mechanism for performing often used commands quickly, without the use of your mouse. Shortcuts can also be assigned for MIDI Controllers.

Keyboard shortcuts may be customized to your preference by using the shortcuts editor. This editor may be found in the Help menu (**Help** / **Shortcuts…**). Note that there are two separate shortcuts windows - a general one, accessed from the main window and a MIDI Editor-specific one, accessed from the MIDI Editor window.



Menu/Action	Description	Keyboard	MIDI Controller
File/New	New session file	Ctrl+N	
File/Open...	Open session from file	Ctrl+O	
File/Save	Save session to file	Ctrl+S	
File/Save As...	Save current session with another file name		
File/Properties...	Edit current session properties	F2	
File/Exit	Exit this application program		
Edit/Undo (add track)	Undo last action (add track)	Ctrl+Z	
Edit/Redo	Redo last action	Ctrl+Shift+Z	
Edit/Cut	Cut selection to clipboard	Ctrl+X	
Edit/Copy	Copy selection to clipboard	Ctrl+C	
Edit/Paste	Paste clipboard contents	Ctrl+V	
Edit/Paste Repeat...	Paste/repeat clipboard contents	Ctrl+Shift+V	

Keyboard Shortcuts Editor window, available from the ***Help / Shortcuts…*** *menu item*

It is very straightforward in its use. Simply find the item you want to create a shortcut for, left-click in the shortcut cell, and choose the key you wish to be assigned to that function. You will then see whatever key or key combination you chose appear in the context. The “X” icon on the right of the window can be used to undo any changes or remove a default shortcut.

Take care when using shifted combinations, as if that combination produces a non-alphanumeric character it may not function as a shortcut. For example, `Shift+1` will likely end up as a symbol such as “!” in the menu and, upon attempting to use this combination as a shortcut, it will not work. `Alt` and `Ctrl`-modified shortcuts should, however, be fine.

Qtractor will display a warning message if you try to duplicate a shortcut in the same menu, but not if you use the same shortcut in both the main and MIDI Editor menus. Instead, a warning message will appear in the Messages window. This warning is driven by the Qt framework and is, unfortunately, outside of Qtractor’s control.

Manual - 5 Qtractor Menus

Manual - Table of Contents

5. Qtractor Menus

5.1. Main Workspace

File

- **New** – create a new session project
- **Open…** – open a previously created session project
- **Open Recent** – access a list of several of your last used session projects
- **Save** – save your current session project
- **Save As…** – save your current session project with naming conventions
- **Properties…** – open the session properties dialog
- **Exit** – exit the program

Edit

- **Undo** – undo the last action
- **Redo** – redo the last action
- **Cut** – delete and copy an item to the clipboard
- **Copy** – copy an item to the clipboard
- **Paste** – paste an item from the clipboard
- **Paste Repeat…** – paste an item from the clipboard multiple times, with a user-definable period in between each
- **Delete** – delete the selected item
- **Select Mode** – select Clip, Range or Rectangle edit modes, or Automation
- **Select** – select All, None, Invert, Track, Track Range or Range
- **Insert** – insert a range or track range
- **Remove** – remove a range or track range
- **Split** – after selecting a Range or Rectangle, split according to this selection
- **Track** – add, remove and adjust the properties of tracks

- **Clip** – create a new clip, edit an existing clip, or import/export a clip

Track

- **Add Track…** – add either a new audio or MIDI track
- **Remove Track** – remove a track
- **Duplicate Track** – create a copy of the currently selected track, directly beneath it
- **Track Properties…** – open the track properties dialog
- **Inputs** – show current track inputs
- **Outputs** – show current track outputs
- **Instrument** – select the instrument name of a MIDI track, from those defined in the View / Instruments menu
- **State** – set current track state: Record, Mute, Solo, Monitor
- **Navigate** – navigate through tracks: First, Previous, Next, Last, None
- **Move** – change position of current track: Top, Up, Down, Bottom
- **Height** – adjust the vertical height of a track
- **Auto-Monitor** – toggle current track auto-monitoring
- **Auto-Deactivate** – deactivate plugins not producing sound
- **Import Tracks** – import audio or MIDI file(s) as new track(s)
- **Export Tracks** – export audio or MIDI track(s)
- **Automation** – automate the parameters of a track or its plugins

Clip

- **New…** – add a new clip to the currently selected track, bounded by the Edit-head and Edit-tail markers (blue vertical lines)
- **Edit…** – edit the currently selected clip
- **Take** – select from multiple takes recorded, or set a range to create a new one
- **Unlink** – if a MIDI clip is copied within the same track, any subsequent changes will be reflected in both the original and copied clips. This unlinks the clips, deactivating this behaviour
- **Record** – set a clip to be overdubbed on the next recording operation
- **Split** – split the clip at the position of the play-head (red vertical line)
- **Merge…** – merge two or more currently selected clips into one

Normalize – normalize the level of an audio clip to 0dB

-

- **Tempo Adjust…** – adjust the clip's tempo and time signature

- **Cross Fade…** – cross fade two overlapping clips

- **Range Set** – set the edit boundaries (blue vertical lines) to the start/end of the currently selected clip

- **Loop Set** – set a loop between the start/end of the currently selected clip

- **Import…** – import audio or MIDI file(s) as new track(s)

- **Export…** – export audio or MIDI track(s)

- **Tools** – access a variety of tools to manipulate a MIDI clip

View

- **Menubar** – toggle display of the menu bar. Restore with `Ctrl+M`

- **Statusbar** – toggle display of the status bar

- **Toolbars** – toggle display of various toolbars

- **Windows** – choose which windows are displayed: File System, Files, Messages, Connections, Mixer

- **Tool Tips** – toggle display of tool tips (despite the name, these are not tool tips for the GUI icons at the top of the screen, but for the information seen when hovering over a clip)

- **Zoom** – zoom the main view: In/Out/Reset for Horizontal axis/Vertical axis/All

- **Snap** – change the Snap/beat setting

- **Refresh** – refresh the view contents

- **Instruments…** – open the instruments dialog, where you can import/export patch names etc.

- **Controllers…** – manage MIDI controllers

- **Buses…** – open the buses editor dialog

- **Tempo Maps / Markers…** – manage tempo and time signature changes

- **Options…** – open the program options dialog (see below)

Transport

- **Backward** – move playhead backward to the beginning of a range or session

- **Rewind** – move playhead backward at greater than usual speed

- **Fast Forward** – move playhead forward at greater than usual speed

- **Forward** – move playhead forward to the next edit marker or end of the session

- **Loop** – repeatedly play the range marked as a loop

- **Loop Set** – mark selected range as a loop

- **Stop** – stop the transport when rolling

- **Play** – move playhead forward at normal speed to play all un-muted tracks

- **Record** – prepare to record audio or MIDI on record-enabled tracks

- **Punch** – prepare to record the selected punch range

- **Punch Set** – mark the selected range for punch recording

- **Metronome** – turn the metronome's audio or MIDI output on or off

- **Follow Playhead** – enable view scrolling so that the playhead is always visible

- **Auto Backward** – have the playhead return to the beginning of a range or session automatically when it stops

- **Continue Past End** – have the playhead continue to move forward even after it reaches the end of the session

- **Mode** – set the JACK Transport mode: None, Slave, Master, Full

- **Panic** – stop any MIDI or audio that may be playing, either via the Tracks pane or the Files pane

Help

- **Shortcuts…** – list the keyboard key combination equivalents for various menu items, commands and MIDI Controllers. The content of this menu differs from that of the MIDI Editor's shortcuts menu

- **About…** – view information about Qtractor

- **About Qt…** – view information about Qt, the graphics toolkit used for Qtractor's user interface

5.2. MIDI Editor

File

- **Save** – save the current clip

- **Save As…** – save the current clip with naming conventions

- **Unlink** – if a MIDI clip is copied within the same track, any subsequent changes will be reflected in both the original and copied clips. This unlinks the clips, deactivating this behavior

- **Record** – set a clip to be overdubbed on the next recording operation

Properties… – open the clip properties dialog

-
- **Range Set** – set the edit boundaries (blue vertical lines) to the start/end of the currently selected clip
- **Loop Set** – set a loop between the start/end of the currently selected clip
- **Track** – set the inputs, outputs and properties of a track
- **Close** – close the current clip

Edit

- **Undo** – undo the last action
- **Redo** – redo the last action
- **Cut** – delete and copy an item to the clipboard
- **Copy** – copy an item to the clipboard
- **Paste** – paste an item from the clipboard
- **Paste Repeat…** – paste an item from the clipboard multiple times, with a user-definable period in between each
- **Delete** – delete the selected item
- **Select Mode** – select On, Off or Draw edit modes
- **Select** – select All, None, Invert or Range
- **Insert** – insert a range
- **Remove** – remove a range
- **Tools** – access a variety of tools to manipulate the clip

View

- **Menubar** – toggle display of the menu bar. Restore with `Ctrl+M`
- **Statusbar** – toggle display of the status bar
- **Toolbars** – toggle various toolbars on and off
- **Windows** – toggle display of the Events window
- **Tool Tips** – toggle display of tool tips (despite the name, these are not tool tips for the GUI icons at the top of the window, but for the information seen when hovering over a note/ MIDI controller data)
- **Note Duration** – visualise note durations in the bottom pane
- **Note Color** – set notes/values to be displayed in different colors according to pitch

Note Type – display either Note On or Key Press events

-

- **Value Color** – set notes/values to be displayed in different colors according to velocity value

- **Value Type** – determine the information to be displayed in the bottom pane

- **Drum Mode** – displays MIDI Note On events as diamonds instead of rectangles

- **Zoom** – zoom the view: In/Out/Reset for Horizontal axis/Vertical axis/All

- **Snap** – change the Snap/beat setting

- **Scale** – lock the addition/editing of notes to a particular scale, from the common to the esoteric

- **Refresh** – refresh the view contents

- **Preview Notes** – toggle whether notes are audible when inputting/editing in the matrix

- **Follow Playhead** – enable view scrolling so that the playhead is always visible

Tools

At least one note must be selected for this menu to be active.

- **Quantize** – quantize the timing or scale of notes

- **Transpose** – transpose the pitch of, or shift the timing of, a note/notes; reverse a sequence of notes

- **Normalize** – normalize the velocity value of notes using a percentage amount or absolute number

- **Randomize** – randomize note pitches, times, durations or velocity values by percentage amounts

- **Resize** – resize note durations or velocity values

- **Rescale** – rescale note times, durations or velocity values

- **Timeshift** – alter the timing of notes on a curve, such that the acceleration from the beginning note to the ending note is either increased or decreased

Transport

See Main Workspace above.

Help

- **Shortcuts…** – list the keyboard key combination equivalents of menu items and other commands. The content of this menu differs from that of the main workspace's shortcuts menu

- **About…** – view information about Qtractor

- **About Qt…** – view information about Qt, the graphics toolkit used for Qtractor’s user interface

5.3. View / Options

- **General** – configure global options, such as UI behaviour, session templates, back-ups and transport mode
- **Audio** – set capture/export format, playback options and metronome
- **MIDI** – set capture/export format, playback options, control options and metronome
- **Display** – configure the look-and-feel of the UI, including colours and fonts, plus enable/disable logging
- **Plugins** – set plugin paths, toggle plugin editor GUIs, toggle dedicated audio outputs for tracks and enable/disable various experimental features

Manual - 6 Appendixes

Manual - Table of Contents

6. Appendixes

6.1. References

Qtractor - An Audio/MIDI multi-track sequencer

http://qtractor.sourceforge.net/

http://sourceforge.net/projects/qtractor/

Qt 4 - C++ class library and tools for cross-platform development and internationalization

http://qt-project.org/

JACK Audio Connection Kit

http://jackaudio.org/

ALSA - Advanced Linux Sound Architecture

http://www.alsa-project.org/

libsndfile - C library for reading and writing files containing sampled sound

http://www.mega-nerd.com/libsndfile/

libvorbis - Ogg Vorbis audio compression

http://xiph.org/vorbis/

libmad - High-quality MPEG audio decoder

http://www.underbit.com/products/mad/

libsamplerate - The secret rabbit code, C library for audio sample rate conversion

http://www.mega-nerd.com/SRC/

LADSPA - Linux Audio Developer’s Simple Plugin API

http://www.ladspa.org/

QjackCtl - JACK Audio Connection Kit - Qt GUI Interface

http://qjackctl.sourceforge.net/

http://sourceforge.net/projects/qjackctl/

Cakewalk - Powerful and easy to use products for music creation and recording http://www.cakewalk.com/

ftp://ftp.cakewalk.com/pub/InstrumentDefinitions/

SoundTouch - Sound Processing Library

http://www.surina.net/soundtouch/

rncbc.org - The personal web presence of Rui Nuno Capela, creator and developer of Qtractor and other applications or application GUIs such as QjackCtl (a GUI for jackd), Qsynth (a GUI for fluidsynth), Qsampler, etc.

http://www.rncbc.org/

How To - Contents

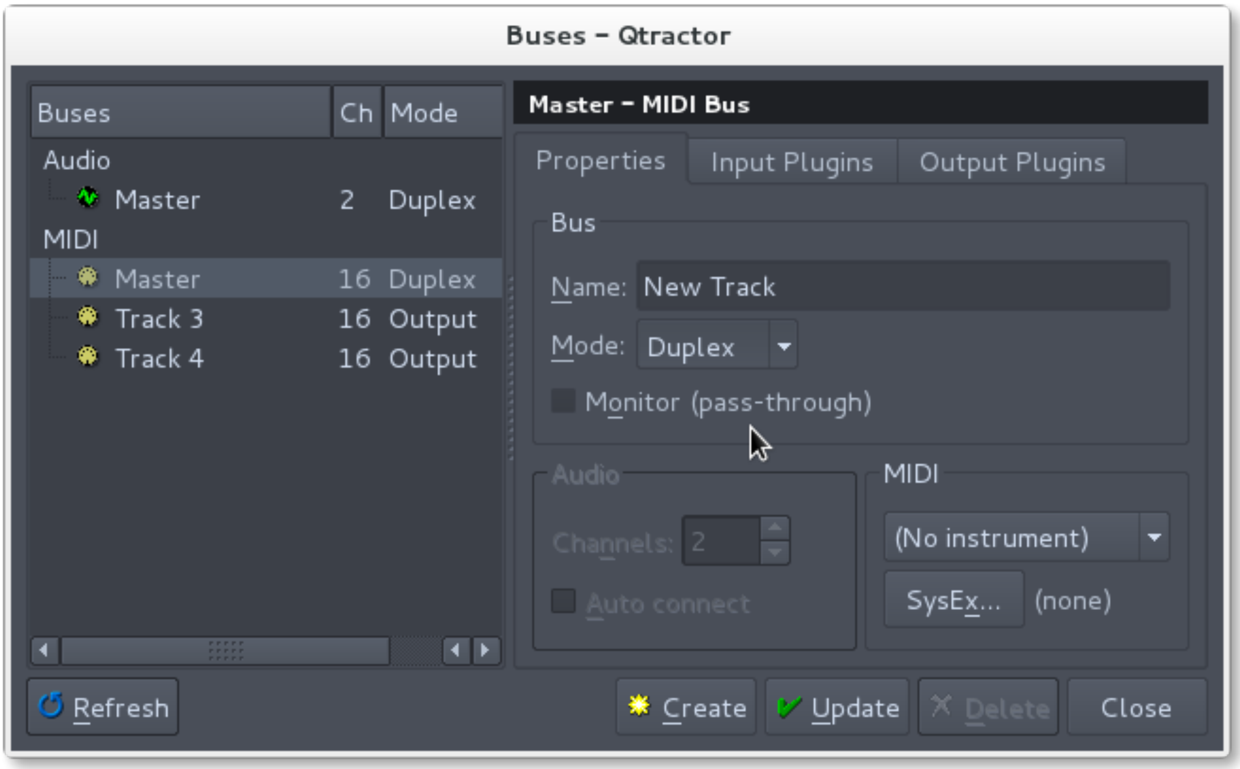
How To’s

How To - Create Individual MIDI and Audio Buses-Ports

How To - Contents

You can create individual input, output, or duplex **MIDI/audio buses** (with corresponding ports) that can be assigned, after they are created, to individual tracks in the **Track Properties menu** for each track.

To do so, from the **Main** menu go to **View** > **Buses**, which opens up the Buses menu.



Buses – Qtractor

Buses	Ch	Mode
Audio		
Master	2	Duplex
MIDI		
Master	16	Duplex
Track 3	16	Output
Track 4	16	Output

Master – MIDI Bus

Name: New Track

Mode: Duplex

□ Monitor (pass-through)

Channels: 2

□ Auto connect

(No instrument)

SysEx... (none)

Illustration 1.1: Buses Menu showing audio & MIDI buses, Properties tab, Plugins tabs, Create and Update buttons

Select one of the buses of the type of bus that you want to create (MIDI or Audio) and it's properties will be shown. Once you change the **name**, the **Update** and **Create** buttons will be available. You can update the currently selected bus or create a new one with the name you provided. You can further change other options that are visible in the menu.

You can now assign them as you like in the **Track Properties** menu for each track.

Using individual MIDI buses for each MIDI track enables you to assign the full range of the 16 MIDI channels through one bus allowing you to avoid having to select a different channel for each softsynth or external MIDI device as you would have to do, if you are just using the Master MIDI bus. This also allows you to use one MIDI bus, connected to one external MIDI port, for different tracks for each instrument on a multi-instrument synthesizer or sound module. The same would be true for a multi-voice/instrument softsynth.

For further details, see manual section 4.1.1 Routing General Concepts and Information.

How To - Sample MIDI Composition Workflow

Introduction

This How To introduces one possible approach to composing with MIDI tracks. It gives a brief outline of each stage of the process - namely track set-up, composition, recording, external routing and audio export. It's aimed mainly at beginners who aren't overly familiar with sequencers/DAWs, but some of the Qtractor-specific information may also be of interest to more experienced users. Note that a number of the tasks mentioned here are already detailed in the wiki, so this How To contains several links to it in order to avoid repeating information.

Different people have different needs in terms of workflow, so this How To is by no means a definitive guide. However, it should hopefully provide some kind of direction for people who are uncertain where to start. Finally, it assumes that you already have Qtractor up and running with JACK etc, so if this is not the case, please check the set-up information in the wiki before continuing.

- NOTE - This guide was originally written for Qtractor V:0.5.x, when it was necessary to bounce any MIDI data to audio tracks in order to export it. However, it is now (since V:0.6.7) possible to export "internal" MIDI tracks (i.e. those which use plugin instruments, such as the xsynth example detailed below) directly as audio, thus avoiding the bounce step. Nevertheless, for "external" MIDI tracks (those which send data to external instruments driven by JACK-MIDI or ALSA-MIDI, the output of these instruments then being routed back into Qtractor), the below workflow still holds true. It follows that a MIDI session which has both internal and external tracks will necessitate a "hybrid" workflow, wherein you first bounce the external tracks to Qtractor audio tracks, then export these together with the internal tracks.

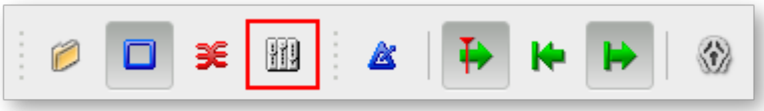
For internal tracks, the workflow described by this How To will have the following differences:

- **Section 1**: No need to create dedicated audio outputs or audio tracks
- **Sections 2, 3, 5**: Ignore
- **Section 6**: Create the Aux Sends in the MIDI tracks (note that they will still be *audio* Aux Sends, even though in MIDI tracks)

[TOC]

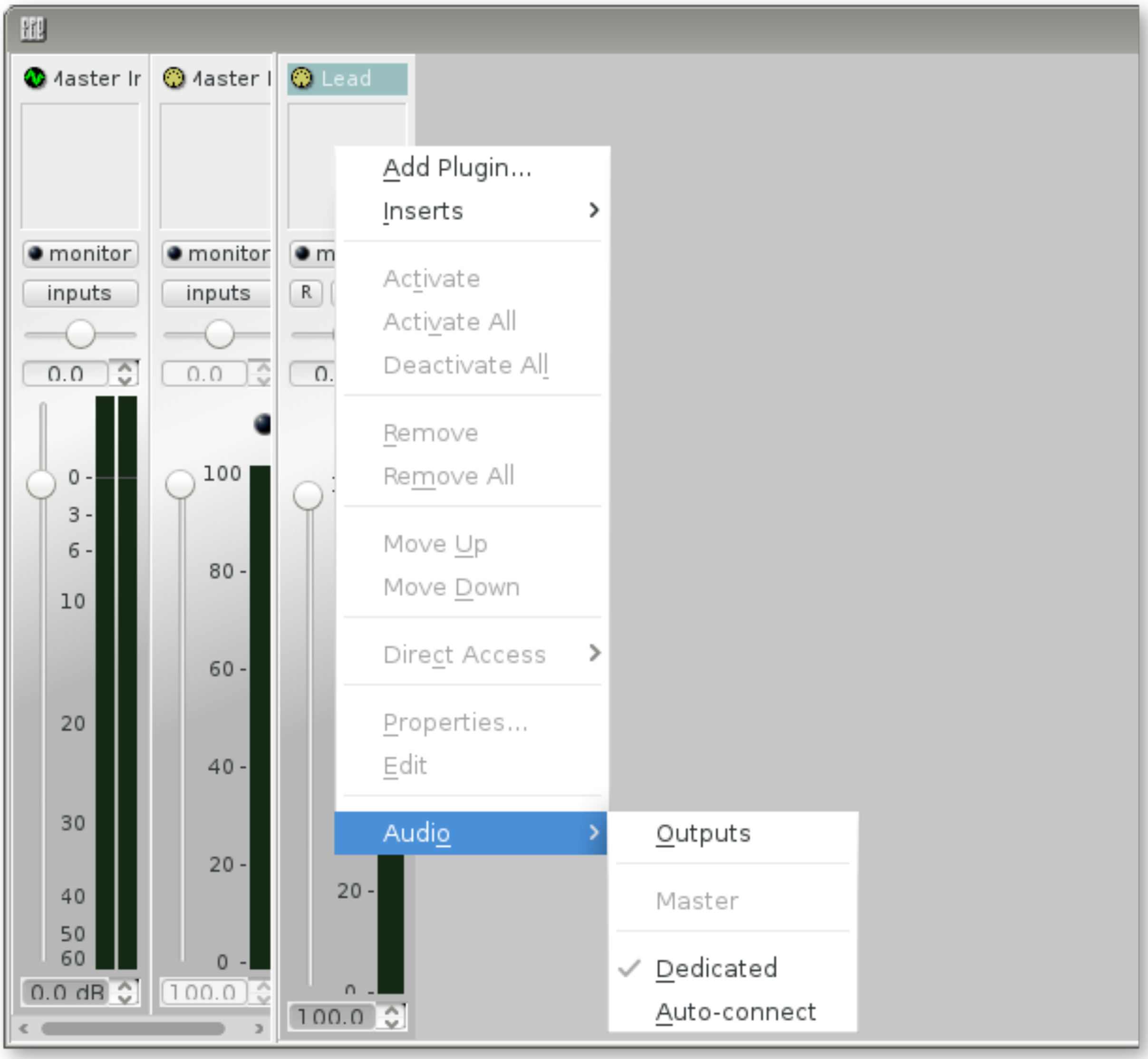
1. Creating Tracks

To keep things simple, we'll create just two MIDI tracks - Lead and Bass. Create a MIDI track, as detailed here, naming it "Lead". Then, open up the Mixer, by clicking on its icon.



The Mixer icon

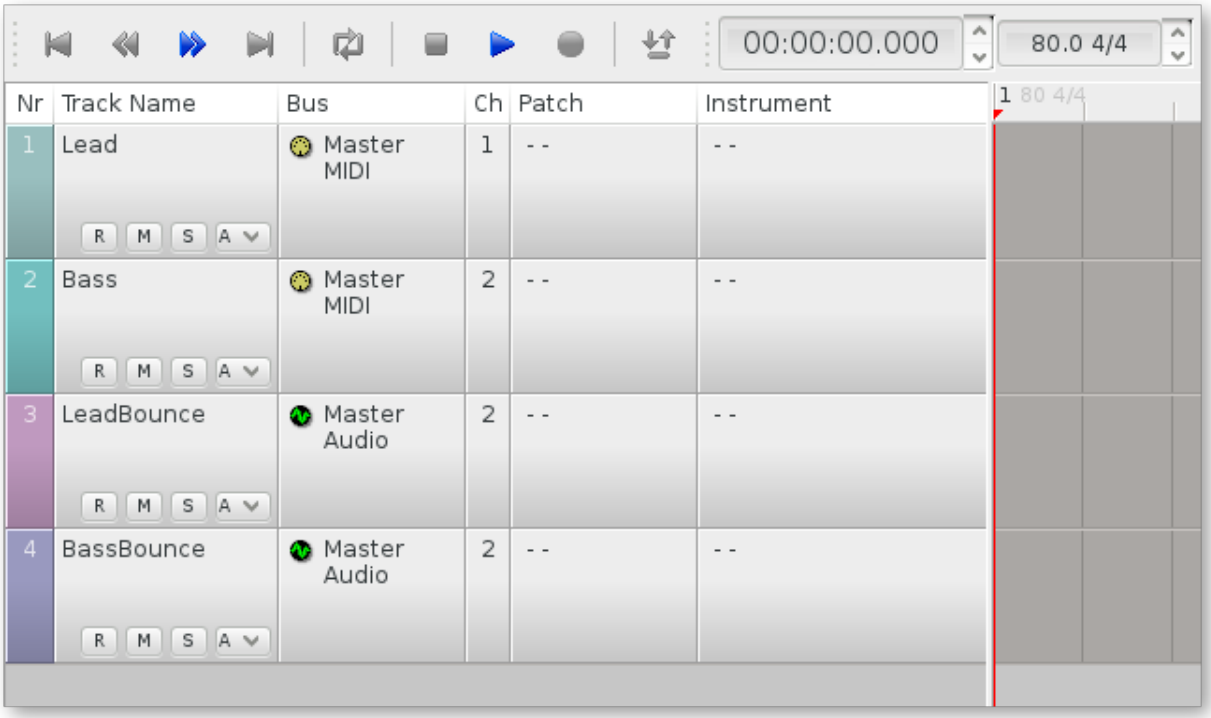
Right-click in the area directly under the “Lead” text, navigate to **Audio** and choose **Dedicated**. This will create an output specifically for this track, which we’ll later connect up. Note that you can choose to **Auto-connect** the output – doing so means that it will automatically be connected to *system:playback_1* and *2*. Whether you should do this or not will depend on your set-up, but for the purposes of this How To, we can leave it enabled.



Creating a dedicated output

Then, create another MIDI track, naming it “Bass” and repeat the above.

Next, we’ll create two Audio tracks, to which we’ll later record, or “bounce”, the output of the MIDI tracks. Using the same Track window as you used for the MIDI tracks, this time, select **Audio**, instead of **MIDI**, under **Type** and create a track named “LeadBounce” (there’s no “dedicated output” step here). Once done, create another Audio track, called “BassBounce”. Qtractor should then look like this.



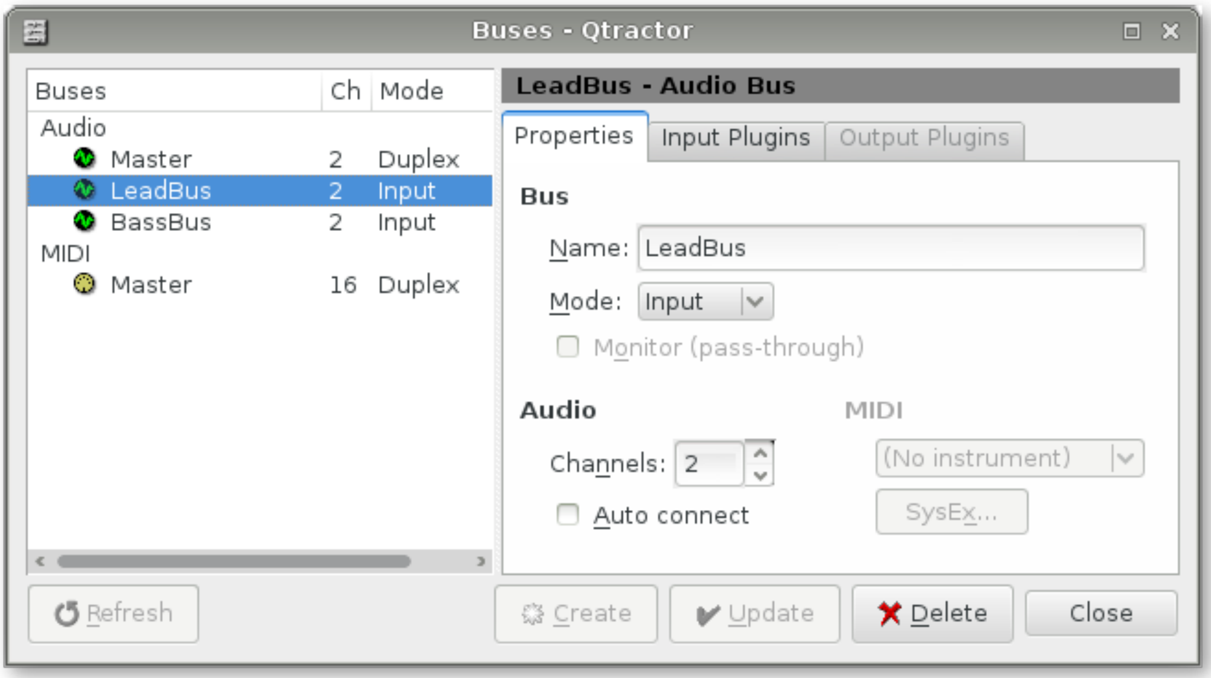
Nr	Track Name	Bus	Ch	Patch	Instrument
1	Lead	Master MIDI	1	- -	- -
2	Bass	Master MIDI	2	- -	- -
3	LeadBounce	Master Audio	2	- -	- -
4	BassBounce	Master Audio	2	- -	- -

MIDI and corresponding Audio tracks

2. Creating Buses

In order to send the output of the MIDI tracks to their Audio tracks, we use **Buses**. Buses channel the output from a particular source to a particular destination. For more detailed information on them and on the concepts of routing, please see the wiki.

The method of creating a Bus is covered in this How To. Using these instructions, create an **Audio** bus, with the **Mode** set to **Input**, calling it “LeadBus”. Then, create another in the same way, calling it “BassBus”. As with the MIDI tracks’ dedicated outputs, you can choose to Auto-connect or not (for the purposes of this How To, it doesn’t really matter). Your Bus window should then look like this.



Buses	Ch	Mode
Audio		
Master	2	Duplex
LeadBus	2	Input
BassBus	2	Input
MIDI		
Master	16	Duplex

LeadBus - Audio Bus

Bus

Name: LeadBus

Mode: Input

□ Monitor (pass-through)

Audio

Channels: 2

□ Auto connect

MIDI

(No instrument)

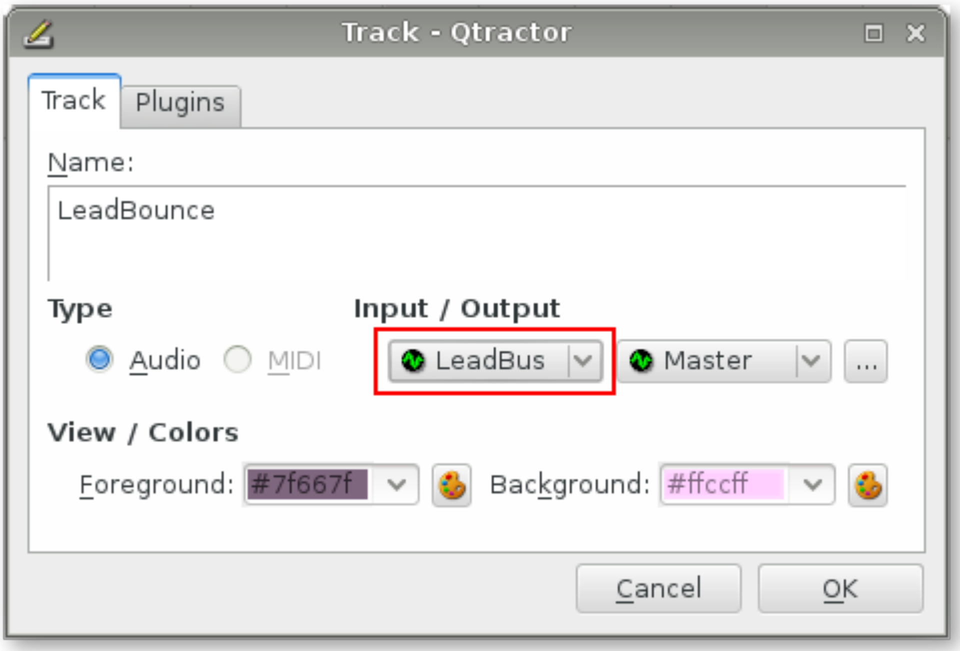
SysEx...

Creating Buses

Note that you can also create mono Buses to record into a mono audio track. To do this, you would set **Channels** to **1**, as covered in this How To.

3. Connecting everything up

Although we now have MIDI tracks to compose in, Audio tracks to record in and Buses to route the audio, we won't be able to use them until we connect everything up correctly. Double-click on your **LeadBounce** track in the main view and, in the **Track** window which appears, set its Input to **LeadBus**.



Name:

LeadBounce

Type

● Audio ○ MIDI

Input / Output

LeadBus | Master

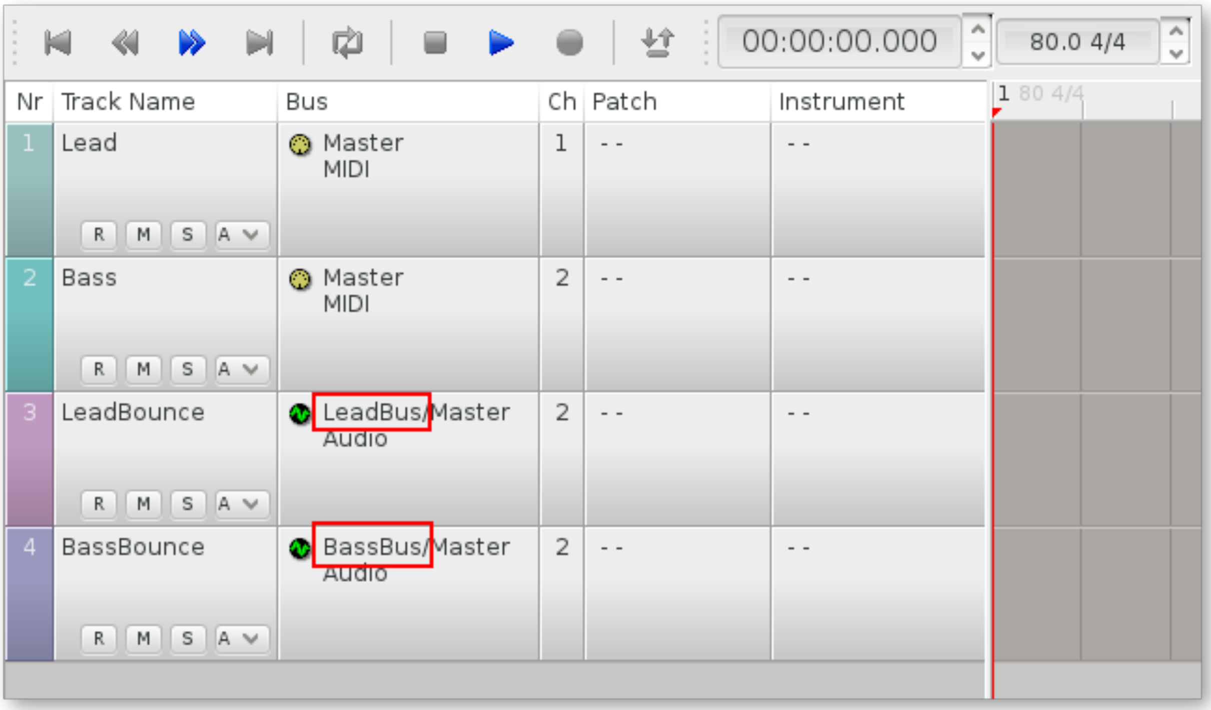
View / Colors

Foreground: #7f667f

Background: #ffccff

Creating Audio track Inputs

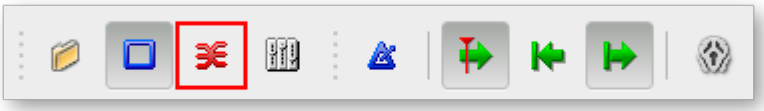
Next, do likewise for **BassBounce**, connecting its Input to **BassBus**. Qtractor should then look like this.



Nr	Track Name	Bus	Ch	Patch	Instrument
1	Lead	Master MIDI	1	- -	- -
2	Bass	Master MIDI	2	- -	- -
3	LeadBounce	LeadBus/Master Audio	2	- -	- -
4	BassBounce	BassBus/Master Audio	2	- -	- -

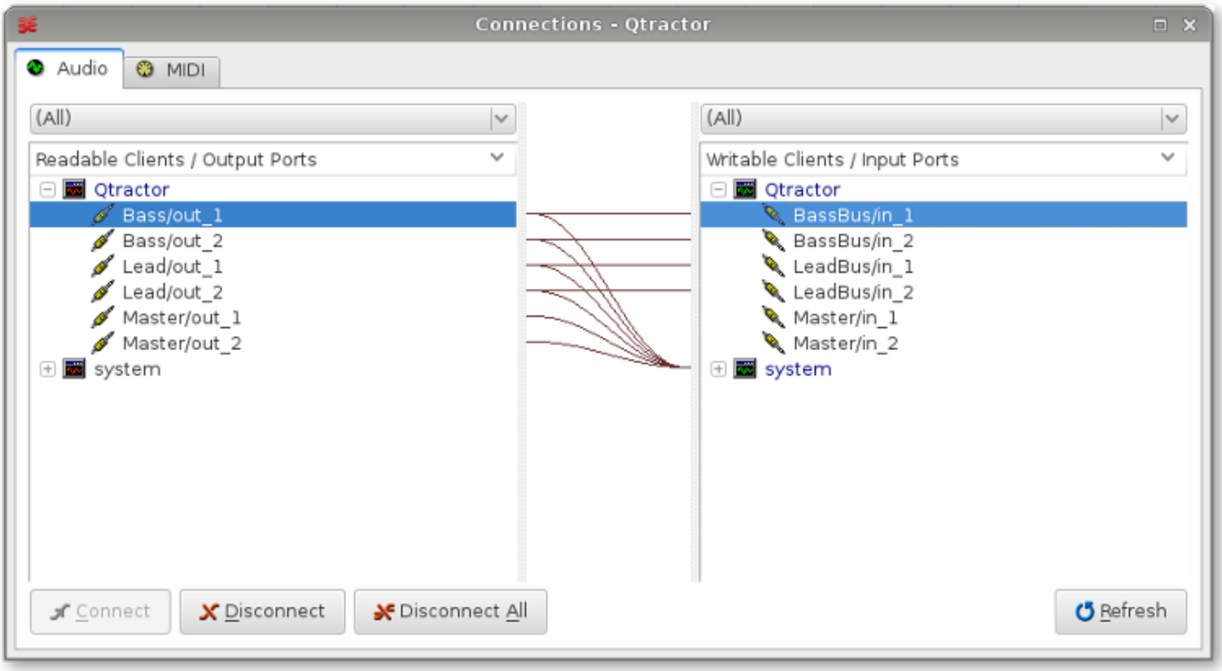
Audio tracks with their Inputs routed from Buses

The Audio tracks are now ready to receive input from their respective Buses, but we still need physically to *connect* the flow of audio. We do this via the **Connections** window. Click on its icon to open it.



The Connections icon

If you've used QjackCtl (also by the author of Qtractor) you'll be familiar with the layout. On The **Audio** tab, connect the Lead and Bass outputs on the left (which are our MIDI tracks' dedicated outputs) to their corresponding inputs on the right (which are our Buses). You can either drag the connections across, or highlight the source and destination and click **Connect**. The Connections window should then look something like this.

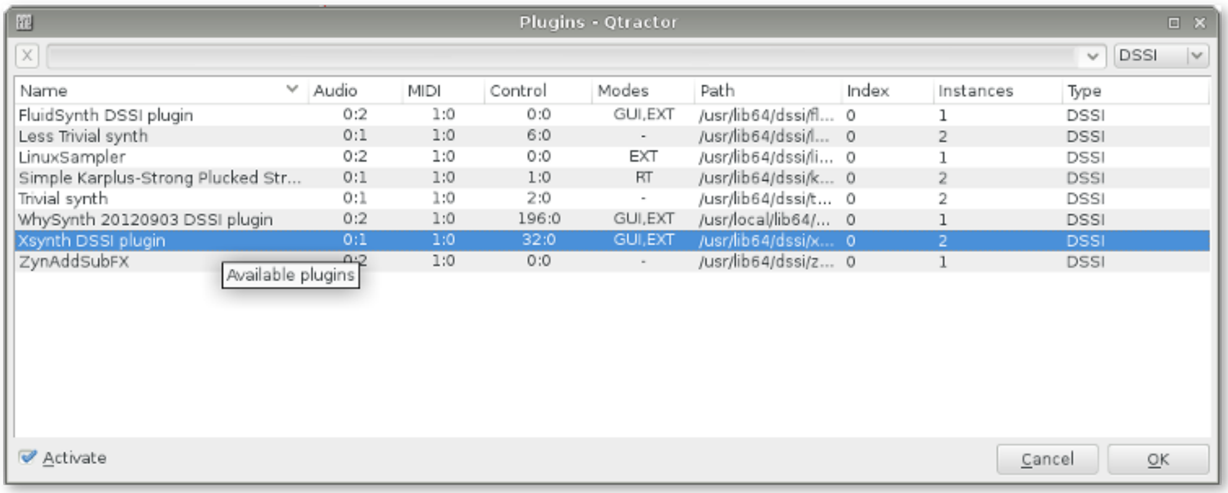


Connections window with Dedicated Outputs connected to Buses

Now, the output of the MIDI tracks will be routed to their respective Buses, which will, in turn, be routed to their respective Audio tracks to be recorded.

4. Compose!

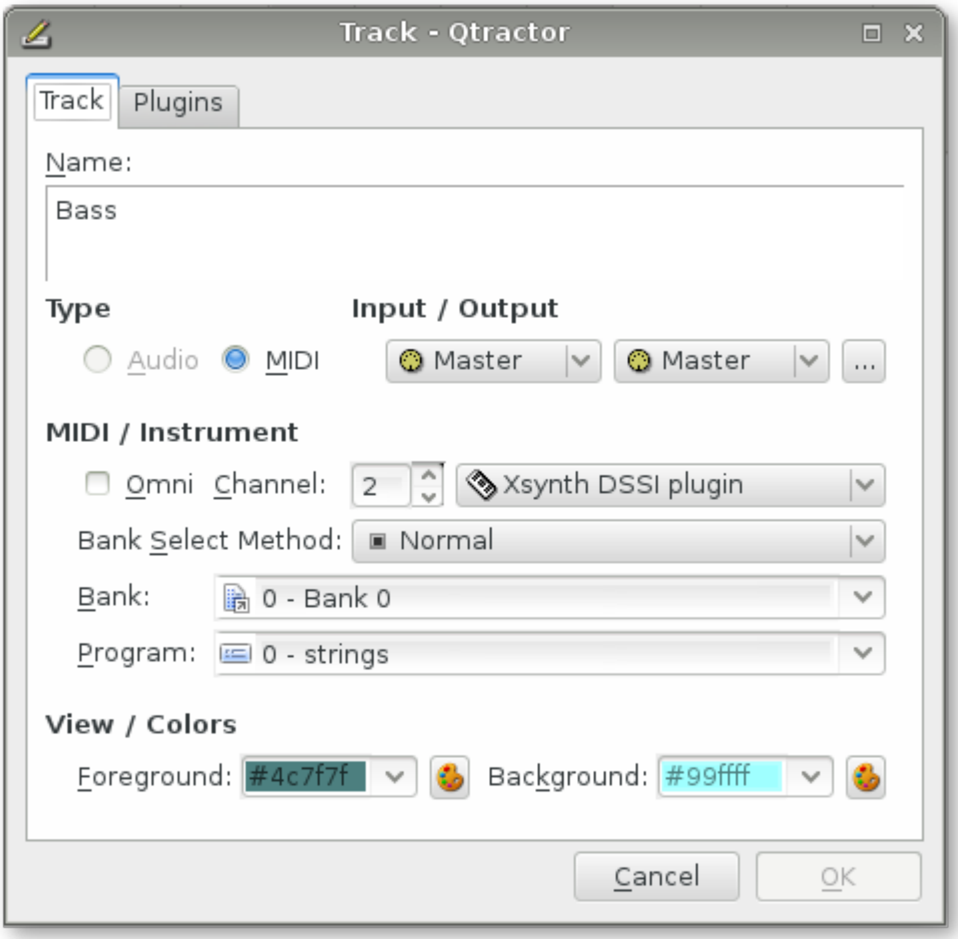
Exactly how you go about composing will depend on your needs, but for the sake of this How To, we'll do things simply and use the soft synth, xsynth (though any other will do). Back in the Mixer window, right-click under the "Lead" text, as you did previously, but this time, choose **Add Plugin**. This will bring up the Plugins window. Here, you can select from among anything you have installed, of the various plugin types supported (LADSPA, DSSI, VST (Linux native) and LV2). For more details on these, see the wiki. xsynth is a DSSI plugin, so resides in the DSSI tab.



Name	Audio	MIDI	Control	Modes	Path	Index	Instances	Type
FluidSynth DSSI plugin	0:2	1:0	0:0	GUI,EXT	/usr/lib64/dssi/fl...	0	1	DSSI
Less Trivial synth	0:1	1:0	6:0	-	/usr/lib64/dssi/l...	0	2	DSSI
LinuxSampler	0:2	1:0	0:0	EXT	/usr/lib64/dssi/li...	0	1	DSSI
Simple Karplus-Strong Plucked Str...	0:1	1:0	1:0	RT	/usr/lib64/dssi/k...	0	2	DSSI
Trivial synth	0:1	1:0	2:0	-	/usr/lib64/dssi/t...	0	2	DSSI
WhySynth 20120903 DSSI plugin	0:2	1:0	196:0	GUI,EXT	/usr/local/lib64/...	0	1	DSSI
Xsynth DSSI plugin	0:1	1:0	32:0	GUI,EXT	/usr/lib64/dssi/x...	0	2	DSSI
ZynAddSubFX	0:2	1:0	0:0	-	/usr/lib64/dssi/z...	0	1	DSSI

The Plugins window

Choose the synth patch you want via xsynth's GUI and make sure that the plugin is enabled in Qtractor (green light next to its name in the Mixer). Then, double-click your MIDI track in the left pane of the main view and, in the **Track** window which appears, select "xsynth DSSI plugin" in the **MIDI / Instrument** area. The **Bank** and **Program** fields should then reflect whatever you chose in the synth's GUI and, once you click **OK**, this information should also appear in the main view. For DSSI plugins, you should be able to select patches via either the **Track** window or the plugin's GUI, with the other automatically updating, but this may not be the case with LV2 plugins. Please see the note in the wiki about this.



Name:
Bass

Type
○ Audio ● MIDI

Input / Output
Master | Master | ...

MIDI / Instrument
□ Omni Channel: 2 Xsynth DSSI plugin
Bank Select Method: Normal
Bank: 0 - Bank 0
Program: 0 - strings

View / Colors
Foreground: #4c7f7f Background: #99ffff

Track window with patch loaded

Once you've set up your synth for the "Lead" track as above, repeat the process for the "Bass" track.

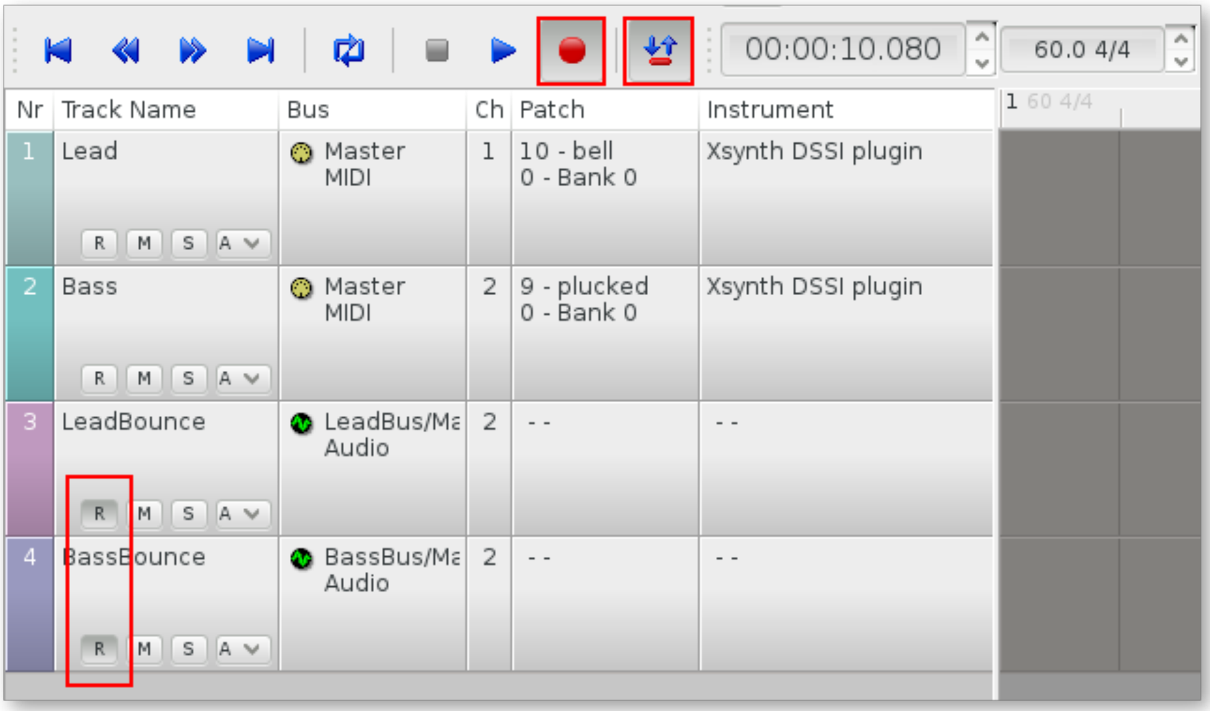
*Note that when working with soundfonts (for example, via the FluidSynth DSSI, or Calf Fluidsynth LV2, plugins), you can use the **View** -> **Instruments…** menu to import their patch names. To do so, click the **Import** button and select the desired `.sf2`/`.sf3` file; the patch names will then be imported by Qtractor.

You can also use an external synth, rather than a plugin instrument. If you choose to go down this route, you would use the **Connections** window to route the MIDI track's output bus (the default is *Master*) to your synth and, if necessary, set a specific channel in the **Track** window described above. Then, again in **Connections**, you would route the synth's audio output to your Qtractor audio track's input bus (in the context of this How To, either **LeadBus** or **BassBus**). For more details on using external synths, see the wiki.

Once your synths are set up for both tracks, you can use whatever means you like to compose. You may choose to use an external MIDI controller to play in your performance in real-time, or input notes manually via the MIDI Editor.

5. Recording

Once you're happy with your MIDI composition, it's time to record it as audio. Arm the Audio tracks for recording ("R" button on the track, via either the main view or the Mixer), move the playhead (red vertical line) to somewhere before your MIDI composition starts, enable the red "Record" icon, then click the "Play" icon to start the transport rolling (and the "Stop" icon to stop). You can also use the Punch in/out facility to specify where recording starts and ends, setting these points with the Edit-head and Edit-tail markers (blue vertical lines).



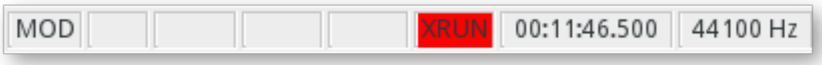
00:00:10.080

60.0 4/4

Nr	Track Name	Bus	Ch	Patch	Instrument
1	Lead	Master MIDI	1	10 - bell 0 - Bank 0	Xsynth DSSI plugin
2	Bass	Master MIDI	2	9 - plucked 0 - Bank 0	Xsynth DSSI plugin
3	LeadBounce	LeadBus/Ma Audio	2	- -	- -
4	BassBounce	BassBus/Ma Audio	2	- -	- -

Qtractor set up to record, with Record and Punch in/out icons highlighted

If you've done everything correctly, once you start the transport, the signal from the MIDI tracks should be routed via the Buses and be recorded in the corresponding Audio tracks. During the bouncing process there's always a chance that x-runs can occur, adversely affecting your audio file, so it's best to check for this. The precise positions of x-runs are unfortunately not marked visually in the track area, but when one occurs a red warning indicator will appear in the session state information at the bottom-right of the screen. X-runs are also logged, together with their time of occurrence, in the Messages pane at the bottom of the screen, so use this as a reference. Once you have your audio safely recorded, you can adjust gain levels, panning etc via the Mixer.



MOD | XRUN | 00:11:46.500 | 44100 Hz

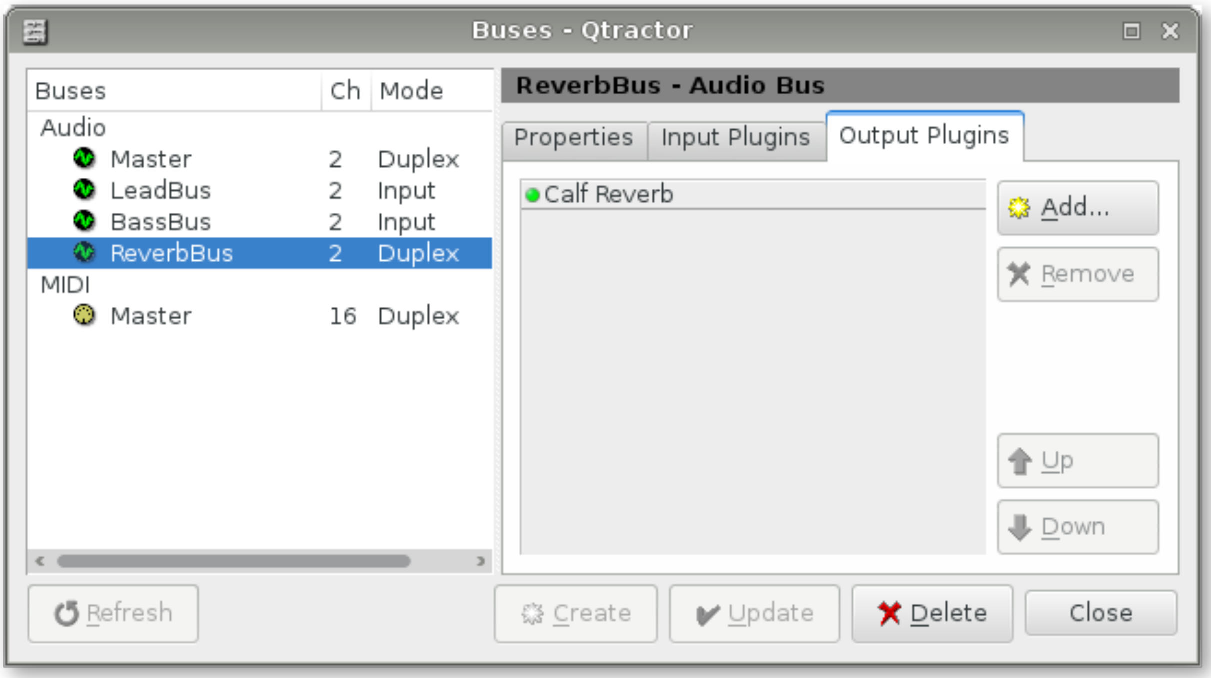
Session state information showing x-run indicator

When playing back your audio tracks, note that you may, depending on your set-up, have to mute the MIDI tracks (using the "M" button), to avoid the audio output doubling up.

6. Effects plugins and Aux Sends

In most cases, you'll want to add some effects to your tracks. This can be done very easily via the **Add Plugin** menu of the Mixer, as covered in *4. Compose!* above. You can also add several plugins to the same mixer strip if desired; they are processed in order from top to bottom. If you need to change the order of an already-inserted plugin, you can simply drag and drop it.

You may also want to send *several* tracks to the *same instance* of a plugin in order to share it. This is a reasonably common requirement, particularly for reverb, and is catered for in Qtractor with **Aux Sends**. Although we have only two tracks here, to illustrate the theory, we'll use Aux Sends to send the two tracks to another Bus, so that they can share a reverb plugin. First, create a **Duplex** Audio Bus via the Bus window we used earlier, calling it "ReverbBus". In the **Output Plugins** tab, right-click in the blank area, choose **Add Plugin**, then select the reverb plugin you want to use, making sure it's active (green light). You can also do this via the Mixer. If you're not sure which plugin to use, the Calf LV2 set is a decent place to start.



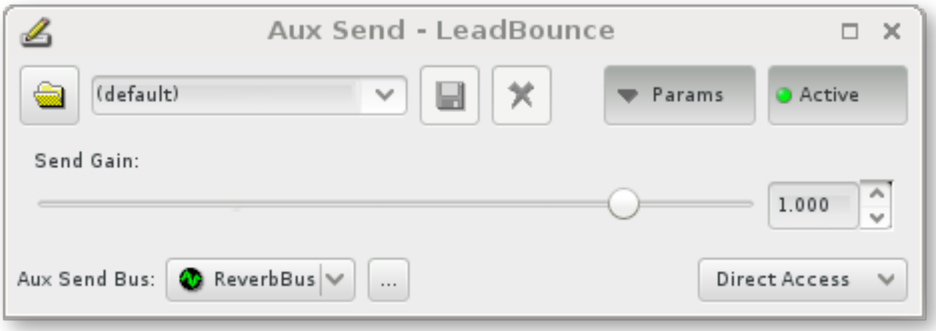
Buses	Ch	Mode
Audio		
Master	2	Duplex
LeadBus	2	Input
BassBus	2	Input
ReverbBus	2	Duplex
MIDI		
Master	16	Duplex

● Calf Reverb

Creating a reverb Bus

You'll probably also need to connect ReverbBus' outputs to *system:playback_1* and *2*, via the **Connections** window we used earlier, so that you can hear the reverb effect.

Next, in the Mixer, right-click under the Audio track name "LeadBounce" and choose **Inserts** -> **Audio** -> **Add Aux Send**. In the window which appears, set the Aux Send to "Active" (top right button), choose **ReverbBus** in the **Aux Send Bus** combo box and choose the amount you want to send via the **Send Gain** slider. Then, do the same for "BassBounce".



The Aux Send window

What value to set **Send Gain** to depends on exactly what you want to do and how you have your reverb plugin set up, but the basic theory is to have an "overall" setting in the reverb plugin itself, then adjust the amount you send to this plugin via each track's Send Gain slider. In this way, you can control the amount of reverb applied, in order, for example, to mimic an instrument's placement on the stage. Use a combination of Mixer levels, plugin settings and Aux Send gain amounts to achieve the effect you want.

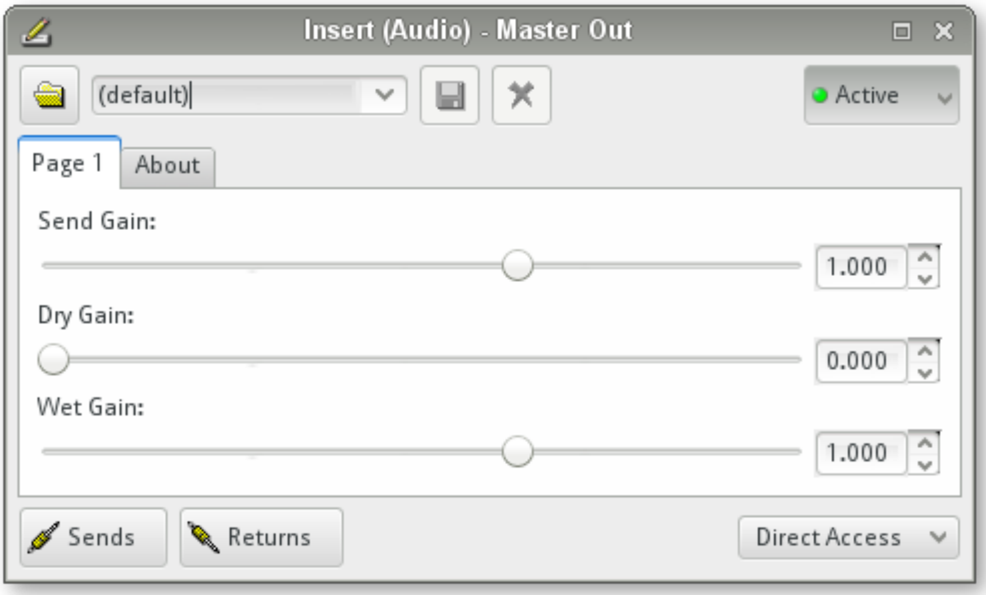
There are a couple of things to be aware of with regard to the way Qtractor routes audio when using Aux Sends:

- **Pre/post-fader placement**: All plugins in Qtractor are *pre-fader* - there is no facility to place them post-fader. This can cause issues in cases such as the above when you want to send an audio track to a reverb bus, as you would generally (though, again, it depends on what you want to do) place the send post-fader: the reason for this is that with the send in the post-fader position, the dry/wet (non-reverb/reverb) ratio is maintained even if the fader is adjusted. This limitation can be mitigated to an extent by inserting any kind of plugin which controls gain *before* the Aux Send and using that, rather than the Mixer's gain fader, to control the track's gain
- **Panning**: If you manipulate the Mixer's pan setting with an Aux Send enabled, you'll only be panning the original signal; the "sent" signal will enter its destination Bus panned to the centre. This means that you'll hear the original signal wherever you've panned it, but the Bus' copy in the centre (unless you've panned it again there). If you want to pan both the signals in tandem, add a plugin which controls panning before the Aux Send and use this, rather than the Mixer's pan setting, to pan the track

7. External routing using Inserts

In Qtractor, **Aux Sends** are used for routing audio internally, as described above. **Inserts** are similar, but are used to route audio externally. Once you have your mix set up, you may want to route it to an external source, such as JAMin. The details of how to use JAMin are outside the scope of this How To, but we will cover one way of connecting it up to Qtractor. The manual features some information on how to use the program if you're unfamiliar with it.

With JAMin running, in Qtractor's Mixer, right-click below the name of the **Master Out** Bus (right hand side) and choose **Inserts** -> **Audio** -> **Add Insert**. In the window which appears, set the Insert to "Active" (top right button). Leave **Send Gain** at 1.000, **Dry Gain** at 0.000 and **Wet Gain** at 1.000, unless you have some other set-up in mind.



Send Gain: 1.000

Dry Gain: 0.000

Wet Gain: 1.000

The Insert window

Next, use the **Sends** button to connect the Insert to JAMin and the **Returns** button to connect JAMin back to the Insert. These buttons open up the same **Connections** window we used earlier, so this process should be straightforward. Once you've connected everything up, repeat the above process for **ReverbBus**, but connect *only* the **Sends** (leave **Returns** unconnected). ReverbBus will now have two entries in its Mixer strip – the reverb plugin above and the Insert below. This means that the reverb plugin will be processed first, followed by the Insert, which is the order we want.

The audio from both your Buses should now be routed into JAMin - where you can use its many features to fine tune your mix - then back into Qtractor's **Master Out**. In this way, the Master Out strip of the Mixer, at the point below the JAMin Insert, will have a "complete" stereo signal, containing the output of both buses. Beyond (that is, below) this point, you can perform any processing/analysis relevant to the mix as a whole, such as goniometer and mono checks.

Note that upon connecting JAMin up, you may find that it is also sending its output directly to `system:playback_1`* and 2, which will cause some unpleasant doubling up of audio. To fix this, disable the outputs in JAMin, via* **Ports** -> **Out** -> **out_L** & **out_R**.

There's an important point to be aware of when working with Inserts and other external connections: Qtractor will only restore such connections *if they are present when the session is saved*. For example, if you proceed as above, save, then close and restart both Qtractor and JAMin, the connection will be correctly restored. However, if you close JAMin first, make a change in Qtractor and save (i.e. with Qtractor/JAMin in a "non-connected" state), upon restarting Qtractor/JAMin, the connection will be lost and you'll need to make it again.

This behaviour can be avoided by always ensuring that you have your external connections in order before closing Qtractor (easier said than done when using many external applications). It can also be mitigated somewhat by using QjackCtl's *Patchbay* feature to make the connections persistent; however, this too is rather unwieldy as it means adding each item manually to the Patchbay. A third approach would be to use a session manager, such as QJackCtl's *Session* feature (which employs "JACK session"), or NSM.

8. Exporting

We’ve now covered how to set up MIDI tracks, bounce their audio to Audio tracks, add plugins and route the mix to JAMin. The only thing that remains is to export the result into a single audio file. This can be done by right-clicking within the left pane of the main view and choosing **Export Tracks** -> **Audio…**. For details on the options available, please see the wiki. In the context of this How To, we’d highlight both **Master** and **ReverbBus** in order to export them together.

2018/02/01; Qtractor 0.8.6

How To - Set the number of channels an audio track records

How To - Contents

The number of channels that an audio track will record is determined by the channel count of its input bus. If you use the **Master In** bus (with its default setting of 2 channels) as the input bus for a track, the track will record two channels of audio data. If you're recording from a single channel source, such as a typical microphone or electric guitar pre-amp, you should create an input bus with a single channel, making sure it's connected to the external source you want to record, and then select it as the input bus in the track properties. This will result in a single-channel track which can be panned normally for mixdown to a stereo bus (such as **Master Out**), and avoids wasting disk space on empty or duplicate audio tracks.

For further details, see manual section 4.1. Routing–Connections, Ports, Tracks and Buses.