# Qtractor

**Linux Audio and MIDI
Multi-Track Workstation**

**Version 0.1.3**

*Frugal Damsel*

May 2008

User Manual Revision 0.1.3

*by*

Rui Nuno Capela and James Laco Hines

# Table of Contents

# Table of Contents (Continued)

# 1. Abstract

Qtractor is an multi-track Audio and MIDI sequencer application written in C++ around the Qt4 toolkit using Qt Designer. The initial target platform will be Linux, where the Jack Audio Connection Kit (JACK) for audio, and the Advanced Linux Sound Architecture (ALSA) for MIDI, are the main infrastructures to evolve as a fairly-featured Desktop Audio/MIDI Workstation GUI, specially dedicated to the personal home-studio.
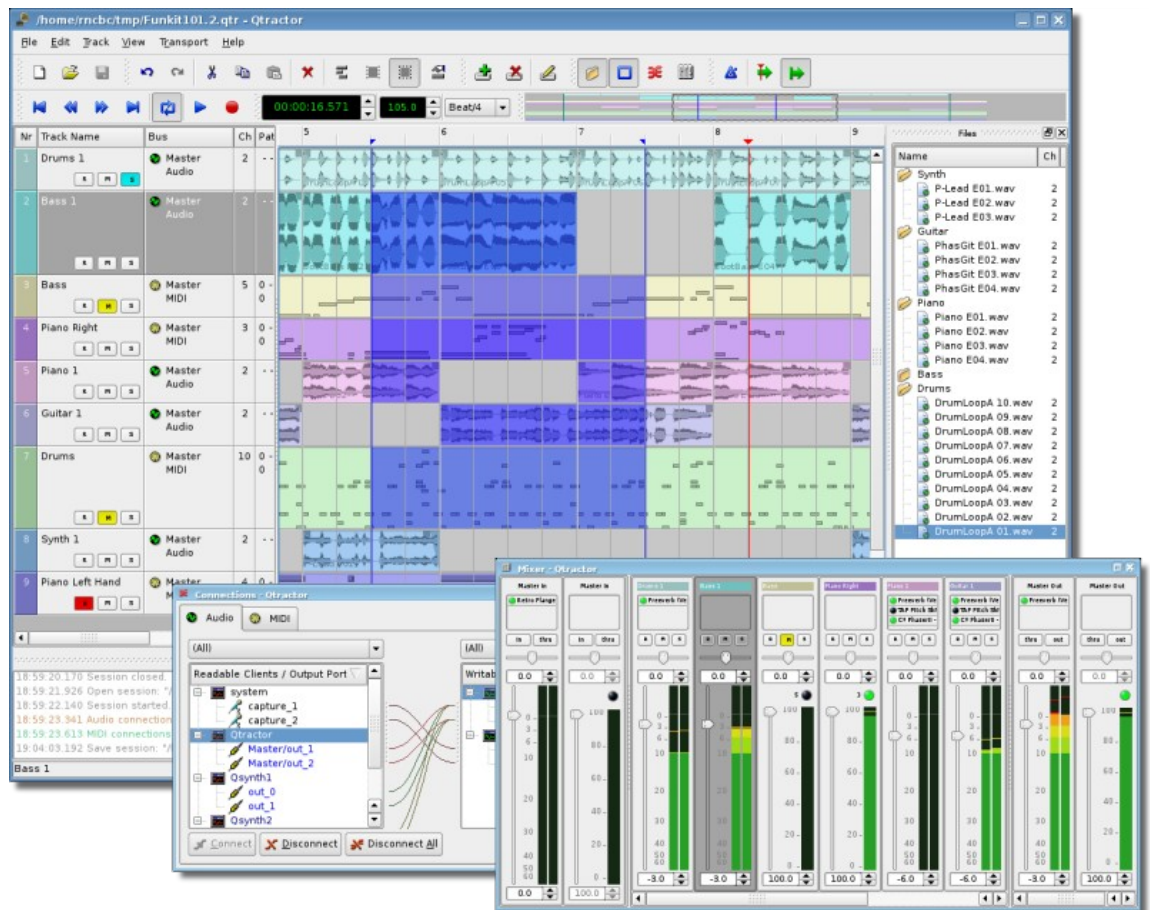
*Figure 1: Main GUI Window: Tracks, Mixer and Connections*

# 2. Introduction

As a new Linux Audio offering, Qtractor [1] targets and positions itself comfortably tagged for the *techno-boy bedroom home-studio*. However, in general, it is just yet another digital audio and MIDI multi-track composition and arranger software application. The design and functionality model takes as fundamental, the now usual multi-track composing techniques for modern music-making. It aims to be intuitive and easy to use, yet powerful enough for the serious recording enthusiast.

Qtractor is designed to be the heart of your home studio. Even though built on a monolithic application architecture (Linux kernel), this is not the whole of it. Besides its name, it should be noted, Qtractor is not a music *"tracker"* type of program. However, it has all the potential to do that job, if needed.

Qtractor is a non-destructive sequencer and arranger. It does not affect, alter or modify in any way, the audio and/or MIDI files that are displayed as Clip Objects. What *is* destructive are files resulting from capture and recording operations, and explicit changes made through specialized Clip editing (e.g., *MIDI Editor*).

Currently the Qtractor project is simply the hobby of one developer. Development was started April of 2005, initially as a Qt3 application. Since October 2006, it is officially a Qt4 [2] application.

Qtractor is natively hardwired and exclusive to the JACK [3] audio infrastructure, and the ALSA [4]  sequencer for MIDI, thus currently being a Linux-only application.

Qtractor is free open-source software, licensed under the GPL and is welcoming all collaboration and review from the Linux Audio developer and user community in particular, and the public in general.

# 3. Linux System Requirements

The software requirements for build and run-time are listed as follows:

## 3.1. Mandatory:

- **Qt 4** (core, gui, xml) - C++ class library and tools for cross-platform development and internationalization.
  http://www.trolltech.org/products/qt/

- **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/

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

## 3.2. Optional Support Libraries (opted-in at build time):

- **libvorbis** (enc, file) - 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://breakfastquay.com/rubberband/

- **librubberband** - Rubber Band Audio Time Stretcher,
  an audio time-stretching and pitch-shifting library
  http://breakfastquay.com/rubberband/

- **liblo -** Lightweight OSC implementation (needed for DSSI GUI support)
  http://liblo.sourceforge.net/

- **DSSI** - An API for soft synth plugins with custom user interfaces
  http://dssi.sourceforge.net/

- **VST-SDK** - Steinberg's Virtual Studio Technology
  http://www.steinberg.net/

# 4. Download

Qtractor is still in its alpha stages of development, but already fully functional. The latest versions are publicly available from the qtractor.sourceforge.net project web site [1].

The bleeding-edge source code may be found in the CVS repository, through anonymous (pserver) access with the following instructions:

Login to the CVS repository:

```
cvs -d:pserver:anonymous@qtractor.cvs.sourceforge.net:/cvsroot/qtractor login
```

When prompted for a password, hit enter and proceed for check-out (all in the same line):

```
cvs -z3 -d:pserver:anonymous@qtractor.cvs.sourceforge.net:/cvsroot/qtractor co
qtractor
```

Prepare the `configure` script on the just created `qtractor` source tree directory:

```
cd qtractor
make -f Makefile.cvs
```

Hopefully, the source tree will be now ready for build and installation.

# 5. Linux Compiling and Installation

## 5.1 Basic Compiling and Installation

The standard procedure for source distributions based on *autoconf* follows, through the now quite usual build command sequence:

```
./configure && make
```

and complete installation by having the appropriate (root) authority:

```
make install
```

which will end installing by copying the `qtractor` binary executable and desktop and icon files to common standard base system locations.

**NOTE:** To see all configuration options of the configure command, use ./configure –help

## 5.2 Compiling Qtractor for Native Linux VST Support (Optional)

VST support is not that easy. At least it does not work out of the box. First off, due to its licensing issues, you'll have to go through the nuisance and download yourself the VST SDK from the Steinberg Media Technologies GmbH site, specifically digging through the 3rd Party Developers section. It doesn't matter much whether you pick the VST 2.3 or VST 2.4 version, but you need to pick one and just one only. Do not use the recently announced VST 3.0. It will not work.

Once you have downloaded the VST SDK zip-archive, for which you'll have to accept their license and supply some personal data, you'll have to unpack the pertinent header files, which are found under the respective folder. Just copy those couple of files to somewhere on your system:

- **VST SDK 2.3**: `vstsdk2.3/source/common/`
  `aeffectx.h`
  `AEffect.h`

- **VST SDK 2.4**: `vstsdk2.4/plugininterfaces/vst2.x/`
  `aeffectx.h`
  `aeffect.h`

If you choose to copy those files into some standard include directory (eg. `/usr/local/include` or `/usr/include`) all will be handled automagically by the `./configure` build step. Otherwise you'll need do supply the path yourself, as in `./configure —with-vst=/path/to/vstsdk2.x/include`.

Once properly built, you will need to grab some native VST plugins and tell where those might be found. Currently, you'll need to set the `VST_PATH` environment variable with the paths where the plug-ins will reside and can be picked up by Qtractor.

Bash Command: `export VST_PATH=`*/path/to/vst_plugins*

Some ready made Linux VST plug-ins can be found on the following web sites:
http://www.linux-vst.com/
http://www.linux-vst.com/
http://cern.linux.vst.googlepages.com/home

# 6. Configuration

Qtractor holds its run-time settings and configuration state per user, in a file located as `$HOME/.config/rncbc.org/Qtractor.conf` . Normally, there is no need to edit this file, as it is recreated and rewritten every time `qtractor` is run.

# 7. Master GUI (Main)

The Qtractor user interface is a design that is modern and standard for most audio workstations of the present. The interface is easy for average user interaction, and intuitive enough that you can easily interact and discover the software's full potential of the inner-core application.

**Figure 1** shows an overall aspect of the GUI, with an example session loaded into the workspace.

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

- Menu and Tool-Bars at the top
- Track list information on the left
- Track (audio region) data on the right, with time scale above it.



*Figure 2: The Main Menu and Toolbar*

The Track Data section is where most of the action takes place, and contains visual displays of the waveforms, or audio data This section is used for editing Clip objects and maneuvering around your Project Session.

Qtractor also has other useful windows, such as the Mixer window, and the Connections window. Both these windows can be opened using the F8 key for Connections window (Jack PatchBay), and the F9 key for the Mixer window. These items are also selectable from the View menu.

Two utility windows are additionally featured: the *Messages* window, specially suited for debugging, and the *Files* window, where audio and MIDI files are organized and selected on demand.

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.*

# 8. Sessions

## 8.1 Understanding the Session Project

A Qtractor Session Project contains all the information about all your Clip Objects, placement of Clip Objects, Mixer setup, Plugins, Tempo, Time Signature and Connections Patchbay. When creating or saving a project, all this, and any related settings are saved on your hard disk within this Session Project file.

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[3] 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 [8]). This method, while it works very well, is not the recommended method due to possible errors in the real-time sample rate conversion.Realtime sample rate conversion is also going to use quite a bit more valuable CPU resources.

Please note: ***It is NOT recommended you record audio tracks while in this sample rate conversion mode.***

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.

## 8.2 Time Signature and Tempo

Although you may select any time signature and tempo for your Session, by present design Sessions can only contain a single constant tempo. Tempo must be regarded as a global setting of a Session. Qtractor does not presently support Tempo Mapping, but may eventually do so.

A new feature, which applies to existing audio clips, will reflect all tempo changes with a corresponding time-stretching effect. Time-stretching is thus applied in real-time at the buffering level, as a custom WSOLA algorithm based and derived from the SoundTouch [12] library.

## 8.3 Session Properties

To access the Session Properties, access the File/Properties menu item. Here, you can name your Session Project, set the Time Signature, Bars/Beats, Ticks/Beat, etc.
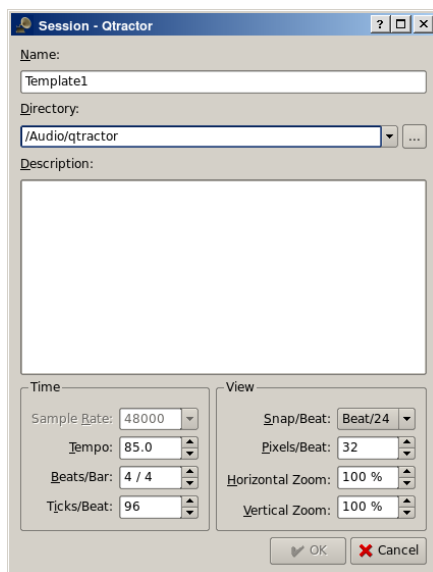


Figure 3: Session Properties Dialog

### 8.4 Session Options

Access the Options windows via the View/Options menu item. This window, and its tabs allows you to control the global parameters of Qtractor, and these settings are saved within your Session file.
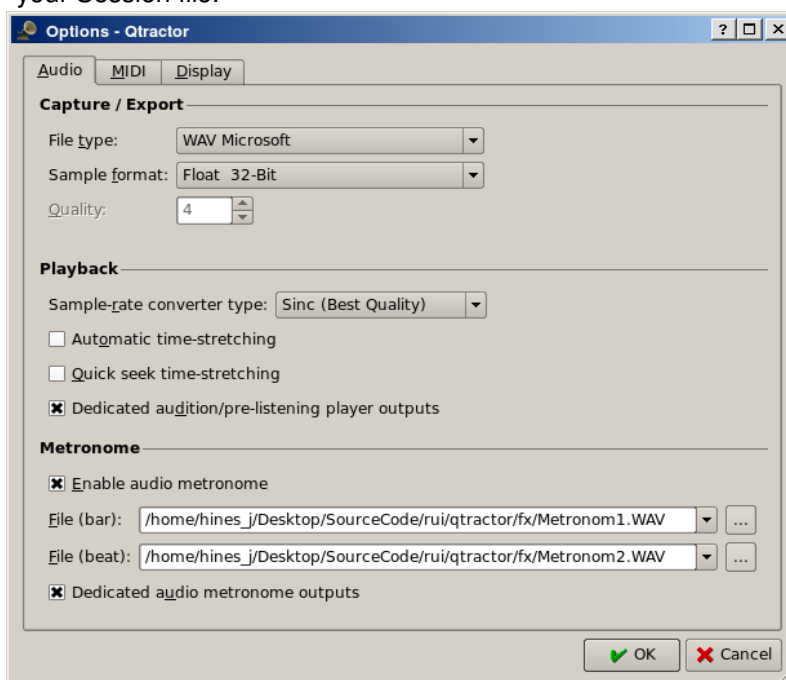


Figure 4:  Options Dialog (Found under the View menu)

# 9. Files

Sound file selection is made available through a tabbed mini-organizer. Audio and MIDI file lists are kept separate on their respective tabs. Files can be explicitly added and grouped into a hierarchical tree list. Individual and multiple files can be drag-and-dropped from the desktop environment and within the provided tree list. This lists all the files which are referred in the working arrangement session. File items can be drag-and-dropped directly into the track window, thus creating new clips in the working arrangement. This is mainly used as a your audio/MIDI file data pool.
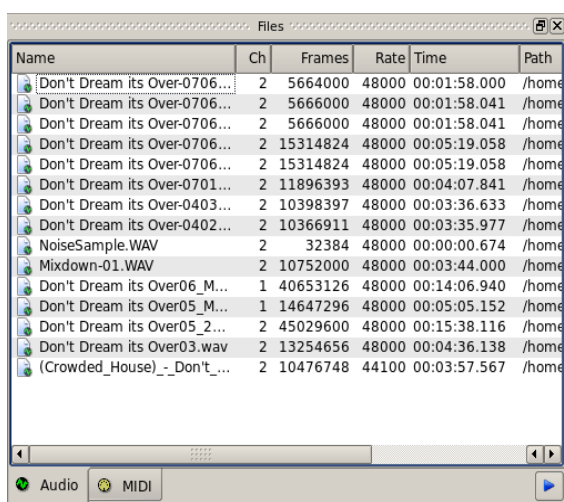


The Files Pool can also allow you to preview the files shown in the windows either by double-clicking the file name, or by click the play button on the lower right hand side of the Files Pool window.

Audio file format support is provided by libsndfile [5], and supports wav, aiff, flac, au, etc. Optional libraries provide support for both ogg and mp3 formats. libvorbis [6] (ogg) and libmad [7] (mp3). MIDI file support covers the usual SMF formats 0 and 1, through a native, home-brew implementation.

# 10. Clips

## 10.1 Clip Summary

Clip Objects are the elemental items of a session arrangement, and can contain either Audio or MIDI data. A Clip Object is merely a region of an actual sample or MIDI file. A Clip Object is also non destructive, and can be copied, truncated and time stretched as if it were actual audio/midi data.
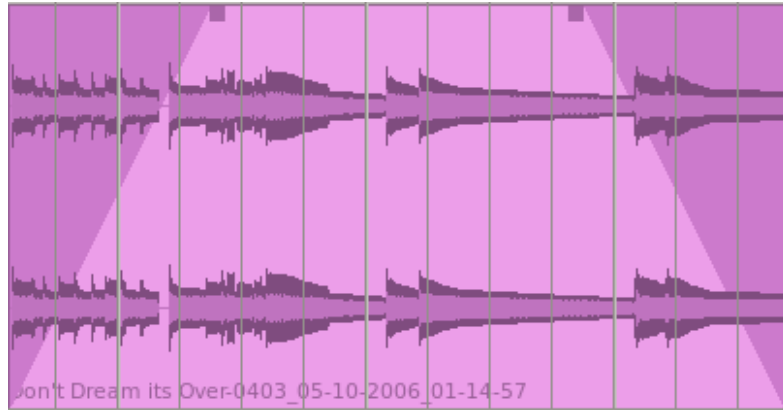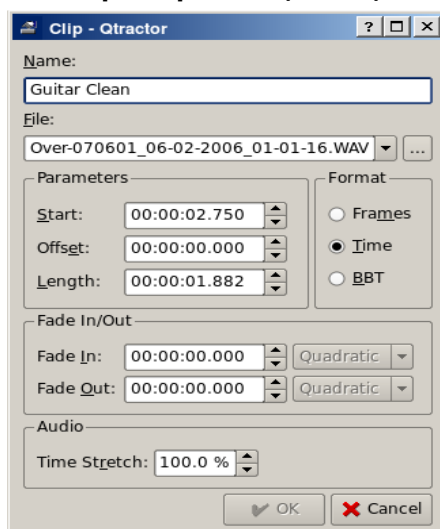


*Figure 6: An example of an audio Clip with Fade-In and Fade-Out applied.*

## 10.2 Clip Properties (Audio)



Clip properties include its label (Name), File path, start time (location), offset and length (in frames), fade-in and fade-out length (in frames) and Time Stretch percentage, respectively, from the start and end of the clip. Although fade-in and fade-outs are always displayed as straight lines, the actual audio volume (gain) and MIDI velocity effect can be opted to be of either linear, square or cubic characteristic, in as for an approximation to the logarithmic model of human ear perception.

*Figure 7: Clip Properties*

Clips are placed on tracks, either by importing audio and MIDI files as new tracks, or by dragging and dropping files into the track-view arranger window. Empty Clips may also be created by right clicking on a track, and choosing Clip/New. After being placed on their respective tracks, you may perform clip-region operations such as drag, copy, cut, paste, delete, truncate, fade in/out etc. Altering clip fade-in and fade-out is accomplished by dragging the handles (square boxes) on the top ends of any clip. A Clip may be split by positioning the playback head where you want to split it, right click the Clip, and choose Clip/Split. If you use this often, it is far more convenient to assign a keyboard shortcut to accomplish this task.

Most clip editing operations are accomplished through the usual mouse interaction, by first selecting one or multiple clips and/or regions, and applying the edit action upon the resulting selection. There are three selection modes available: clip, range and rectangular modes.
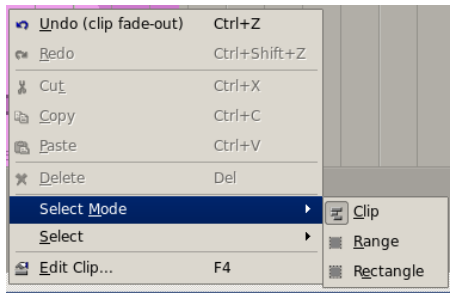
Figure 8: Demonstrates a right-click over a clip allowing for Mouse Mode Selection.

- In Clip mode, Clip Objects are selected as a whole with no sub-clip regions possible.
- In Range mode, Clip Object regions are selected on all tracks between a given time interval or range.
- In Rectangular mode, only the regions that fall under a rectangular area are selected, this means for adjacent tracks and clips only.

### 10.3 Clips and Tracks

Tracks may be armed for recording, making way for creating new audio and MIDI clip files with captured material. Tracks can also be muted and soloed on mix-down, which also applies when exporting. Most editing operations should be possible while playback is rolling (but not completely safe though; there are many procedural helpers, but not completely assisted with lock-free primitives, yet).
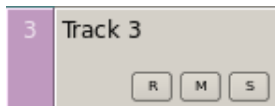


*Figure 9:  Track Functions: Record, Mute and Solo*

MIDI Clip Objects are representations of a sequence of events of one single MIDI channel, as extracted from a SMF format 0 file or of one single track, as from a SMF format 1, either in whole or in part.

## 11. Engines  and Buses

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, for all purposes, to central JACK real-time audio processing cycle, through a lock-free ring-buffer. These audio file ring-buffers are recycled (filled/emptied) at one second threshold, and has 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 a 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. JACK transport support is not an option, as  playback positioning is constantly kept in soft-chase fashion. Audio frame relocation is accounted from successive JACK client process cycles (i.e. buffer-period

resolution).

On 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 responsible inlets on capture and recording. Output buses are the main signal outlets and are responsible as playback and, more importantly, as mix-down devices.

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 output bus determines the number of channels the track supports. Clips bounded to disparate multichannel audio files, for which their number of channels do not match with proper bus/track's one, are automatically resolved on mix-down. Figure 2 shows one typical signal flow block diagram.

By default, "Master" buses are automatically created at session startup, being stereo for
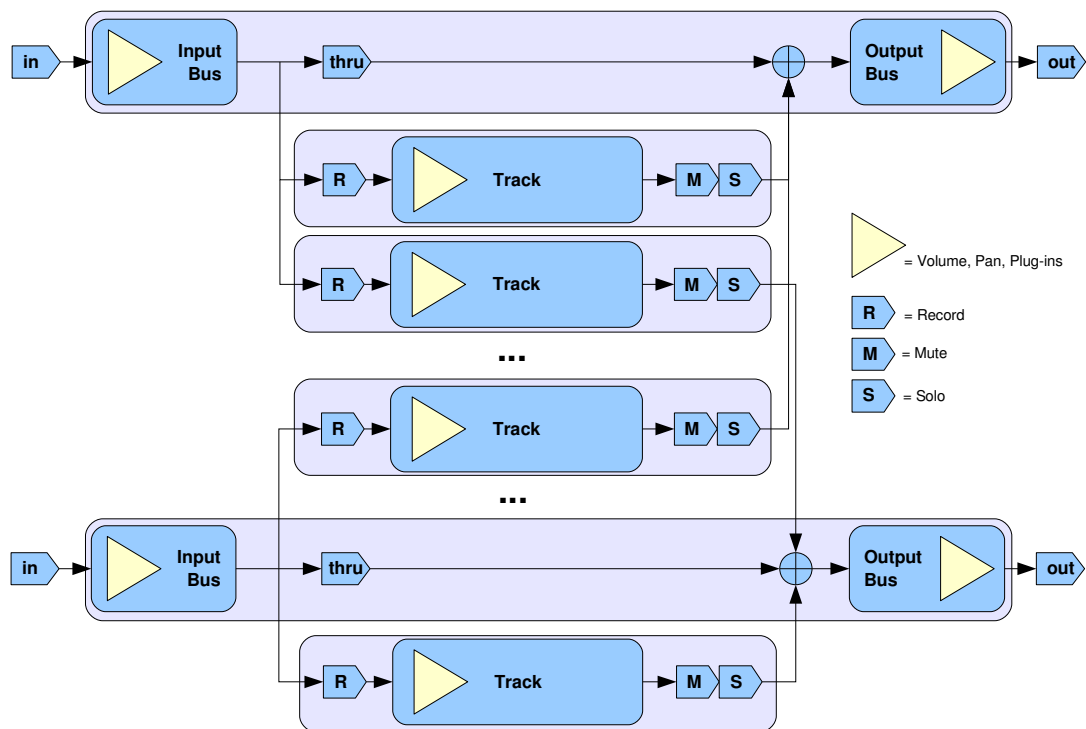


*Figure 10: Buses / Tracks signal flow diagram*

audio (2 channels ports, auto-connected) and single port for MIDI (16 logically addressable channels). Bus ports are accessible for arbitrary connection to and from external client applications or devices, through the *connections* window interface.

# 12. Track View
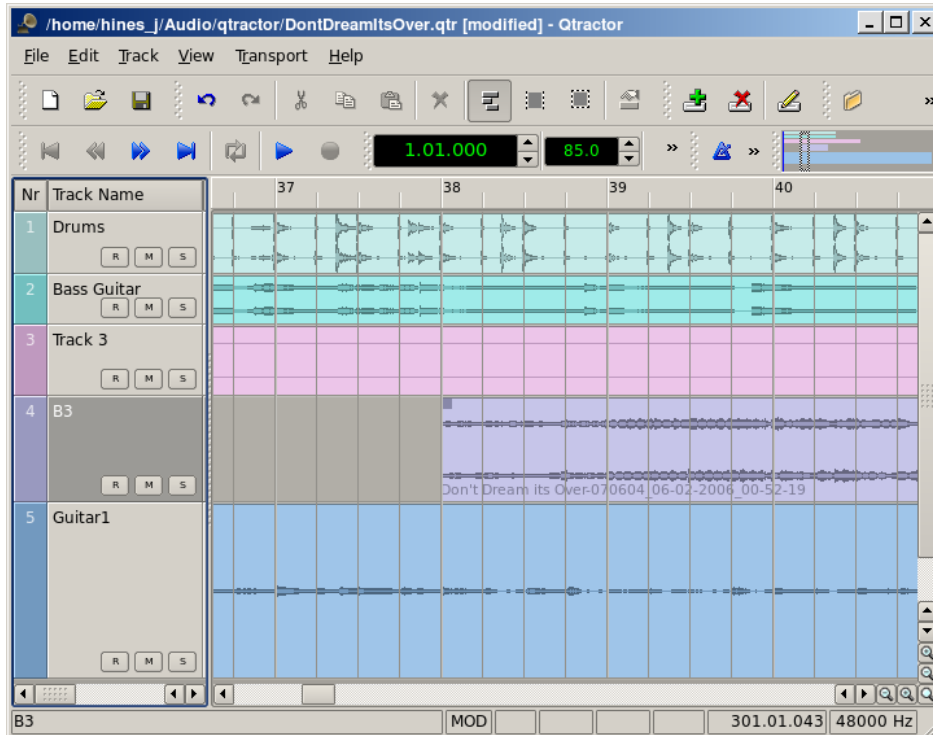


*Figure 11: This is a snapshot of the main window with track layout.*

Tracks are arranged as a sequence of one or more overlapping 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 made on this track list-view window.

The track list-view window has two panes, the left one displays the list of tracks with their respective properties and the center-right pane is the proper track-view canvas window where main multi-track composition and arranging activity is pictured and performed. As 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. When not set to "None", the snapping is always 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 and track vertical resizing operations are also available.

# 13. Mixer



Figure 12: Full Mixer Window

The mixer window serves for session control, monitoring, recording and assistance in mix-down operations. The mixer is divided in three panes: the left accommodates all input buses, the center with individual track strips and the right for the output buses. Each mixer strip offers a volume and pan control and monitors each one of the respective buses and tracks. Audio strips also offers the possibility to chain plug-in effects (LADSPA [9]).

Monitoring is presented in the form of peak level meters for audio and note event velocity for MIDI, both with fall-off *eye-candy*. MIDI mixer strips also feature an output event activity LED.

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 respective channel controller-7 and system-exclusive master volume for output buses. MIDI pan control is only available for track strips and is implemented through channel controller-10. MIDI input buses have volume and pan controls disabled.

# 14. Connections Patch-bay
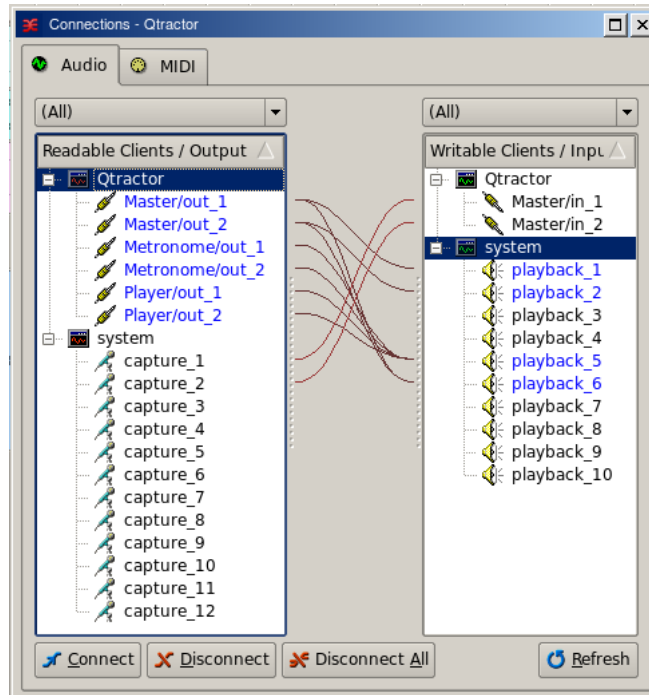


*Figure 13: Jackd Connections Window*

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. Incidentally the *Connections* window can also be used to make connections between external client application ports, either JACK clients for audio, or ALSA sequencer clients for MIDI. In fact, it almost completely replicates the very same functionality of QjackCtl [10]. All connections on the existing input and output buses are properly saved and restored upon session recall.

# 15. Audio Effects Plugins
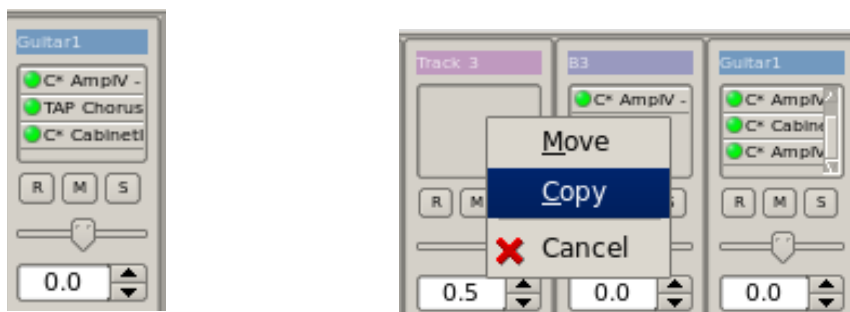


## 15.1 Summary

There are three types of Plugins supported within Qtractor. LADSPA, DSSI and VST. Plugin support is available for all audio input and output buses and for all audio tracks. All Plugin types are aggregated seamlessly as one single instance on a multi-channel context and can be individually selected, activated and moved within the plug-in chain order.

Also, you may drag-and-drop all plugin instances over the mixer strip channels. You can move, drag and drop inside the same strip or over to, and from any other. You may also copy plugins from one channel strip to another as well. A mini menu will ask if you want to copy or move the plugin. Individual plugin control parameters can be modified in real-time through

provided dialog windows and maintained as named presets for re-usability.

## 15.2 LADSPA

LADSPA [9] has been the Linux audio plugin standard for many years. There are literally hundreds of LADSPA plugins available for Linux. LADSPA plugins give the user many standard options such as Eqs, Filtering, Reverb, Chorus, Amp and speaker simulation, etc.

## 15.3 DSSI

DSSI plugin support is available for DSSI effects plugins. DSSI Instrument plugins are not yet supported. You must have the core DSSI subsystem installed in order for this type of plugin to function. When DSSI is present, the DSSI-VST wrapper may also be used. This wrapper uses WINE (http://www.winehq.org) to allow a user to run native Windows® VST applications. The DSSI paths may be set within the Options dialog under the Display tab, or with an environment variable. (see section 15.5).
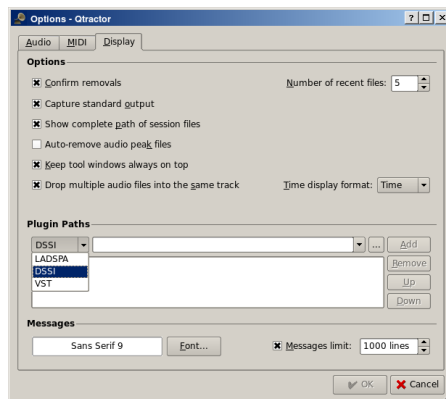


## 15.4 VST (Linux Native)

Native Linux VST plugin support is also available. Presently, there are only a few native Linux VSTs available, but more should be on the way soon, thanks to some aggressive ongoing projects.

Please see section 5.2 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 variabe.**

# 18. MIDI Instruments
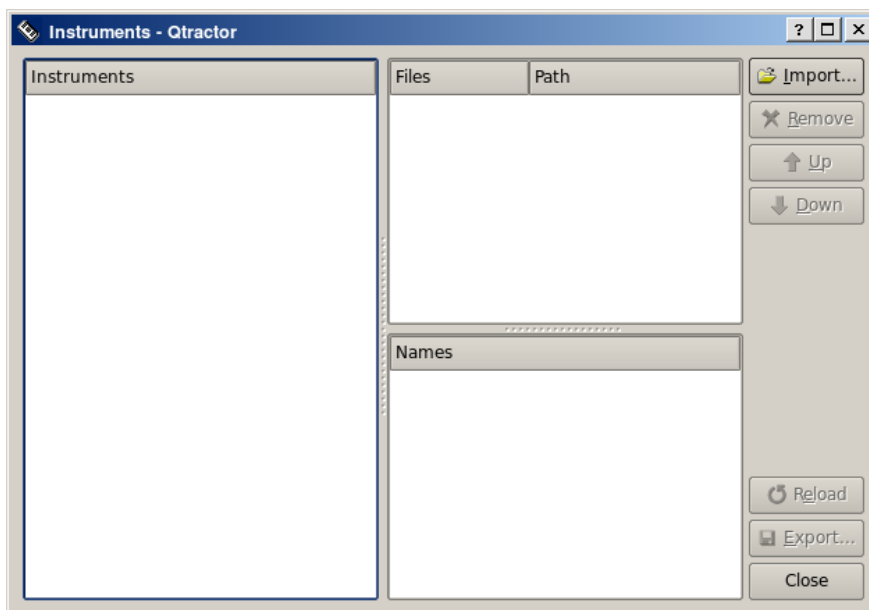


*Figure 17: MIDI Instruments Editor*

As a special feature, Cakewalk [11] instrument definition files (.ins) are natively supported, thus offering a convenient MIDI bank-select/program-change mapping for existing MIDI instrument patch names, and easier, intelligible selection of MIDI track channels.

# 19. MIDI Editor
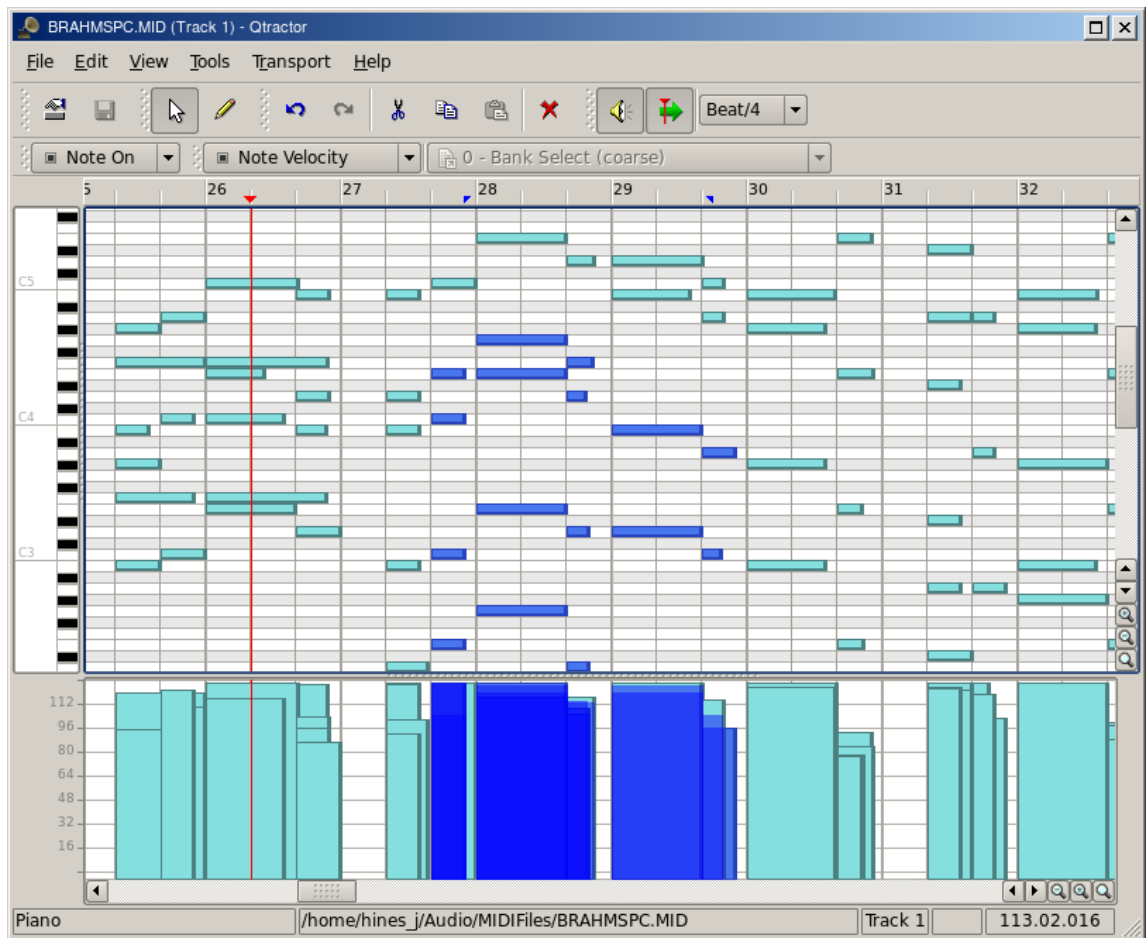


Each MIDI clip content may be readily edited under a dedicated and fairly complete piano roll *MIDI Editor*t, with individual pitch, velocity and controller, editable trough the usual GUI operations such as: Multi-extended selection, drag-and-drop, move, cut, copy, paste, deletion of every event in the MIDI sequence is rightly accessible on the fly (Figure X).Figure 18: The MIDI Editor Window

Special home-brew tools for batch processing are also implemented and applicable to the any event selection: quantize, transpose, normalize, randomize and resize. All MIDI editing operations are available and processed in real-time, effective while playback. Several *MIDI Editor* instances may be active and open in any time, provided each one refers to its own clip.

All MIDI content may be saved as standard MIDI files (SMF Format 0 or 1).

# 20. Audio / MIDI Export
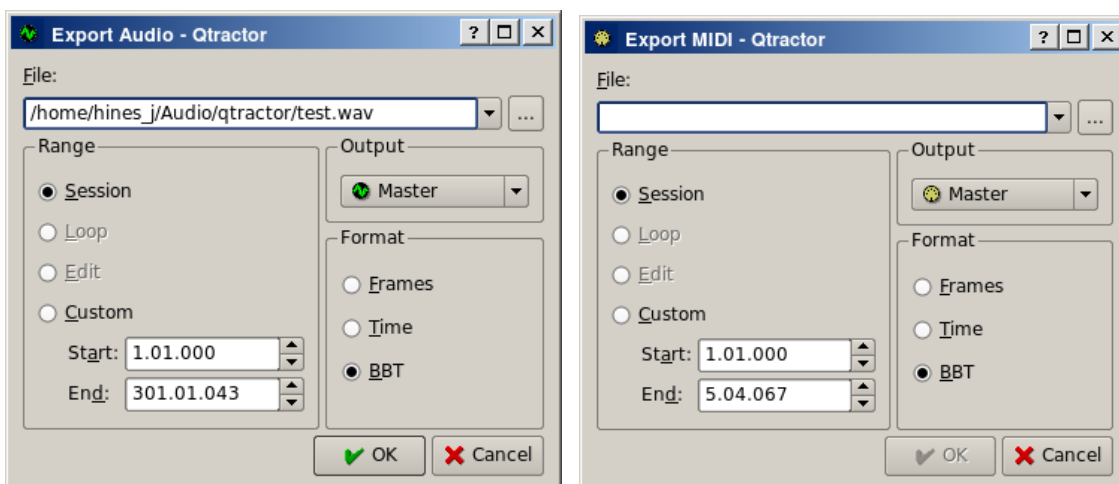


*Figures 18 & 19: The Export Audio and Export MIDI Windows*

All or part of the session may be exported to one audio or MIDI file. *Audio export* is implemented through the special JACK *freewheel* mode, thus faster than real-time, resulting in the complete and exact mix-down of selected audio material into a designated audio file of the opted format (wav, flac, au, aiff or ogg). *MIDI export* is just the same but for MIDI material only, resulting in the merging and concatenation of selected MIDI tracks and clips into a single MIDI file (SMF  Format 0 or 1).
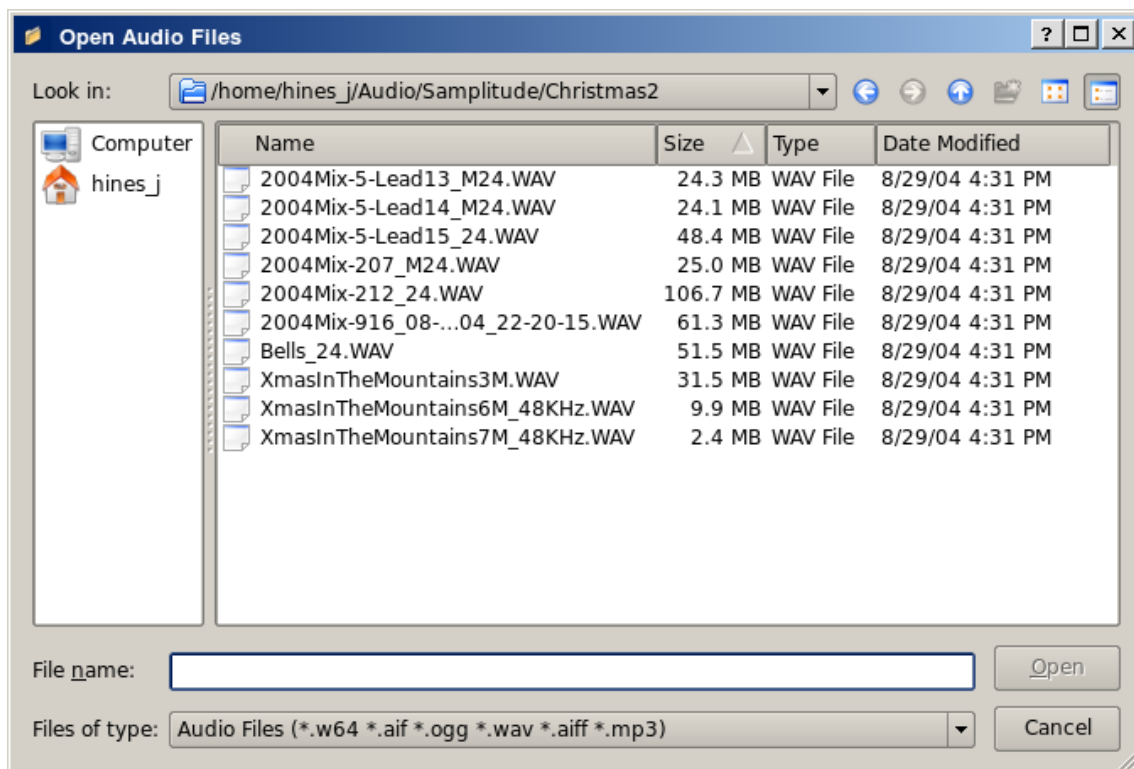


*Figure 20: The Import Audio Window*

# 21. Keyboard Shortcuts Editor

Keyboard shortcuts are useful for the power user, in such that it provides for a quick mechanism for performing often used commands quickly, without the use of your mouse.

Keyboard shortcuts may be customized to your preference by using the Shortcuts Editor.

This editor may be found in the Help menu.
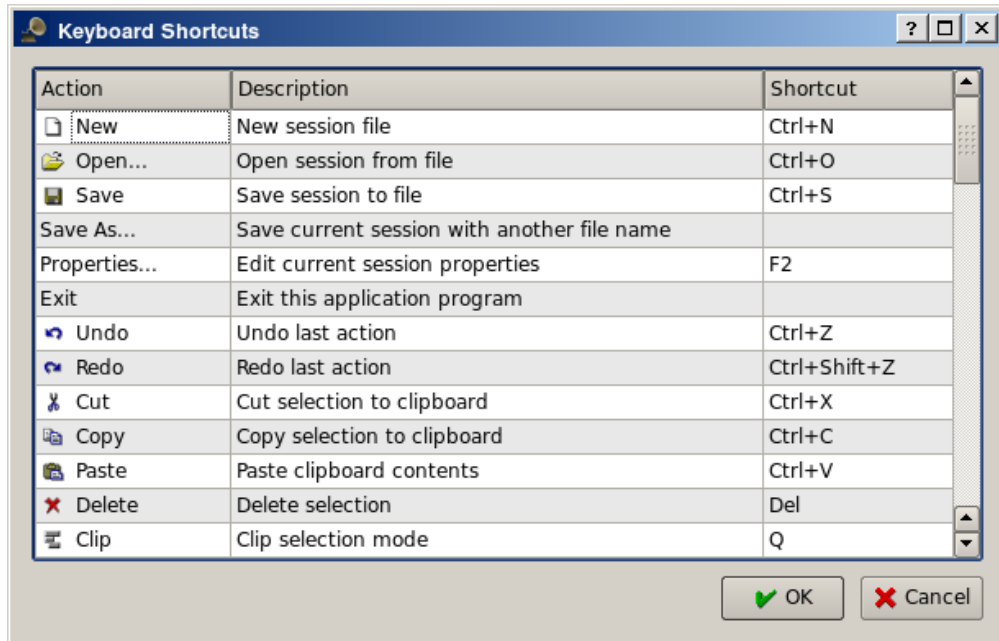


*Figure 21: The Keyboard Shortcuts Editor*

It is very straight forward in its use. Simply find the item you want to create a shortcut for, left click in the shortcut cell, and type on 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.

# 21. Menu Items

### 21.1 File Menu

**New:** Creates a new Session Project.
**Open:** Opens a previously created Session Project.
**Open Recent:** Contains a list of several of your last used Session Projects.
**Save:** Saves your current Session Project
**Save As:** Saves your current Session Project with naming conventions
**Properties:** Opens the Session Properties dialog
**Exit:** Exits the program.

### 21.2 Edit Menu

**Undo:** Will undo the last action.
**Redo:** Will Redo the last action.
**Cut:** Deletes and copies the item to the clipboard.
**Copy:** Copies the item to the clipboard
**Paste:** Pastes the item from the clipboard.
**Delete:** Deletes the selected item.
**Select Mode:** Selects the edit mode, one of Clip, Range or Rectangle.
**Select:** None, Range, Track or All
**Clip:** New, Edit and Split

### 21.3 Track Menu

**Add Track:** Will add either Audio or MIDI track.
**Remove Track:** Removes a track.
**Track Properties:** Calls the Track Properties dialog.
**Inputs:** Shows current track inputs.
**Outputs:** Shows current track outputs.
**State:** Current track state: Record, Mute, Solo, Monitor.
**Navigate:** Current track navigation: First, Previous, Next, Last.
**Move:** Move current track navigation: Top, Up, Down, Bottom.
**Import Tracks:** Import either Audio or MIDI track.
**Export Tracks:** Export either Audio or MIDI track.

### 21.4 View Menu

**Menu Bar:** Will toggle whether the menu bar is shown.
**Status Bar:** Will toggle whether the Status Bar is shown.
**Tool Bars:** Toggles various Tool Bars on and off.
**Files:** Toggles whether the Files dialog is on or off.
**Messages:** Toggles whether the Messages dialog is on or off.
**Connections:** Displays the Connections dialog.
**Mixer:** Displays the Mixer window.
**Zoom:** Zoom the main view: In, Out, Reset.
**Snap:** Change the snap-per-beat setting.
**Refresh:** Refresh the view contents.
**Instruments:** Displays the Instruments dialog.
**Buses:** Opens the Buses Editor dialog.
**Options:** Opens the Program Options dialog.

### 21.5 Transport Menu
**Backward:**
**Rewind:**
**Fast Forward:**
**Forward:**
**Loop:**
**Loop Set:**
**Play:**
**Record:**
**Metronome:**
**Follow Playhead:**
**Auto Backward:**
**Continue Past End:**

### 21.6 Help Menu
**Shortcuts:**
**About:**
**About Qt:**

# 22. Future Thoughts

As of its current status, there are many and rather fundamental functionality still missing that tear Qtractor apart from a finished product, let alone for the quest of its own goals. It's still a work in progress. In my own personal agenda priority, the following are the ones for taking care in times to come, in no special order. Check marks (✔) indicate nearly complete, or completed features.

**General**
- DSSI and Native VST Plugin Support ✔
- LV2 Plugin Support
- Snap/Zoom Menu Accessibility ✔
- Buffer/CPU% and Under-Runs Monitoring
- OSC or D-BUS Interface ✔
- Integrated Scripting (angelscript?)
- File/Session Project Management
- File Error Message Handling
- Redesigned Session Project Dialog
- Session Project Templates

**MIDI**
- JACK-MIDI Support
- MIDI SysEx Librarian
- MIDI Event List and Filter
- MIDI Editor Draw Mode
- MIDI Groove/Swing Quantize
- MIDI Controller Feedback
- MIDI Sync (MTC/SMPTE, MIDI Clock)
- Automation / Dynamic envelope curves

**Tracks**
- Auto Cross-Fading of Overlapped Clips
- Punch in/out and Loop Recording (eg. Takes)
- File Revision Resource Management

**Clips**
- Clip Split Command ✔
- Clip Locking, Muting

- Clip Plugins
- Pitch-Shifting of Individual Clips ✔
- Paste Repeat Command
- Clip Linking

**Mixer**
- Mixer Presets
- Mixer Fader Groups
- Effect Aux Send/Returns (pseudo-plug-in)

# 23. Conclusion

As fundamental as is, Qtractor might be just some clone of earlier and existing software, being blatantly one of the Cakewalk's [11] Pro Audio series. It is however more than that, when regarded from the free open-source software development point of view, much like some cauldron framework, user-oriented, programmable, pattern sequencer, eventually targeted as a potential toolbox and workbench for easy, direct, live music-making and experimentalism

# 24. Acknowledgements

I am grateful to the free software open-source community in general and to the Linux Audio developers and users in particular, who dedicated their valuable time to the development and support of free audio and MIDI software, being Qtractor just one humble manifestation of such class of human endeavor.

Qtractor is free / open-source software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2 or later.

Qtractor logo/icon is an original work of Andy Fitzsimon, borrowed from the public domain openclipart.org gallery.

All or some product names mentioned in this document may be trademarks of their respective holders.

# 25. References

[1]     **Qtractor** - An Audio/MIDI multi-track sequencer
        http://qtractor.sourceforge.net/
        http://sourceforge.net/projects/qtractor/

[2]     **Qt 4** - C++ class library and tools for cross-platform development and
        internationalization.
        http://www.trolltech.org/products/qt/

[3]     **JACK** Audio Connection Kit
        http://jackaudio.org/

[4]     **ALSA** - Advanced Linux Sound Architecture
        http://www.alsa-project.org/

[5]     **libsndfile** - C library for reading and writing files containing sampled sound
        http://www.mega-nerd.com/libsndfile/

[6]     **libvorbis** - Ogg Vorbis audio compression
        http://xiph.org/vorbis/

[7] **libmad** - High-quality MPEG audio decoder
http://www.underbit.com/products/mad/

[8] **libsamplerate** – The secret rabbit code, C library for audio sample rate conversion
http://www.mega-nerd.com/SRC/

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

[10] **QjackCtl** – JACK Audio Connection Kit - Qt GUI Interface
http://qjackctl.sourceforge.net/
http://sourceforge.net/projects/qjackctl/

[11] **Cakewalk** - Powerful and easy to use products for music creation and recording
http://www.cakewalk.com/
ftp://ftp.cakewalk.com/pub/InstrumentDefinitions/

[12] **SoundTouch** – Sound Processing Library
http://www.surina.net/soundtouch/

[13] **rncbc.org** - My personal web presence, blog, forum and other mundane material
http://www.rncbc.org/

# 26. Tutorials

Coming Soon.

# 27. Appendix I: Debugging Qtractor

Although Qtractor is a mostly stable program, there could be problems that eventually show up. This short guide will explain to you how to build Qtractor with debugging code built in, making it easier to locate where the code exhibits problems.

Rebuild it all from scratch, with:

```
./configure --enable-debug && make
```

Enable core dumps in a shell session:

```
ulimit -c unlimited
```

From the same shell comman 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 just named `core` or something like `core.`*1234* (*1234* is the process-id number of the crashing program) located on the last directory the program was current.

Load the core dump file into `gdb`

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

at the gdb prompt just enter:

```
gdb> bt
```

or

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

# 28 Appendix II:  Changelog

```
Qtractor - An Audio/MIDI multi-track sequencer
----------------------------------------------

ChangeLog

0.1.3  2008-05-02  The Frugal Damsel
```

- As one may find convenient sometimes, the global time display
  format (frames, time or BBT) may now be changed on the main
  transport time spin-box context menu.

- Left-clicking on the track list number column now toggles all
  track content clip selection.

- Prevent audio-buffer initialization mashups when editing short
  audio clips while playback is rolling and within clip region.

- Audio peak files gets a bit simplified, dropping the peak frame
  count from its header; peak waveform graphics are now rendered
  as straight lines when over the end of audio file.

- The drop-span option (View/Options.../Drop multiple audio files
  into the same track) now also applies when importing tracks (as
  in Track/Import Tracks/Audio...) to concatenate multiple audio
  clips into one and the same new track.

- Audio and MIDI meter level colors are now user configurable (as
  global configuration options, View/Options.../Display/Meters)

- First attempt for Qt4.4 build support, regarding the bundled
  atomic primitives, which have changed upstream as advertised
  (thanks to Paul Thomas, for spotting this one first time).

- Record monitor switch is now an accessible button option on all
  track mixer strips; for visual consistency, the old bus "thru"
  switch button has been renamed to "monitor".

- Force track-view position reset to origin on session close.

- Fixed segfault on inserting an external file into files widget.

- Mixer splitter sizes are now better saved/restored when closed.

- Track record monitoring is now a state option, being toggled
  from the Track/State/Monitor menu; applies both to audio end
  MIDI tracks: when set all input will be pass-through to the
  current assigned output bus, including track plug-ins chain.

- Session dialog gets split in its own tab components, between
  descriptive, time and view configuration ones.

- Drifting correction among audio and MIDI engines is now back,
  but avoided while recording or should it be while looping?
  (EXPERIMENTAL REGRESSION)

- Time-stretching percent value gets its semantics inverted,
  as thought consistent with ones general sense for relative
  stretching ie., lower to shrink and higher to make longer.
  this is a major up-side-down change and should affect all
  sessions saved with time-stretched audio clips.

- Slack space in main tracks and MIDI clip editor views are now
  proportionial to viewport width, leaving enough room for drag
  and moving content past the current session length, specially
  at the lower zoom levels.

- Clip end time is now also shown on tool-tip.

- When armed for recording, MIDI tracks are now monitored and
  filtered through their own output bus, thus having the same
  behavior as audio tracks; this also implies that all record
  armed tracks won't playback their current content material
  when recording is engaged and rolling; track mute and solo

states are now honored on record monitoring.

- Audio clip pitch-shifting makes its first appearance, with
  the optional help from Chris Cannam's RubberBand library.

- A new MIDI editor tool is available: note/pitch randomize.

- Avoid (re)setting the default session directory if a session
  cannot be open or loaded for some reason.

- Another nastiness bites the dust: a subtle but progressive
  drifting has been swept away from the audio buffer looping;
  zero buffer flushing is now also taken into account, which
  was the cause for serious drifting on time-stretched clips.

- A major digital audio processing bug was tamed: audio clip
  fade-in/outs are now linearly piece-wise applied, even at
  the clip edges, giving a much smoother rendering and thus
  mitigating the nasty click-and-pop artifacts that were in
  fact due to some early design optimization with a poor and
  sloppy implementation.


0.1.2  2008-03-23  The Frantic Dame

- Session length fixed (yet again) while extend recording; also
  improved follow-playhead switching while playback/recording.

- Whitespace sanitization gets leaner for all recorded filenames.

- Run-time SSE optimization detection has been improved while on
  configure; additionally, IEEE 32bit float specific optimizations
  have also sneaked in.

- SSE optimization is now featured over all audio monitoring, and
  most specially on audio bus buffering, lowering the CPU burden a
  bit while doing track and bus gain, pan, metering and mix-down.

- Fixed MIDI clip move into new track, preserving the original
  channel, bank and program whenever possible.

- Fixed session cursor seeking, specially regarding overlapped
  clips, once gain.

- The MIDI editor gets new menu access to current MIDI clip track
  (see File/Track/Inputs, Outputs, Properties); selection of MIDI
  events has also been improved, specially regarding overlapped
  note events.

- Clip split command enters the stage (see Edit/Clip/Split) about
  splitting the current (selected) clip at the current playhead
  position (red cursor line). (EXPERIMENTAL)

- Creating new clips from scratch is now finally permitted (see
  Edit/Clip/New...); additionally, the clip properties dialog is
  now also allowing for changing the filename (and track/channel
  as special to MIDI clips). (EXPERIMENTAL)

- Record armed tracks are now properly monitored and fed through
  their own output audio bus on mix-down, which includes plug-in
  effects processing (EXPERIMENTAL).

- The files widget get alternating coloured rows.

- VST plug-in preset values are now being restored properly;
  individual parameter changes are now being queued for the
  also convenient undo/redo command pattern.

- Some audio clip buffer-sync tweaks have sneaked in, improving
  and fixing the rendering of full-overlapped, integrally cached
  and/or offset clips altogether.

- Stuffed one primordial shot on XInitThreads() at the main head,
  and let native VST plug-ins start behaving as they should, or
  not; this might be in fact problematic and dangerous for people
  who won't ever try the JUCE based plugins as from JUCETICE
  (http://www.anticore.org/jucetice), due to some broken locking

mechanism in xcb; thanks anyway to mighty kRAkEn/gORe@JUCETICE
for this precious hint and from who knows best.

- True deterministic session length update has due fixed.

- Track menu has new accessible actions:
  Track/Inputs - show current track input bus connections;
  Track/Outputs - show current track output bus connections;
  Track/State/Record - arm current track for recording;
  Track/State/Mute - mute current track;
  Track/State/Solo - solo current track;
  Track/Navigate/First - make current the first track;
  Track/Navigate/Previous - make current the previous track;
  Track/Navigate/Next - make current the next track;
  Track/Navigate/Last - make current the last track;
  Track/Move/Top - move current track to top;
  Track/Move/Up - move current track up;
  Track/Move/Down - move current track down;
  Track/Move/Bottom - move current track to bottom;

- View menus have new accessible actions:
  View/Zoom/In - horizontal and vertical zoom-in (Ctrl +);
  View/Zoom/Out - horizontal and vertical zoom-out (Ctrl -);
  View/Zoom/Reset - reset both zoom levels to default;
  View/Snap - select current snap-per-beat setting;

- Plug-in forms don't auto-open on session reload anymore.

- Keyboard shortcuts icon item (Help/Shortcuts...) sneaks in.


0.1.1   2008-02-16   The Futile Duchess

- After some great user demand, keyboard shortcuts are finally
  configurable, as found provisionally under Help/Shortcuts...,
  for the main application menu and for the MIDI editor as well.

- Debian package gets SSE optimization disabled as default.

- At least some transport actions get to be non auto-repeatable
  when pressed for much too long, as Play and Record, avoiding
  the tumbling imposed from the keyboard.

- For the first time ever, jackd auto-start is now allowed (!).

- OSC service support through liblo gets optional at configure
  time, now leading the way to proper DSSI plug-in hosting.

- All plug-in widget controls count are now capped to one hundred.

- Plugin paths setup is now made available on the options dialog,
  overriding each of respective default settings, as implicit from
  the LADSPA_PATH, DSSI_PATH and VST_PATH environment variables
  (see View/Options.../Display/Plugin Paths).

- Clip fade-in/out lengths are now kept relative to tempo changes
  and also to clip offset and length changes (clip resizes).

- Automatic time-stretching for all audio clips when session tempo
  changes, may now be disabled/enabled as a global session option
  (see View/Options.../Audio/Playback/Automatic time-stretching).

- Double-clicking on an empty area (de)selects all clips on track.

- MIDI capture (record) quantization is now an option, possibly
  handy for some jerky performance musicians, as the one found
  in myself ;) (see View/Options.../MIDI/Capture/Quantize).

- The global options dialog (View/Options...) has seen its Display
  tab page being moved back and to the right.

- Major rewrite of the plug-in infrastructure, adding primordial
  support for DSSI and native VST plug-in flavors.

- Drag-and-drop of plug-in instances are now allowed intra- and
  inter-mixer strip chains, either on tracks or buses.

- Turning track record off while recording is rolling was leaving
  the session in a inconsistent recording status, now fixed.

- A random but instant crash upon audition/pre-listening player
  onset was hopefully fixed.


0.1.0   2008-01-05   The Frivolous Debutante.

- Audio clip time-stretching makes its debut, with code adapted and
  refactored from the SoundTouch library, under the (L)GPL umbrella.

- New "Options.../Audio/Playback/Quick seek time-stretching" global
  option, providing a quick seek mode (hierarchical search) while
  doing all audio buffer time-stretching.

- Changing session tempo will automatically apply the corresponding
  time-stretch percent factor to all in-place audio clips. Audio
  clip dialog also includes a new time-stretch property setting.

- Tempo changes are now affecting clip offsets correctly, keeping
  the clip offset constant in time units (ticks), as are clip start
  and length properties already.

- Mixer splitter sizes are now properly saved/restored when hidden.

- Extended multi-selection is now featured on the files widget; all
  drag and drop functionality has been almost completely rewritten.

- SSE optimization is now enabled where available (via configure).

- Options for having separate dedicated ports for the audition/pre-
  listening player output, audio metronome output, MIDI control
  input/output and MIDI metronome output, are now in place.

- A brand new subtle option sneaks in, affecting the drag-and-drop
  of the main track-view: View/Options.../Drop multiple audio files
  into the same track, whether to drop multiple external files into
  new or existing track as concatenated audio clips.

- The audio metronome makes its debut as an alternative to the MIDI
  existing one; parameters include bar and beat audio sample files,
  accessible from the View/Options.../Audio/Metronome dialog.

- Files widget action refactorization; the files context menu gets
  its due item icons and a new menu item for direct audio player
  accessibility.

- MIDI time resolution changes (ppqn, ticks per beat) now tries to
  keep all session MIDI clip times unchanged as far as possible.

- MIDI channel volume and panning control change events, CC#7 and
  CC#10 respectively, are now rendered unfiltered on playback.

- First rendition of the long due implementation of an audition or
  pre-listening audio player is now in place; the files window got
  this new play/stop control button on its lower-right corner.

- Actual instrument definition note (keys) and controller names are
  now in effect on the MIDI editor, whenever applicable.

- Fixed instrument bank selection method, "Bank MSB" (2), which was
  broken enough to never send the correct bank number.

- Mouse-wheel horizontal scrolling is now accessible on every view,
  while pressing a modifier key (Shift or Ctrl).

- New auto-backward transport option: when enabled the playhead will
  be reset backward automatically whenever transport stops playing.

- A suicide-crash has been fixed while invoking the bus dialog from
  the respective mixer bus strip context menu.

- Master (default) buses are always set to Duplex mode, being now
  an enforced update policy while on the Buses dialog.

- A stupid lockup bug (infinite loop) was spotted on the track bus

assignment method and squashed (thanks, lexridge).

- New keyboard shortcuts for toggling the Connections tool (F8) and
  the mighty Mixer tool (F9) windows.

- Avoid showing a context menu while right-clicking on time rulers.

- Audio clip waveform drawing gets additional closing points.

- It is now possible to change the length/size of a clip by dragging
  its left or right edges, while in the main track-view. Shift+drag
  will also time-stretch to the resulting audio clip length.

- Another off-by-one mistake was corrected, which was causing audio
  clips to go out-of-sync on loop turnover boundary; also changed
  the loop turnaround strategy, now honoring already cached periods.

- A race-condition has been mitigated in the audio-buffer thread,
  that was exposed and lead to sudden application freezing upon
  changing some composite audio clip commands.

- Take absolute audio peak values only, making peak files a little
  bit shorter and hopefully faster to load and draw as waveforms.


0.0.9   2007-11-30   The Adolescence Prime

- Drag-and-dropping of MIDI files without specific track or channel,
  into existing tracks, is now rejected. The drop operation is now
  allowed on the track-view blank area only, meaning the same as the
  complete MIDI file import into session.

- Record actual MIDI clip length to last play-head position,
  instead of time of last event in the recorded sequence.

- Connections item lists gets properly sorted, as intended.

- Clear connection persistance once an explicit discconnection is
  issued on any of the (intrinsic) bus ports.

- MIDI output buses now get the panning slider to spit out some
  GM system master balance (sysex) messages, being now enabled.

- Mouse hovering on the clip fade-in/out handles, while in the
  main track-view, gets its long due cursor pointer feedback.

- Fixed a off-by-one boundary issue on MIDI clip event playback,
  which were enqueuing duplicated MIDI events on every read-ahead
  output thread processing cycle (1 sec).

- Transport menu and toolbar are now featured on the MIDI Editor.

- Use actual session name when asking to save changed session.

- Transport loop setting keyboard shortcuts swapped: Ctrl+L will
  set the loop immediately, and Shift+Ctrl+L will toggle on/off.

- Changing MIDI event duration may now affect MIDI clip duration.

- Alternate sharp-note color lines have returned to MIDI Editor
  canvas, as was the shadow color marking the end of MIDI clip.


0.0.8   2007-11-12   The Twelfth Tight

- Bus context menu is now accessible from respective mixer strip.

- Fixed a subtle crash-suicide issue when invoking the bus dialog
  with a double-click over the corresponding mixer bus strip; also
  fixed the sloppy ganguing mistake when changing mixer bus gain
  (volume) and panning values.

- MIDI Omni mode (sort of) makes its entrance as a new MIDI track
  property, meaning that is now finally possible for the capture of
  any unfiltered MIDI channel event, without regard to the current
  channel assignment, which still applies for playback purposes.

- Audio (pass-)through has been also implemented, now being a common
  and consistent property of both audio and MIDI buses, provided
  those are set in duplex mode (input and output).

- Re-touched follow-playhead and continue-past-end tool icons, again
  to be a bit softer and not so bright as to hurt someones eyes.


0.0.7   2007-10-31   The Eleventh Tower

- Suspend auto-follow-playhead while dragging or moving content over
  the main track and midi editor views; re-touched follow-playhead
  and new continue-past-end tool icons.

- Playback is now forced complete full-stop whenever play-head goes
  behind the current session length and/or loop-end and the newer
  "continue past end" option is enabled from transport menu.

- Tool/child windows position and size preservation fixed.

- MIDI (pass-)through has been finally implemented, after several
  kind requests, it applies as a property of duplex-mode MIDI buses;
  this new setting is configurable from the View/Buses... dialog;
  also from respective new input bus mixer-strip button; when enabled,
  implies all incoming MIDI events at the input bus will pass-through
  unchanged to the corresponding output bus, as found useful just for
  direct monitoring one's performance without the help of any extra
  circumvent or kludgy connections.

- Mouse cursor shape changes accordingly while hovering in header
  time-scale rulers, both in main track-view and MIDI editor.

- Track-view clipboard paste action has been refactored, with the
  user interface consistent with the same functionality featured on
  the MIDI editor: the clipboard selection being pasted now floats
  at the mouse pointer and can be moved around before placed into
  its final position, either with the mouse or keyboard arrow keys.

- A unitialized member variable bug affecting all MIDI clips has
  been discovered and squashed; this one has been lurking for quite
  some time and was causing wrong clip editing results, specially
  when tempo or meter differs between session and the MIDI file.

- Keyboard step-moving is now allowed while pasting in MIDI editor.

- Track-view clip selection can now be drag-moved into the void
  (bottom) view area creating brand new tracks automagically.

- Losing focus resets all current keyboard step-moves in progress,
  affecting the main track-view and all MIDI editor as well.

- As done before on the MIDI editor, the main track-view current
  selection may be step-wise moved using the keyboard arrow keys
  and the enter/insert keys for final placement; horizontal step
  movement is quantized according to current snap setting; vertical
  key-step movement is only allowed to selected clips belonging to
  one single and the same track.

- Immediate session loop settting, accessible from the main menu.

- Track properties dialog gets fixed again in its auto-size treat.

- Another audio-buffer thread bug was scrubbed off, which was
  causing spurious and audible garbage on certain loop workloads.


0.0.6   2007-10-09   The Tenth Commencement

- A bit more of precision is achieved over the metronome regular
  ticking and both the audio and MIDI monitor meters.

- A rudimentary MIDI metronome is now in place; parameters, like
  MIDI channel, bar and beat accent note, velocity and duration,
  are readily configurable from the main menu, View/Options...
  /MIDI/Metronome dialog.

- Track properties dialog now gets tightly auto-sized, depending

whether its an audio or a MIDI track.

- MIDI clips are now auto-extendable when adding or moving events
  beyond the clip length and while in the MIDI Editor, of course.

- MIDI editor current selection can now be step-wise moved using
  the keyboard arrow keys and the enter/insert keys as for final
  placement; horizontal step movement is subordinated to current
  snap per beat setting, no less than unity; vertical step moves
  are obviously quantized to the next semitone.

- Get configure to try and detect the correct qmake location and
  insert it the search order, so let the qt4 tools take precedence
  when --with-qt option is given and older qt3 ones coexist and
  are found located ahead in the PATH.

- Drifting correction on audio and MIDI engines was seriously
  infected in some kind of snafu conception, as evidence lead
  to even worse drifting being detected to much of great despair,
  specially after recording and/or bouncing audio tracks from
  MIDI sequenced material. Credit must certainly go to Christian
  Schoenebeck on splatting this sloppy one on the face.

- MIDI editing actions while playing now get immediate feedback;
  this was possible to some internal interface redesign of all
  MIDI editor accessory classes, making the MIDI clip now being
  the main editing target object instead of just the MIDI event
  sequence as it was previously.

- Simple as it could ever be, the build executive summary report
  is now given on configure.

- The internal decoded frame list for MPEG 1Audio Layer III (mp3)
  audio files (ie. via libmad) has been made one-time cached as
  global shared objects, benefitting from the fact that the list
  is always completely (re)built during the peak file computation,
  and thus speeding up all frame accurate access operations (seek)
  over this specific audio file format.

- More eye-candy is sneaking in the MIDI editor: there's new view
  options on the View menu: Note Color and Value Color, affecting
  note event colors according to pitch and/or velocity.

- New view option on the MIDI editor: on menu View/Note Duration
  switches whether events are shown proportional to their durations
  or as simple vertical candlesticks.

- New snap-per-beat divisors are now available (Beat/3, 6, 12, 24
  and 48), giving support for triplets for the very first time
  (after a heads-up suggested  patch from Marko, thanks).


0.0.5   2007-09-08  The Ninth Hitch Nail

- MIDI edit tools (quantize, transpose, normalize, randomize,
  resize) are all functional and ready for experimentalism;
  gets in its own top-level menu and form with named preset
  store and recall functionality (EXPERIMENTAL).

- Main form backward and rewind transport actions are now being
  immediately enabled when playing from the session (zero-time)
  start position.

- Audio and MIDI export sneaks in and in form and accessible
  from the main Track menu (EXPERIMENTAL).

- MIDI track channel is now properly set on session (re)load;
  track background color changes was missing the alpha setting.

- The mix-down buffering was fixed again, now taking multi-track
  overlapping clips into consideration (was a lot more broken
  since the recent glitch-looping fix).

- Dirty MIDI clip editing control has been fixed but still
  somewhat hacky nevertheless.

- First attempt on solving a nasty MIDI editor bug, which was

quietly and severely crippling MIDI files while saving offset
edited clips.

- Session loop (re)setting is finally now an undoable command.

- Yet another insidious bug has been swept away from under the
  carpet: once again on audio looping, some astonishing old and
  crappy session cursor seek-backward statement was lurking to
  be laid off. Gone now, simply as it is, growing old on this :)

- A tremendous bug has been fixed: audio looping is now a little
  more glitch-free as the mix-down buffering was badly broken
  even since its primordial implementation. Rejoice.

- Minor improvements on track-view cursor updates and visual
  tracking while recording.

- MIDI editor windows get their keyboard accelerator/shortcuts
  back in business, whether opted as tools always on top or not.

- The infamous "Keep tool windows always on top" global option
  is now infecting the Connections, Mixer, Plugin and MIDI Editor
  window instances with no probable regrets. To be used with
  discretion, of course.

- Session update/initialization gets it clean on startup.

- Range selection action (Ctrl+R) is now back in business with
  the added bonus of being accessible from all MIDI editor
  instances too.

- Common edit-head and tail cursor positions are now under common
  control and display from all MIDI editor instances; the session
  loop-start/end points are now also shown on every time-line and
  share the same control behavior across all MIDI editor instances.


0.0.4   2007-07-19   The Eighth Wanderer

- Main toolbar tempo spin-box gets loose from keyboard change
  tracking (Qt >= 4.3); custom spin-box compilation fix for
  Qt 4.1.

- Illusive but nasty Connections/Patchbay item tooltip crash bug
  has been hopefully fixed (Qt >= 4.3); QComboBox::editTextChanged()
  signal replaces old QComboBox::textChanged().

- Combo-box setup history has been corrected on restore, which
  was discarding the very initial default (factory) contents.

- Make debian package build depend on libqt4-dev; win32 console
  flag is back to qmake project file.

- MIDI instrument selection (e.g. on track form) gets fixed and
  improved.

- Sorting method for the connections port list has been refactored;
  potential crash bug fix on connections sorting method.

- Messages class accessor methods constness fix.

- Got rid of some autoconf redundand thingies on configure;
  late debian changelog update.

- Desktop categories update: AudioVideo.

- README correction.


0.0.3   2007-06-23   The Seventh Draught

- Crash fix on the connections widget, was a matter of refactoring
  the refresh/clear slots.

- Help menu added to MIDI editor form (redirected and same to main
  form anyway).

- Clip properties form gets its proper sanity check, querying any
  existing clip editor whether its safe to apply the new settings.

- Mixer sliders get their long due valued correction, hopefully.

- Transport backward and play/stop are now made accessible from
  the MIDI editor widget, through their keyboard accelerator
  shortcuts (backspace and space, respectively).

- Transport actions (play, rec, rew and ffwd) are now kept stable
  on a single point, instead of being scattered all over the main
  form code; transport visual feedback might get affected,
  specially regarding the MMC processing.

- Application icon is now officially installed
  into ${prefix}/share/pixmaps.

- Spec file is now a bit more openSUSE compliant; just made RPM
  requirements as exigent as the new debian ones.

- Paste cursor is now properly preserved after leaving MIDI editor
  views.

- Amazingly why this was not spotted before, the main application
  logo-icon has been downscaled to the 32x32 pixel standard icon
  size.

- MIDI clip editor clipboard gets singleton status and is now
  shared on all MIDI editor instances. Similarly to the main
  track-view, shift/ctrl-left-clicking on the MIDI editor views
  sets the current session play-head position.

- A desktop entry file has been included on install, at last.

- Clips in main track-view get more info in the form of tooltips.

- Major rounding fix to time-scaling and most specially on all
  those internal MIDI I/O methods.

- Extended range selection from the time-ruler and key-list
  headers is now possible by click-and-drag the mouse pointer.

- Play-head cursor is now also displayed and/or set position
  on all open MIDI clip editors time line view. As in the main
  application form, a new local follow play-head option is also
  featured on the MIDI editor view menu and toolbar.

- All MIDI file save operations are now logged to main messages
  and files are added to the main files list view.

- Initial debianization.

- MIDI capture/record file format (SMF Format 0 or 1) is now
  an user option, introducing the new MIDI tab on the global
  View/Options dialog.

- A bad old-time session cursor glitch has been apparently fixed.

- Make sure the generic clip properties form is modal.

- Major rewrite and adaptation to the session time-scale properties,
  making its way for a brand new command instance: the session-tempo
  command.

- Changing the snap-per-beat combobox value on the main window
  toolbar does not make it as an undoable command anymore.

- Long due MIDI clip editor integration has come to reality.

- Major rewrite on the MIDI sequence file read/write methods,
  in preparation to the coming MIDI clip editor.

- Status-bar session length label now gets rightly updated,
  while extended recording, of course.

- Clip properties fade-in/out lengths, gets their old due
  constraints.

- Main transport time display is now an editable custom spin-box;
  the Tempo (BPM) spin-box has seen new colors (green on black).

- Transport time display format option adds the new choice of
  absolute frame number, alternative to previous time and BBT.

- The frame-time based spin-boxes, on the clip properties form,
  were replaced with a new custom one, allowing for alternate
  frame, time and bbt input/display formats.

- Time-scale helper class has been introduced.


0.0.2   2007-05-26   The Sixth Lord

- Audio/MIDI connections gets slightly refactored, contributing
  for whole robustness, specially in case of incidental engine
  shutdown.

- Mixer window gets a minimum default height bump.

- Clip fade-in/out type changes have been properly fixed.

- Complete refactoring of the command class pattern, making it
  now derived from QObject and not having a reference to the
  main form anymore.

- Inoperative context menu event handler has been removed from
  the track-view.

- MIDI sequence note-on event tracking now uses faster QHash
  class, instead of original QMap.

- Off-by-one bug fix on MIDI track write method, while parsing
  co-incident note-on/off events in the wrong order and thus
  leaving note events with an invalid (zero) duration; obviously
  affecting MIDI recording in very special circumstances.

- Minor and rather innocuous drop at this time in the MIDI event
  class structure: the flags member field.

- Port connections now get their lines correctly drawn; strangely
  enough, the connection lines were being painted only for the
  parent client items, probably since the Qt4 migration (aka.
  Halloween files).

- Early clip editing is in place (clip name, start, offset and
  length parameters, fade in/out length and type).

- Some menu item text capitalizations.


0.0.1   2007-05-07   The Filthy Fifth

- Newer JACK 0.105.0 seems to bitch, probably correctly, about the
  return value of the process callback. Make it to bitch no more by
  ensuring the JACK client is always issuing the innocuous 0 (zero)
  return value.

- Important fixes have been issued, affecting MIDI recording: MIDI
  sequence zero-time event insertion; MIDI file pending note-off
  processing on write-track method.

- Qt4 migration complete.


0.0.0   2006-10-31   The Halloween Files

# 28 Appendix III:  User Documentation Changelog

```
0.1.3 2008-04-30

- Update librubberband requirement.

- Updated Menu Items.


0.1.2 2008-03-16

- Updated Figures 4, 5 and 7 with updated screen snapshots.

- Updated text on Adding new Clips, and Splitting Clips.

- Changed formatting from US Letter to A4.

- Added an Appendix 2 for the Qtractor Changelog.

- Added an Appendix 3 for the Manual Changelog.
```