

Table of Contents

1 Introduction.....	2
1.1 What is Rosegarden?	2
1.2 This Guide's Raison d'être	2
1.3 Requirements	2
1.4 Intended Audience	2
1.5 The Author's Studio	3
1.6 History	4
1.7 Getting This Document	4
1.8 Copyright and License	4
1.9 Translations	5
2 A First Look.....	6
2.1 Starting Rosegarden	6
2.2 Main Window Overview	6
2.3 Manipulating Segments	8
2.4 The Studio	9
2.4.1 Instruments, Banks and Devices	10
2.4.2 No Global Instrument Definitions	10
2.4.3 A Studio Dissected	10
2.4.4 Loading An Instrument Definition	11
2.4.5 Creating A Custom Device Definition	14
2.4.6 Assigning a MIDI Device/Port and Channel	16
2.4.7 Changing the Track Name	16
3 Project 1: Importing a MIDI file.....	18
3.1 Importing the File	18
3.2 Transport Tools	19
3.3 Tweaking the Import	20
3.4 Instrument Parameters	22
3.5 More About Segments	23
3.5.1 Segment Previews	23
3.6 Autosave (CVS)	23
4 Let's Edit Some Notation.....	25
4.1 Opening the Notation Editor	25
4.2 Notation Editor Overview	25
4.2.1 Controlling Playback	26
4.2.2 Changing the View	27
4.3 Transposing	27
4.4 De-Humanizing	28
4.4.1 Smoothing	28
4.4.2 Quantize	28
4.4.3 Adding Marks	29
4.4.4 Auto-Beam	30
4.4.5 Changing Note Durations	30
4.4.6 Let's Add Some Slurs	30
4.4.7 Manual Beaming	30
4.4.8 The Text Cursor	31

Table of Contents

<u>4 Let's Edit Some Notation</u>	
<u>4.5 A Change of Key</u>	32
<u>4.5.1 Multiple Staves</u>	32
<u>4.6 Printing</u>	32
<u>5 Using Audio</u>	35
<u>5.1 Configuring an Audio Editor</u>	35
<u>5.2 Importing Existing Audio</u>	35
<u>5.3 Recording</u>	37
<u>5.4 Manipulating Audio Segments</u>	38
<u>6 That's All, Folks</u>	40

1 Introduction

1.1 What is Rosegarden?

Rosegarden-4, hereafter referred to as simply Rosegarden, is a versatile application that seeks to be a useful tool for those wanting to create music. It combines aspects of a MIDI sequencer, an audio sequencer, and a notation editor into one convenient, easy to use package that provides users with a consistent and mostly intuitive interface.

Rosegarden is a native KDE application that can run on any system that has the KDE and QT libraries installed, and currently requires a minimum of KDE/QT 3.0 in order to run. Many of its features may only work Linux systems, but it is designed to compile and run on any Unix-like platform capable of running KDE.

1.2 This Guide's *Raison d'être*

My goal in writing this guide is to help you explore Rosegarden, and to discover ways to use its features to solve real problems that you might encounter when creating your own music. Rosegarden is easy to use, once you come to understand its framework, but it's not quite like anything you are likely to have used in the past. As such, I wish to take you on a guided tour through Rosegarden, to demonstrate many of its capabilities, and to offer you a foundation from which your own creative projects can spring.

My intention here is not to write an exhaustive technical manual that covers every single possibility in dry detail. This is a meandering journey through Rosegarden, taking you through real-world projects, and showing you things of interest along the way. I hope that by sharing my experiences and explaining how I've used Rosegarden to accomplish various and specific goals, we'll both be able to get more out of time spent with this application.

1.3 Requirements

I assume that you have at least Rosegarden 4 version 0.8.5, and that it's working properly. I leave it to you to surf to the Rosegarden download page at <http://www.all-day-breakfast.com/download.html> and work outward from there if you don't already have a copy. Several distros already have a rosegarden4 or rosegarden-4 package available through mainstream or third-party channels, and if it isn't already, yours will probably be among them soon.

NOTE: Rosegarden is under constant development. As such, some details or screenshots described here may not exactly match the software in front of you. I intend to keep this guide useful, but I don't have the energy to re-write it and create new images from the top down every time someone adds a new icon somewhere or reconfigures a dialog box. I'm attempting to keep this guide useful to those who are conservative enough to stay away from CVS, but I may slip. If something I'm writing about doesn't seem to exist, chances are that it was added since the last official release.

1.4 Intended Audience

The demonstration projects featured in this guide are all based on problems that I have solved for myself in the time since I began using Rosegarden. They are strongly influenced by my own musical interest as an amateur, untrained composer, arranger, and musician.

If it's not something I do, I'm not well-equipped to write about it, and so it behooves you to see where I have gone, and where I'm going, to decide whether my way of using this package bears any relevance to what you want to do with the software yourself. Most notably, I am not an audio guy, and I barely know anything about the subject. If you're dying to dig into some complex audio project, and you know what terms like "DAW," "ADAT," and "MMC" mean, then I'm afraid I will be of little help.

One might say that Rosegarden is my band in a box. I have precious little contact with musical people, and so I make heavy use of my computer as an accompanist in a variety of ways. I find that it is much more rewarding to play music with other people, even if those "other people" are actually me, or a synthesizer. When I'm not accompanying myself with other live instruments, I can take advantage of the vast array of MIDI files available on the internet as an endless source of material for fun projects. (Believe me, there's no other way I would ever get to play Bach. Real musicians would be too busy laughing at me to play their parts. :)

I involve a wide variety of source materials in these projects, but for the purpose of this guide, I will stick with completely original compositions, so as to avoid any potential copyright issues. They may not be *good* compositions, but I own the rights to them, and can mangle them any way I choose without fear of any consequence more dire than rotten food being thrown at my head.

1.5 The Author's Studio

Obviously, what I can actually demonstrate is hardware dependant. I'd like you to bear in mind what is sitting in front of me as I write this, and thus I will share the details of my "studio."

I'm running a 1.0 GHz Athlon system with 256 MB of RAM. The box runs Debian Woody with aftermarket KDE 3.0.3 from a third-party apt-source, and a variety of other add-ons that I've had to compile for myself along the way in order to get my build environment in line for bleeding edge software such as Rosegarden. I'm running a completely stock kernel 2.4.19 that has had no patches applied. (I probably should look at latency tuning, but I am leery, and have thus far avoided it. I enjoy stability, and am not keen on giving applications the authority to trample all over my kernel.)

On the audio side of things, my sole soundcard is a Sound Blaster PCI 128, driven by ALSA 0.9.0rc6 and Jack from CVS, with Woody's supplied LADSPA SDK and plugins.

On the MIDI side of things, I have a joystick port MIDI box for the Sound Blaster, to which I attach a Fatar MIDI-72 controller keyboard and a Roland SC-33 Sound Canvas. All of the source MIDI files that I work with in this guide were sequenced to make use of this particular GS synth, and as such, some of what I say in that area will not apply or even make sense to you if you're using an XG or other type of synth to follow along.

In addition to the Roland, I'm running an ALSA soft synth via iiwusynth 0.2.2. Details of getting these working successfully are covered in the [FAQ](#). I have it compiled against Jack, but have not yet managed to actually get the two to co-exist, so I run one or the other, depending on what I want to do with Rosegarden.

I run Rosegarden from CVS, which I build pretty much every day. As of this writing, the latest stable version is Rosegarden-4 0.8.5, but that version is very old and boring. CVS is where all the action is.

1.6 History

Date Notes

12/10/02 Initial public draft

12/15/02 Merged into CVS tree

12/16/02 First re-write begins...

12/19/02 Continues...

01/01/03 Conversion from "tutorial" to "guide" and shift of development model

01/12/03 Added Studio, re-structured to introduce segments in a logical place

01/19/03 Greatly improved Studio chapter, multiple edits and tweaks

02/19/03 Minor cleanup, preparation for the first translation, Studio improvements...

1.7 Getting This Document

The latest version of the *Using Rosegarden* is always available from

<http://rosegarden.sourceforge.net/tutorial/using-rosegarden/using-rosegarden.html>

You'll also want to snag [toc.html](#) to get the Table of Contents. If you intend to print this, you can grab [using-rosegarden.pdf](#) from the same site. (The PDF version includes the Table of Contents.)

You can also get it from CVS in the same way you get Rosegarden. It's part of the module called **tutorial**. I'm too lazy to go dig out and prettily format the instructions for this at the moment, but I've given you enough of a hint to get there if you dig for it a bit. :)

1.8 Copyright and License

Copyright © 2002, 2003 by D. Michael McIntyre <dmmcintyr@users.sourceforge.net>

Using Rosegarden is released, and its title and contents are protected under the terms of the GNU Free Documentation License, a copy of which is available at <http://www.gnu.org/licenses/fdl.txt>

I acknowledge that words like "Roland," "Yamaha," "General MIDI," "GS," "XG," and numerous others that I haven't even considered, but should have, are owned by important, well-monied corporations and probably need to have cute little symbols in circles or superscript or something attached to their names. I have far better things to do than try to figure out which cute little symbols are supposed to go on which names, but if representatives from any of the corporations who own those words would be so kind as to inform me which little symbols need to be attached to them, I will attach such symbols forthwith.

1.9 Translations

A German translation of this document is coming soon. When it exists, I'll make appropriate acknowledgements here. :)

Folks interested in translating this to other languages can drop me a line at mmcintyr@swva.net (my "direct" line.. :) At this time, getting into a translation means a big job of keeping up with this as I change it to reflect the ever-changing Rosegarden, but now that I have one translator, I'm writing with that in mind, and making a conscious effort to avoid forcing the translators to do unnecessary work. If you want to wait, I'd advise it, but if you're eager to start, there are some sections that are relatively safe to translate at this time. We can talk about it.

It's all new to me, being translated. I used to **be** a translator, but I always try to avoid translating something **into** a foreign language when I can, and have no plans to translate this to other languages for myself. We'll just have to see how it goes and feel this out as we go along, but I'll do whatever I can to make your job easier.

2 A First Look

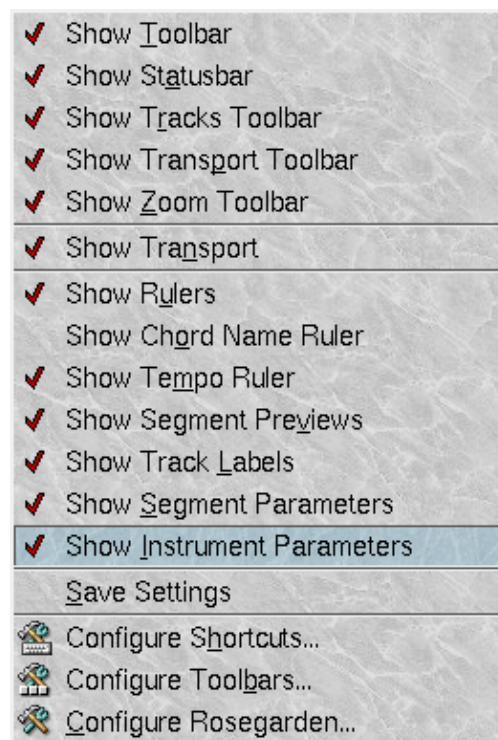
2.1 Starting Rosegarden

Let's begin by starting Rosegarden, and having a look at the **Main Window**. This is where you get an overall view of your composition, and control a number of settings and parameters.

TIP: You might want to consider running Rosegarden from a command line so you can see what's going on behind the GUI. What's gibberish to you may be a great help to the developers if there are problems, and it's sometimes reassuring to watch it working behind the scenes when it's too busy processing a particularly large bit of data to bother repainting its window.

In order to share a common frame of reference, let's turn on anything that isn't already displayed by default.

Use the **Settings** menu to toggle these options. Observe that I had not toggled on the **Chord Name Ruler** when I took this snapshot. Also note that this is, unsurprisingly, where a number of Rosegarden settings can be configured. We'll come back to some of those later on:



2.2 Main Window Overview

In the **Main Window**, just below the menu, you will see the...

Toolbars. In addition to the usual standard controls,



there are several toolbars specific to Rosegarden. You'll choose various cursors from the **Cursor Toolbar**,



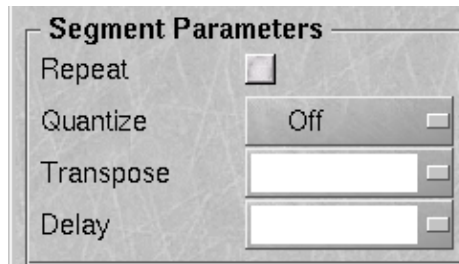
and there's a subset of transport tools available in the **Transport Toolbar**.



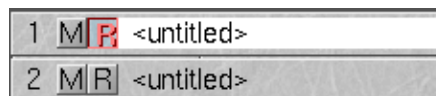
Next in line is the **Zoom** slider:



Moving down into the window itself, you'll find the **Segment Parameters Pane**, where several segment playback parameters can be set:



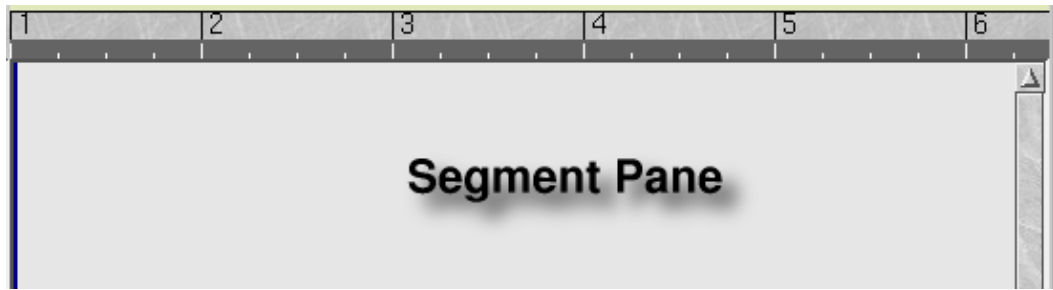
The **Tracklist**, where you manipulate your sequence on a track level. Notice that each track has a **Mute Button M**, to allow you to mute individual tracks, and a **Record Button R**, which selects the destination track for record operations.



Now we move into the meat of things with the **Segment Pane**, where a lot of editing operations take place. At the top, you'll find the various **Rulers**,




Then the **Segment Pane** itself, which is currently empty. We'll do a lot with this later.

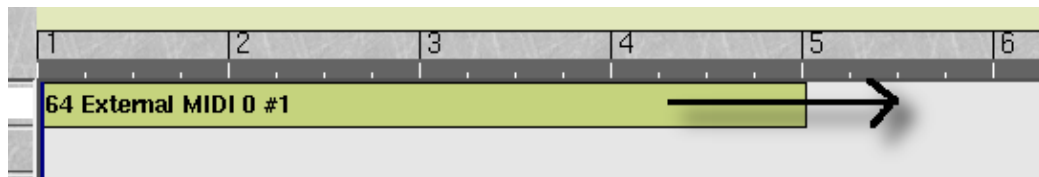


2.3 Manipulating Segments


Let's quit looking, and start *using* Rosegarden. Unless you begin with an imported MIDI file, it all starts with a **Segment**, which is a convenient container that stores a bit of note data or audio. Segments can be cut, pasted, split, re-combined, overlapped and pretty much rearranged at will, and are very flexible.

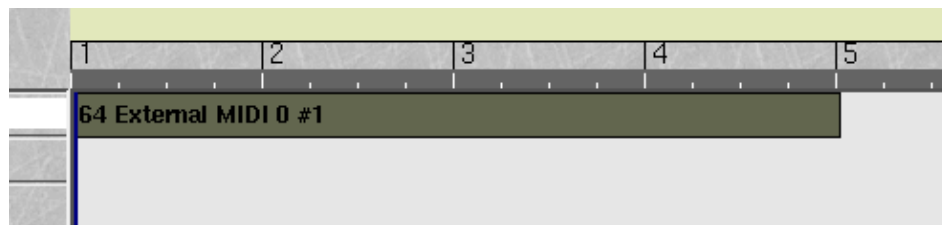
You can begin recording to create one, or you can draw one from scratch.

Use the  cursor to create a new segment. Click and drag it out to the desired length:

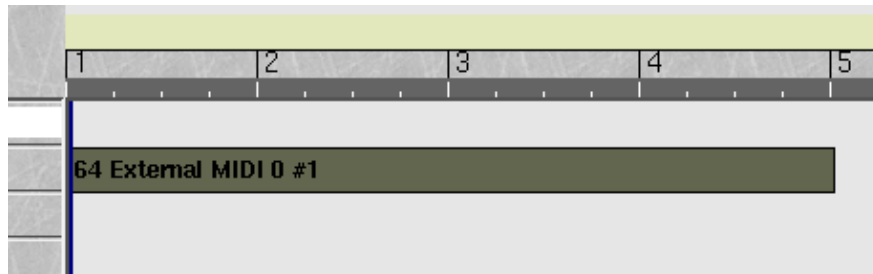


You can now edit it with the **Matrix** or the **Notation Editor**, which we will address later. For now, go ahead and play with the segment.

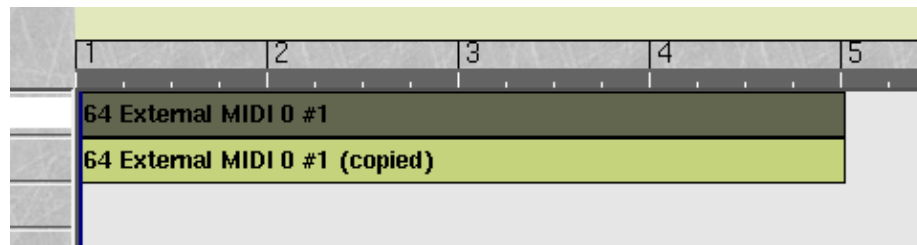
Use the  cursor to select a segment:




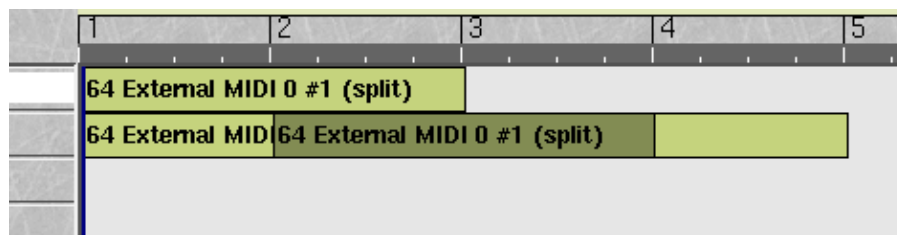
Drag it to a new track:



Hold down **Ctrl** while dragging it back, and make a copy:



Use the  cursor to split one of the segments, then one of the pieces so that it overlaps the segment on the other track:



*I don't have time at the moment to edit more graphics, but I'll mention that you can also merge segments. Select two or more segments on any number of tracks, and use **Segments** → **Collapse Segments** to consolidate them. I can't check that at the moment because my CVS copy of Rosegarden doesn't currently have a working Segments menu. Oops. ;)*

By now, you're probably wondering when we're going to do something useful, so let's go take a look at the **Studio**.

2.4 The Studio

Before importing anything into Rosegarden, we're going to want to set up instrument definitions for all of our MIDI devices so that any bank changes don't get lost in the translation. If there's no instrument definition for a particular program/bank change, then information will disappear at import time and the file will be interpreted as pure General MIDI. (This is of little consequence if you only use General MIDI, but setting things up in advance will save you from having to tweak any imported GS, XG or other device-specific files to correct the lost program changes.)

The **Studio** → **Manage MIDI Devices...** menu, is where all of this is accomplished, and it's your interface for relating to one of Rosegarden's most unique, and difficult to understand features.

NOTE: Many of these features were only recently added to Rosegarden. If you're not running at least version 0.8.5, you'll need to upgrade to 0.8.5 or CVS before you can make use of these features. If this chapter is completely baffling to you, check your version.

2.4.1 Instruments, Banks and Devices

Before you can accomplish anything with the **Studio** facility, you need to understand what a **studio** is. At this writing, the **Studio** is still under heavy development, and a clear picture of everything that it will grow to be has not yet emerged. The basic idea is to provide a central place for configuring an interface between Rosegarden and all the equipment in your own studio.

Every native Rosegarden file contains a **studio** definition that describes your **devices**. The idea is that once you have properly configured your **studio** it will be very easy to control all of your equipment without having to worry about the details of what is connected where. In its present state, it's somewhat underdeveloped, but I'll attempt to describe what we've got to work with now, and expect I'll have to make a lot of changes to this section over time.

A **device** is a layer of abstraction intended to provide Rosegarden with a way to relate to whatever MIDI devices you have connected to your computer. It is a data structure that contains up to numerous **banks**, each of which contains **instrument definitions** for up to 128 program changes. Each available MIDI output device on your system gets its own **device** definition, and all of these are rolled into the same **studio**.

If you have only one MIDI device, and it doesn't support anything more than the basic General MIDI instrument set, then the **studio** need only contain one **device**, with one **bank**, with a single set of **instrument definitions** attached to your only port, and you're in luck, because that's how Rosegarden comes configured out of the box. You can skip ahead to something more interesting.

If, on the other hand, you have two soundcards with a MIDI port each, and you're also using an ALSA soft synth, your **studio** probably needs to contain several **devices**. Any program or bank change that you wish to be able to transmit must have a name (or a number) defined as part of the **device** for that instrument. If your synth doesn't respond to program or bank changes, then you have little use for Rosegarden's **Studio** facility. Most others will want to define program changes of one form or another.

2.4.2 No Global Instrument Definitions

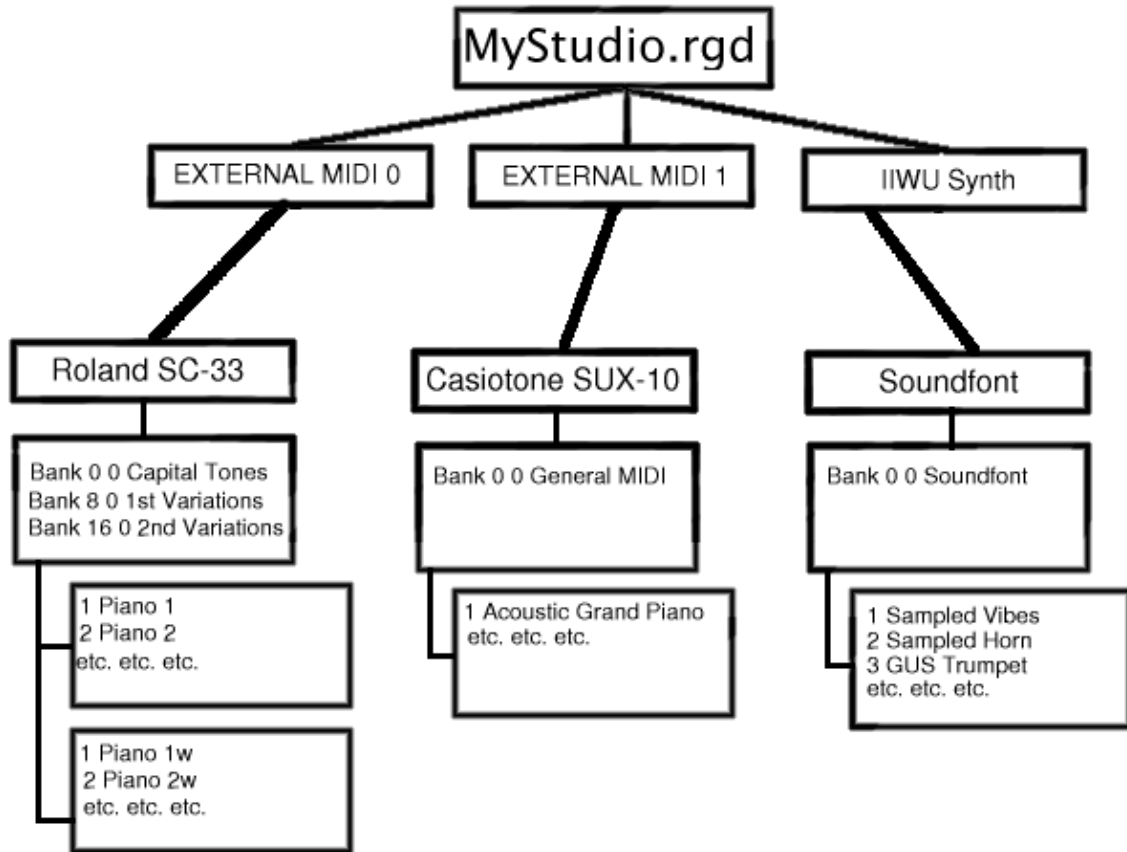
Every native Rosegarden composition (**.rg**) file contains a **studio**, along with note events and other types of data. It's a good idea to configure your **studio** early on, because anything you save with incorrect instrument definitions will need to be corrected later. It's not absolutely critical, because there's also a special subset of the Rosegarden format that stores just the definition for a **device** in a file with an **.rgd** extension, and it's possible to load all or part of the instrument definitions these files contain into existing compositions. However, the sooner you get things configured, the less work you'll need to do later.

Once you've configured your **studio**, you can save a copy of it to `~/.kde/share/apps/rosegarden/autoload.rg`, and it will be applied to anything that you subsequently create, including MIDI files that you import. It will *not* be applied to pre-existing **.rg** files you load.

2.4.3 A Studio Dissected

Let's take a look at what an **.rgd** file (or the **studio** portion of an **.rg** file) might contain. In this scenario, the user has two soundcards with an external MIDI port on each of them, and he is additionally running an ALSA soft synth on one of them to provide a "virtual" third MIDI port. He has a Roland SC-33 attached to the port

on soundcard 0, and a Casiotone SUX-10 attached to the port on soundcard 1. He has loaded or created instrument definitions for all three synthesizers, and they are all rolled into one complex **.rgd** file.

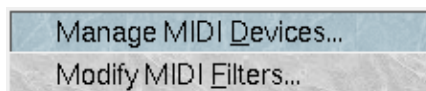


*TIP: These **studios** contain **devices** to represent any hardware that was visible to Rosegarden at the time of their creation. For example, you have three ports available, as in the above scenario. Once you've saved a composition as an **.rg** file, it will thenceforth contain a **studio** similar to the above. If you disable one of your ports, you'll no longer see the **device** attached to it, but it will remain in the file, and will re-appear the next time the port is visible to Rosegarden.*

2.4.4 Loading An Instrument Definition

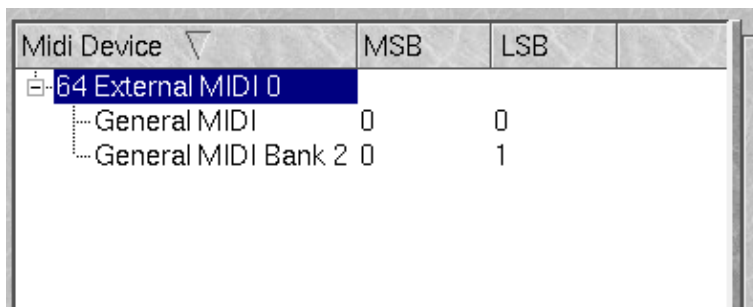
Now that you have enough background to understand what I'm about to show you, let's see about changing the default **studio**.

Again, this is accomplished via **Studio** -> **Manage MIDI Devices...**



To load a **device**, an instrument definition, you'll first want to select a destination port from among those available. Rosegarden can't quite decide what to call these portals through which you transmit notes to something that plays them, and doesn't want to call them "ports." It chooses not to call them anything at all, which can lead to some confusion. There's much discussion on this issue, and I'm hopeful that this will become more clear soon.

In the meantime, you may have something called "External MIDI 0" or something called "Dummy MIDI Device" or anything in between. You'll likely have to resort to trial and error to figure out what is connected to what if you have a complex setup. If you only have one port available, you have no problem, as in this example:

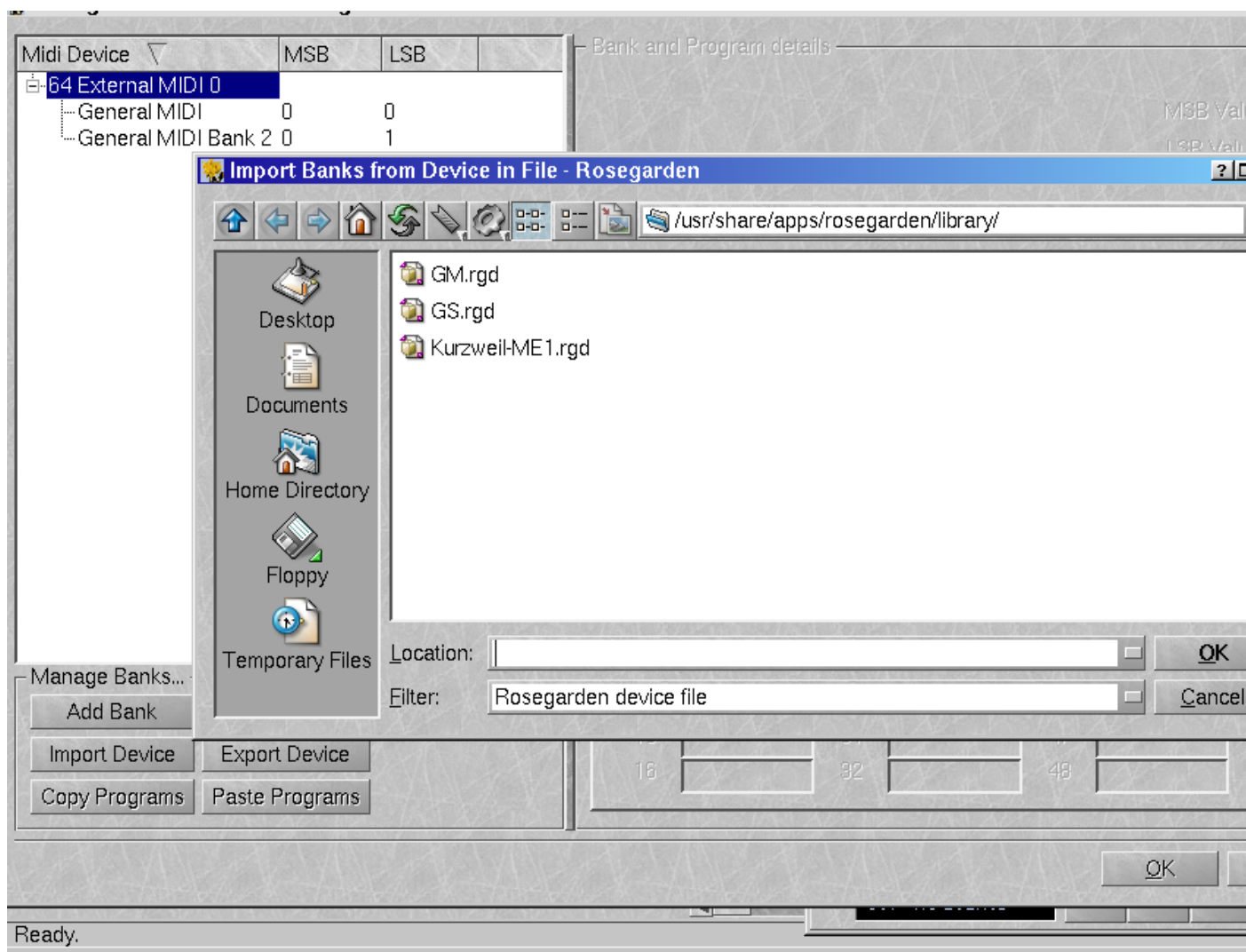


The first level of the tree is the path to the MIDI device, whether it's a physical or "virtual" port. The branches of the tree point to **banks**, each of which can contain instrument definitions for program changes numbered 1–128. MIDI uses two parameters, called **MSB** and **LSB** by Rosegarden, to change banks, and their implementation is device–dependent. Notably, GS and XG have very different ideas about what to do with banks. Rosegarden allows you only one definition per **bank**, each with a unique **MSB/LSB** address.

Now, to load a set of standard **banks** that will define the instruments available for playing via the "External MIDI 0" port, use the **Import Device** button.



You'll be presented with a list of **devices** (.rgd files) from which to choose. By default, you'll be presented with selections from the Rosegarden **library**:



GM is a **device** that contains only the standard General MIDI **bank**. The **GS** device contains multiple **banks** providing an instrument definition for the Roland GS implementation available on the SC-33 (the SC-55's little cousin.) I'm not sure what the Kurtzweil ME-1 **device** contains, but if you have one, I'm sure you'll figure it out. :)

*NOTE: You may also load **devices** from other locations, of course, by browsing around to wherever you've left them in the usual fashion. You may import banks from both **.rg** and **.rgd** files. The **device** definition contained by both types of files is functionally identical, and **.rgd** files are offered only as a logical convenience.*

After you've selected an **.rgd** file from somewhere, you'll be presented with a dialog like this:



The stock **device** files included with Rosegarden are supposed to only contain definitions to be attached to one port, so there should only be one "**Source device**" to load. If you're loading a file that you've created yourself, which contains multiple **device** definitions, then you may select which one of them to load, and to which port to attach it. Currently they are all called "Dummy MIDI Device," which is, quite frankly, horrible, but that will disappear by the next release.

Make a choice, and then you can **Merge Banks** or **Overwrite Banks** to see what your choice was.

If you choose to overwrite, any existing definitions will be lost. If you choose to merge, your existing definitions will remain in place, and any new definitions contained in the **device** you're loading will be merged in.

2.4.5 Creating A Custom Device Definition

If none of the **devices** in the **library** fit, you can use one of them as a starting point for rolling your own, or you can build one from scratch. In order to do that effectively, you need to determine your synth's implementation. Most MIDI synths have a "Tone Table" in the manual somewhere. If you're using a soft synth or some other device that allows random, user-specified instrument assignments, then you'll have to figure out what to call the instruments that are available using various combinations of program and bank changes for yourself.

The first step is to determine how your synth handles bank changes, so you can assign numbers to either the **LSB** or **MSB** parameters. I'm not sure what those names actually mean, but trial and error will get you there. It turns out that Roland GS uses the **MSB** for everything, but your mileage is almost guaranteed to vary.

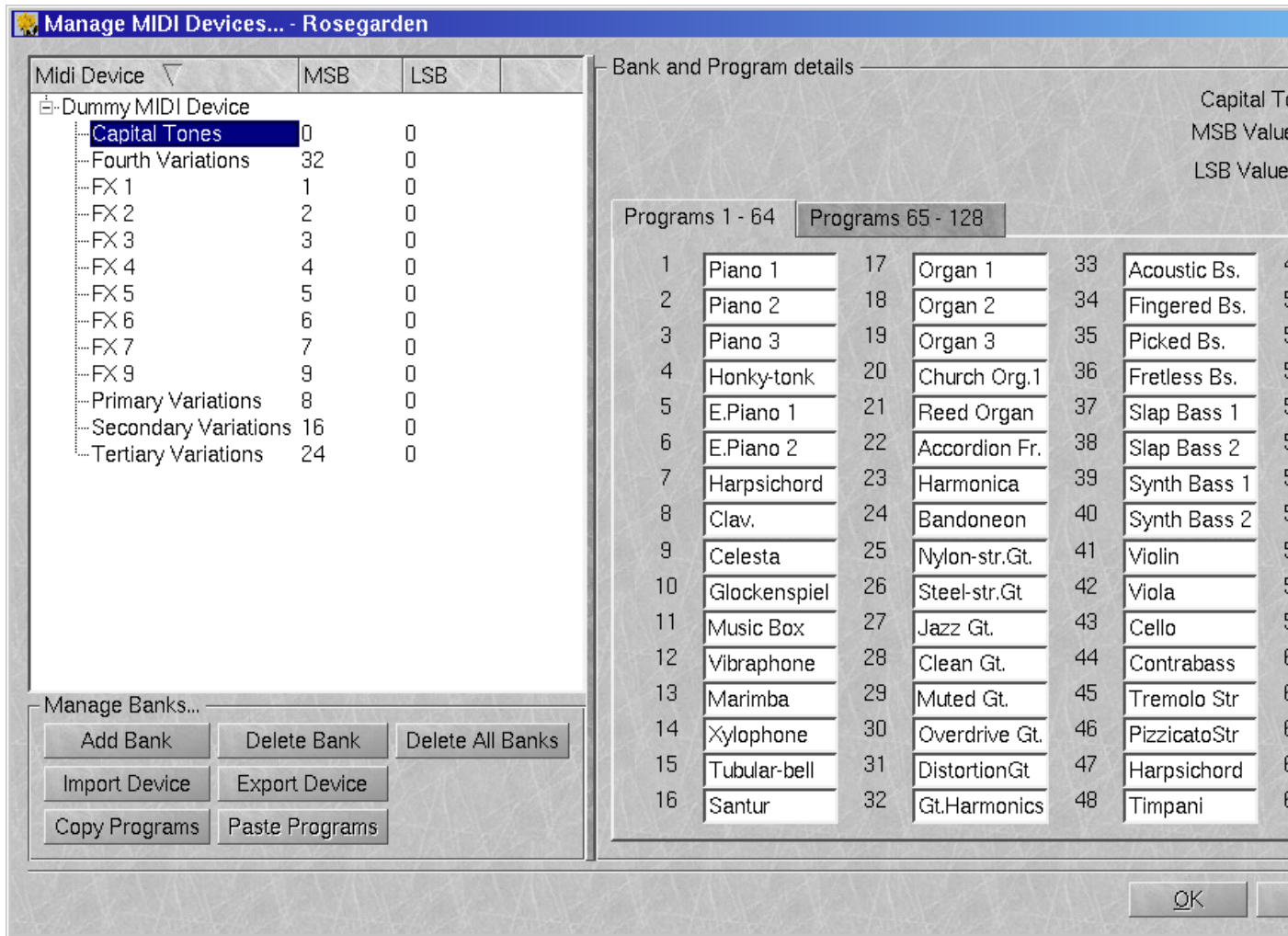
*NOTE: Individual groups of names can only be assigned to a single **LSB/MSB** combination, so this can present problems. When I created the **GS device** definition, I had to compromise. Roland didn't do consistent things with how types of variations were available via particular bank change numbers, and there was no way to keep everything completely logical. Consequently, there are some oddities, such as melodic instruments in bank "FX 1." I named the banks the best way I could figure, using the mechanism Rosegarden provides, but I'm not completely happy with the result.*

Once you've decided what your **device** definition needs to contain, and how it should be organized, entering your data is a straightforward matter. Use the **Add Bank** and **Delete Bank** or **Delete All Banks** buttons as necessary to pave the way.



I don't think you'll have any trouble figuring out what to do. Just fill in the blanks, or delete existing names as needed. To change the name of the bank, click on it to highlight it, then click on it again, and you should get an edit cursor.

*TIP: The **banks** are sorted alphabetically by name, not by **LSB/MSB** number. You may want to bear that in mind when choosing your names, so that you can force them to sort in some reasonable order.*



Once you've defined the names for your program changes, you can save the **device** definition to disk with **Export Device**.

If you think anyone else might benefit from having the **device** definition, you can mail it to me at mmcintyr@swva.net and I will add it to our growing collection of user contributions. There are already several more available than the ones depicted in the old screenshot above, and new contributions are very welcome. We pioneers must do all the work so that future generations can be lazy. Just remember that I had to

do the GS device from scratch too. :)

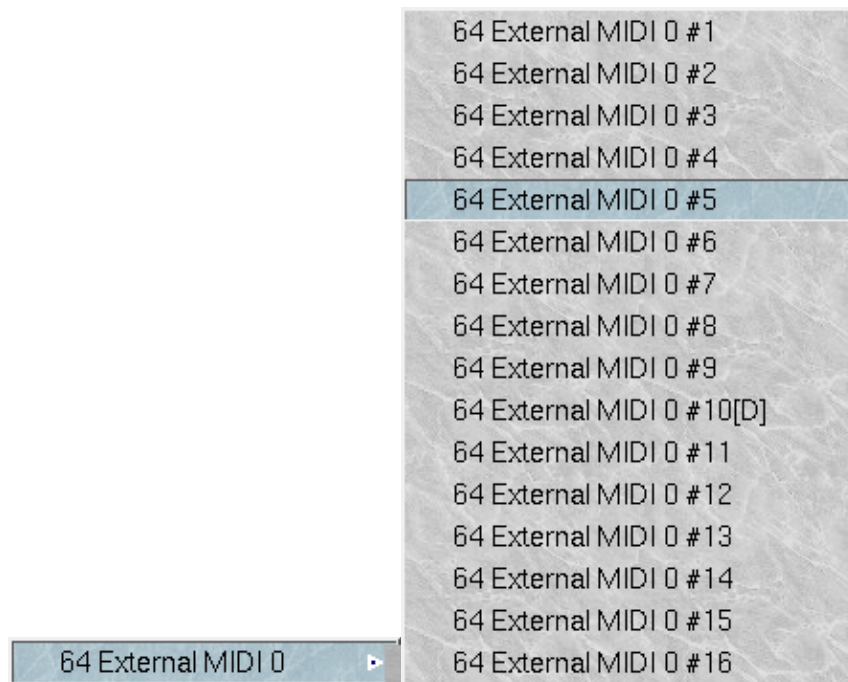
REMEMBER: To avoid having to load instrument definitions again and again, you can get Rosegarden to use your default setup by copying the `.rgd` file you export to `~/.kde/share/apps/rosegarden/autoload.rg`

Now that your instruments are defined, let's look at how to assign them to tracks and channels. This is accomplished in two parts, but we'll look at the **Tracklist** first since we're already in the general vicinity.

2.4.6 Assigning a MIDI Device/Port and Channel

Click on the `<untitled>` in one of the tracks, and you'll be presented with a series of nested menus that allow you to choose the MIDI device/port and channel for the track. If a channel has already had an instrument assignment on another track, the instrument name will appear in parentheses next to the channel number. If you see only numbers, don't worry, as I'll explain the other half of this process in the next chapter.

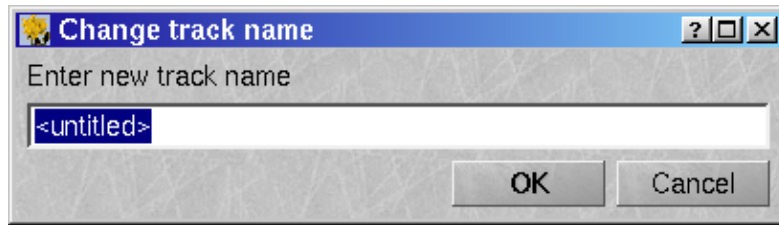
Your menus will almost certainly differ, but this is what it looks like here with default **studio** settings, without my soft synth running:



NOTE: Observe that even with no instruments assigned to any channels, a (D) appears beside channel 10. This channel is used for percussion in most MIDI implementations. It is not currently possible to define names for percussive instruments, nor to define names for drum kits for synthesizers that make use of bank changes in order to switch kits (eg. Roland GS). These items have been on the table, but are currently shelved. I'll explain how to work around this limitation later on.

2.4.7 Changing the Track Name

I'll go ahead and mention this now, while we're already looking at the **Tracklist**. You can change track names by double-clicking on the label. A dialog should thus appear:

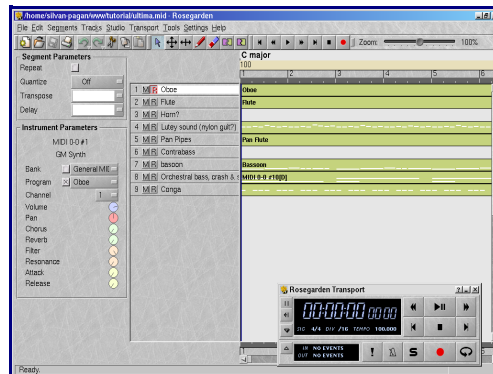


3 Project 1: Importing a MIDI file

3.1 Importing the File

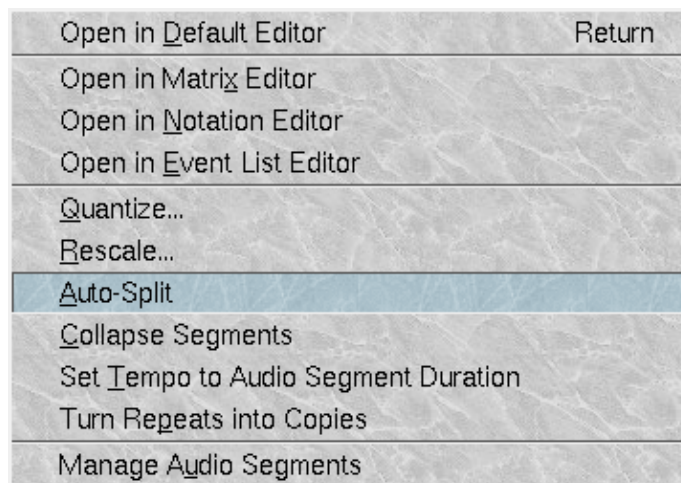
It's time to import our first [midi file](#). Use the **File** → **Import** → **Import MIDI File...** menu to navigate to this [file](#), and then open it.

Once imported, your screen should resemble this. If it doesn't, take another look at the **Settings** menu and toggle all the optional displays on:

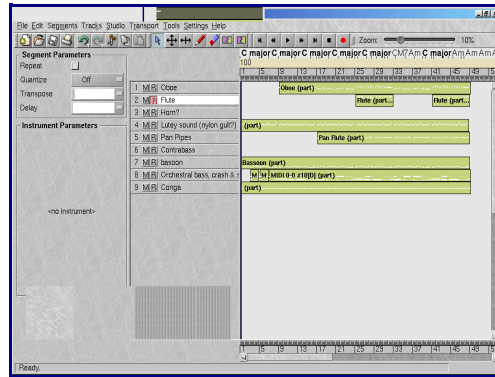


While we're at it, let's go ahead and split these huge segments into smaller pieces.

Use **Edit** → **Select All Segments** from the menu to highlight all of the segments. Then use **Segments** → **Auto-Split** to automatically split them, eliminating sections of silence, and leaving you with little islands of notey bits.



To get a better overview of the composition, use the **Zoom** slider to zoom out a bit. Your screen should then resemble this:



Now that we've imported the file, let's see how it fared. Rosegarden is not purely a MIDI sequencer, and a certain amount of translation takes place when importing MIDI files into Rosegarden's internal format. Configuring your **device** definition in the **Studio** beforehand goes a long way toward eliminating trouble, but sometimes the import process introduces minor problems that need to be corrected, which we will address directly.

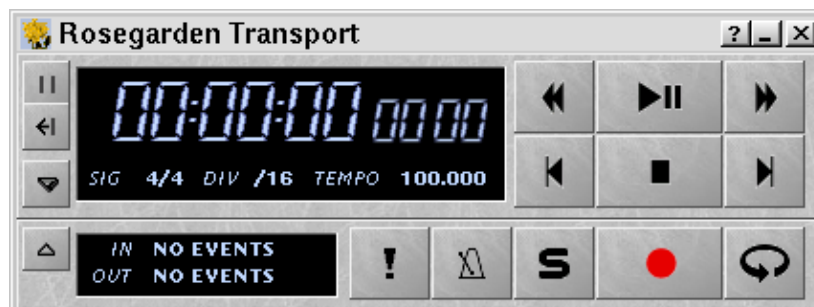
3.2 Transport Tools

First, we need to see what it sounds like. Let's get to know the **Transport** window, and the **Playback Pointer**, which are two of the tools we'll use to get around in our compositions.


Notice that there's a vertical blue bar in the **Segment Pane** of the main window. That is the **Playback Pointer**. You can move it to any arbitrary point in the composition by clicking along the dark gray portion of the **Ruler** at the top, or you can use the **Transport** to move it around. If you don't see it, play with the **Transport** or the **Ruler** until you move it out of hiding.

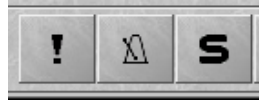


You should have a separate, floating **Transport** window like this one.



Most of these should be self-explanatory. If you can't figure out what any of the buttons do, hover over them a bit, and context-sensitive help will pop up.

Of special note are the **Panic**, **Metronome**, and **Solo** buttons. If they're missing, toggle the "flap" via the  button.



The **Panic** button transmits an "all notes off" message. Use it, for example, when your synth has gotten a note stuck and the incessant droning of that low F# is driving you up the wall.

The **Metronome** does what you'd probably expect. You can turn the metronome on during normal playback, as well as recording, and it is a useful tool to use when practicing playing along with your composition.

The **Solo** button mutes all tracks except for the one that's currently selected, allowing you a convenient way to hear just one part. (To "solo" more than one part, at present, you'll need to set the mute for tracks individually. The **Solo** button only affects a single track.)

3.3 Tweaking the Import

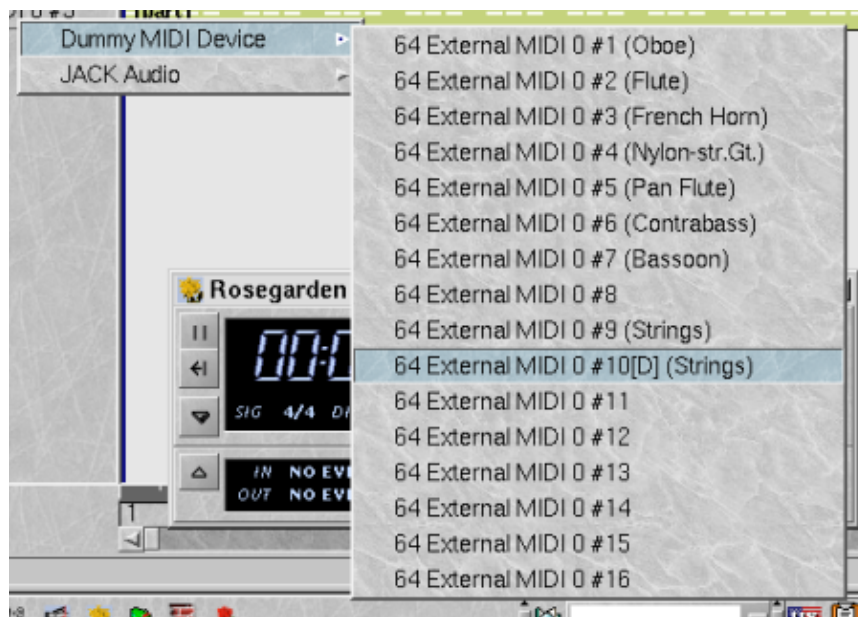
Find the **Play/Pause** button and give this composition a listen. It's very likely that it will sound somewhat lame initially. That's the result of a few issues that Rosegarden needs a bit of help to work around.

This file was created with Cakewalk, and it includes a SysEx bank that would have set up the reverb level and various other parameters on my synth. That has been lost. It includes percussion on two separate channels, which is why you're hearing those raunchy piano notes in the bass (that track was assigned to channel 9), and it's supposed to use the "OrcheSet" drum kit on my synth. The program change on channel 10 was lost.

So, before you write me off as a terrible composer, let's get this back to the condition it was in when it left Cakewalk last year. We need to correct for what has been lost in the translation from raw MIDI to Rosegarden's internal format, and then we'll play with it.

The first thing we really need to do is fix the percussion. The most direct way to accomplish this is to assign the "Conga" track to the same device and channel as the previous track.

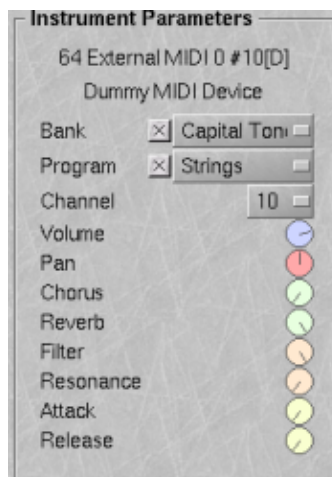
Click on the "Conga" track name, and hold the button down for a moment. Navigate through the various available devices until you find channel 10. Your results will vary depending on a number of variables, but here's what it looks like here: *(Note that the "Dummy MIDI Device" will hopefully change into something more useful soon.)*



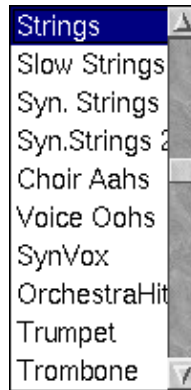
Now, we really need to change the drum kit. As I mentioned in the **Studio** chapter, Rosegarden can't yet cope with having GS-style named drumkits on channel 10. It's on the table, but no decision has yet been made as to exactly how best to implement it. Until then, we'll have to live with having an instrument listed instead of our drumkit.

I jumped the gun in taking the above snapshot. Note that I have already assigned the "Strings" program to this channel. It so happens that "Strings" corresponds to the correct program number I need to use in order to switch my Sound Canvas to the "OrcheSet" drum kit.

In order for you to do the same at home, click on either the "Conga" or the "Orchestral bass, crash & snare" track name to select it. (Remember that they're both bound to the same device and channel, so a change made to one will affect the other.) Now click on the **Program** button in the **Instrument Parameters** pane.



If the **Program** checkbox isn't checked, check it. Now you can scroll through with your mouse wheel, or select a name from the drop-down list. It's helpful if your synth has a display so that you can see when you happen to get to the correct program number. In this case, "Strings" in the "Capital Tones" bank is what we're after.



Of course your mileage with all of this will surely vary depending on your synth. You may not have the option of changing this at all, or may need to use a completely different method for changing it. XG users, I've learned, will need to use bank, rather than program changes, to change drum kits. Soft synth users and others who can make use of them will probably want to load an appropriate soundfont. Something medievalish would be tasty. Season to to suit yourself.

Now that the drums are a little more interesting, it's starting to sound better, but there's still work to do. We need to make up for that lost SysEx bank by cranking up the reverb.

3.4 Instrument Parameters

The SysEx bank that Cakewalk included with this file has been lost in the translation. Luckily, Rosegarden makes it easy to set the initial values for several common controllers right there in the **Instrument Parameters** pane. If you didn't notice this before, take note of the knobs in this pane.



Let's select each track in turn, and dial up the 'verb. Click on the **Reverb** knob and either use your mouse wheel or else drag the cursor in a straight horizontal or vertical line (don't try to drag it around in a circle) to change the position of the knob. Crank it up. You can never have too much reverb. Now lather, rinse, repeat until every track has major 'verbage. (Or reverbificate to taste, if you insist on having less than maximal 'verbage...)

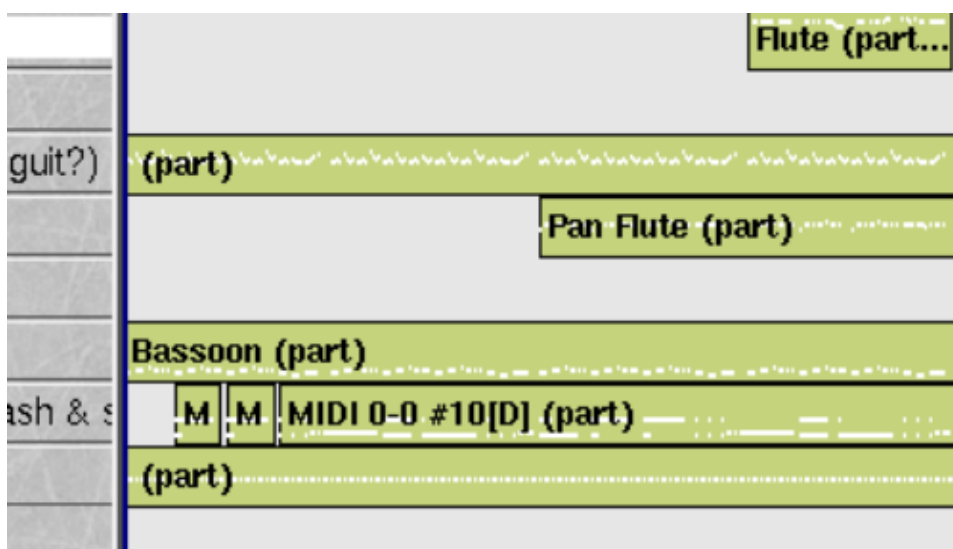
Depending on your synth's implementation, some of these knobs may not actually be useful to you. If any of these knobs don't seem to have any effect, blame it on your synth.

3.5 More About Segments

You've seen what segments can do, but there are a few things that I couldn't demonstrate while the segments were empty.

3.5.1 Segment Previews

Look closely, and you'll notice that each one of the segments contains a miniature "matrix" or "piano roll" view. You can get some sense of what that bit of music is like just by looking at the segment display, which is a useful visual aid when dragging things around. If this feature is not turned on, turn it on with **Settings** → **Segment Previews**. Later, we'll see that we can also manipulate audio segments in similar fashion.



Now that the drum kit has been corrected and both drum parts are playing on the correct channel, it's starting to shape up. Now is a good time to mention the autosave feature.

3.6 Autosave (CVS)

As always, "save early, save often" is the mantra of computer users everywhere. It's never a bad idea to take time out to save your work, but Rosegarden tries to save you from yourself.

*I will only mention it in passing for the moment, because this feature doesn't officially exist yet. You'll **only** find it in CVS.*

If the GUI crashes for some reason, or you have a power outage or the like, the next time you start Rosegarden and attempt to load the file you were working on, you'll be presented with the option of loading the autosave file instead. If you hadn't yet saved the file, you'll find it in your ~/autosave directory with a basic name of "untitled."

*The autosave interval defaults to 60 seconds. Configure it via the **Settings** menu.*

This feature alone is probably worth dipping your toe into the waters of CVS if you're not already building it, because, let's face it folks, this is pre-release software, and it crashes from time to time.

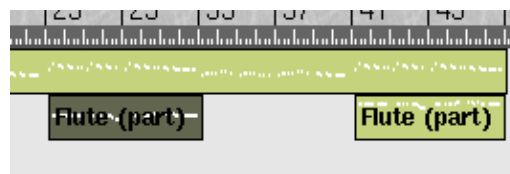
4 Let's Edit Some Notation

For the next phase in our sample project, we want to look at what is necessary to make the flute part in this composition playable on a real flute, and then print a score. In this chapter, we'll make some basic changes to the part and work on cleaning up the humanized performance data so that it renders as more tidy notes. Later on, in Chapter 5, we'll add some new material to this part to make it more exciting to play.

4.1 Opening the Notation Editor

Let's get a look at what this part looks like in notation. It will probably be ugly, since I recorded it on my keyboard instead of entering it note by note.

Let's open the **Notation Editor** and have a look. Highlight a segment from the "Flute" track.



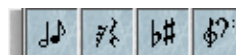
Open the **Notation Editor** by right-clicking on the segment, and choosing "Edit as Notation" from the context submenu. You can also configure this to be your default view via **Settings -> Configure Rosegarden... -> Default Editor**, and thereafter you can get to the **Notation Editor** by merely double-clicking the desired segment.

4.2 Notation Editor Overview

We'll be spending a lot of time here in the **Notation Editor**. As you'd expect, this is where you go to view and edit your data as standard notation. This editor has many more features than I will make use of for my purposes here, but this should be enough of an introduction for you to get wherever you want to be with your own projects.

The first thing you'll notice is that there are different toolbars here. Before looking, let's synchronize our displays. You may not yet have a **Transport Toolbar** for this window. Check **Settings -> Show Transport Toolbar** and the **Transport Toolbar** should appear. It would be a good idea to use **Settings -> Save Settings** at this point so that it remains there, as it's very useful.

Now, let's have a look at the new toolbars. You can toggle several of these via the **Toggle Toolbar**

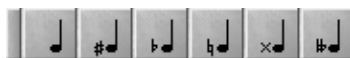


The **Cursor Toolbar** has three cursors, **Erase**, **Select**, and **Text**.



The **Note Duration Toolbar** has both regular and dotted flavors of several note lengths, plus toggles for **Chord Insert Mode** and **Triplet Insert Mode**, which should be self-explanatory. With the **Select** cursor highlighted, you can click on a duration and then click on the staff to insert a note.

You can toggle accidentals via the **Accidentals Toolbar**,



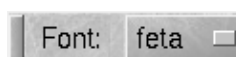
or you can add rests via the **Rests Toolbar...**



Add clefs via the **Clefs Toolbar**,



and you can change the font used to render the notes via this little gadget here:



The "RG 2.1" font isn't as pretty, but it renders more quickly than feta, so may be useful on slower hardware.

Use **ctrl-Z** or **Edit -> Undo** to remove the results of your having played with these things.

4.2.1 Controlling Playback

The **Transport Toolbar** that we turned on in the preceding section is the same as in every other window, with two notable exceptions, which is why I've saved it until now. These extra buttons are useful for controlling playback when the **Playback Pointer** is outside the segment you're editing, and you want to hear what you're working on.



If you haven't moved it, the **Playback Pointer** is at or near the beginning of the composition. This part doesn't come in until much later, so if we hit the **Play** button now, we can't expect to hear anything for awhile.


The **Notation Editor** has its own cursor, which is just like the one in the **Segment Pane** of the **Main Window**, except that this one is magenta. In order to control playback, you need to be able to manipulate both cursors, and those two extra buttons on the end of the toolbar help you do just that.

Let's move the playback cursor to the beginning of this segment. Click on the ruler above the staff to position the local cursor, just as the ruler in the **Main Window** positions the **Playback**

Pointer itself. Click the **Playback Pointer to Cursor**  button to move the playback pointer to the

position of the cursor. Now playback will begin from that point.

You can set it back "out of sight" so that playback will begin before the start of the segment

you're editing by shifting it to the left via the  button.

4.2.2 Changing the View

Notice the **Size**, **Spacing**, and **Smoothing** sliders. (They're camera shy for some reason, and come out black. I'm sure you'll have no problem figuring out what I'm talking about.)

The **Size** slider changes the size of the font used to render the notes.

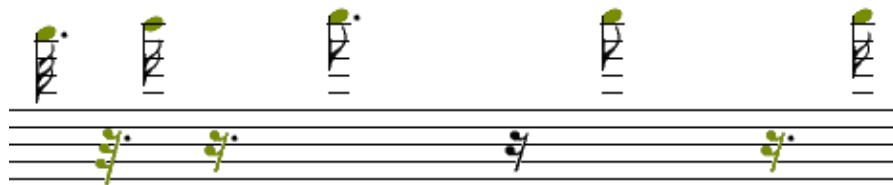
The **Spacing** slider can be used to adjust the horizontal spacing of the notes, to draw them closer or further apart. Useful for spreading out particularly complicated rhythms for easier reading.

The **Smoothing** slider is useful for cleaning up performance data somewhat, to make it more legible.

All of these controls are also available from the **View** menu.

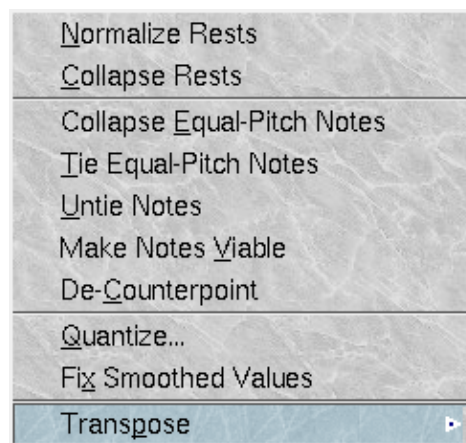
4.3 Transposing

Now that we've had our bit of fun playing with all the new toys, let's look at the part and see what it's going to need.



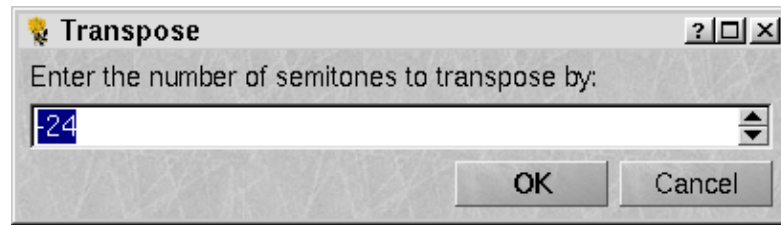
Gack! This definitely should have been a piccolo part. My head hurts just looking at this stuff. Let's transpose it down an octave and see what it looks like. Let's make that **two** octaves! I can get that high with my flute, but my dog would hate me. Two octaves are good.

Select **Edit** -> **Select Whole Staff**, then go to the **Transforms** menu.



Lots of good stuff to play with here, take note, but we want **Transpose...** Since we're going to go down two octaves, we might as well go for the **Transpose...** option at the bottom. Use the arrows to dial down, or else enter -24 into the input box. Positive numbers mean an upward transposition, and negative numbers will

transpose downward.



4.4 De-Humanizing

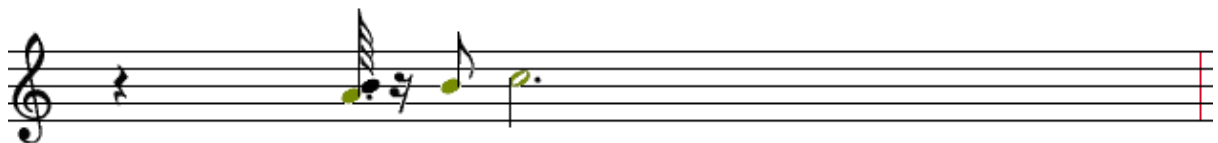
Now that we're down to a reasonable register, let's do something about the ugly notes. The problem with human performance data is that it doesn't scale well into notation. You'll notice that some of the notes are displaying in yellow. That's because they're notes whose timing and duration don't quite make sense to Rosegarden within the context of this measure, so it has colored them to let you know that the notes on the page are only an approximation of what will actually be played. To see a view of the data in a more pure format, use the **Matrix** or **Event List** editors, which we'll take a look at later on.

Since our objective here is to make the notation more readable, we might as well remain here and alter the data until it better fits into place within the context of notation. In this instance, readability is more important to us than the faithful rendering of a human performance, so we're going to end up making "physical" changes to the data along the way.

4.4.1 Smoothing

Let's start by seeing what that **Smoothing** slider can do for us... Each notch makes the rhythm progressively less obnoxious to read, leaving the notes in place. Adjust the slider, and see what results. I found that the final notch was too high, and the next lower notch was looking better, but wasn't really quite there.

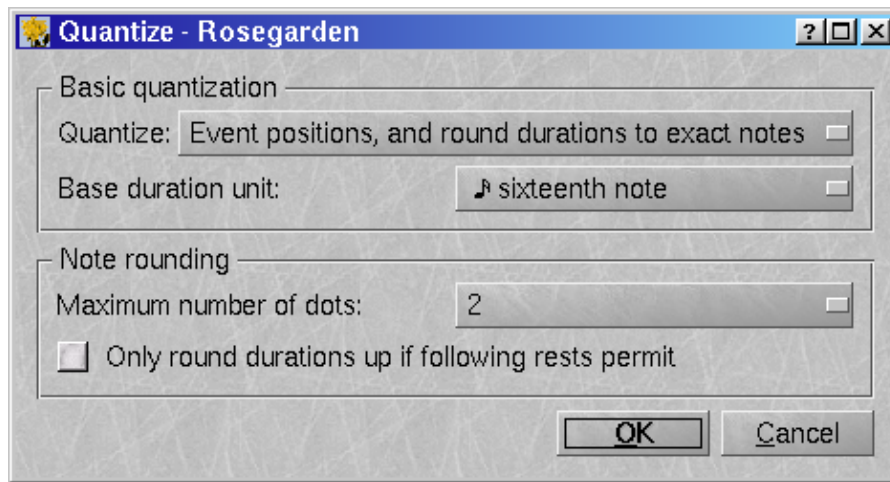
Let's try making these smoothed values permanent. Select the whole staff, and let's try **Transform** -> **Fix Smoothed Values**...



Well, that didn't quite have the desired effect, did it? The timings are still too far off. Remember **ctrl-Z** is a shortcut for **Edit** -> **Undo**.

4.4.2 Quantize

I think this data is so messy that we're going to have to quantize it. Let's **Undo** that last change, and play with **Transform** -> **Quantize** instead. Let's tell it to **Quantize: Event positions, and round durations to exact notes**. We'll leave the **Base duration unit** at 16th, and accept the default **Note rounding** values and see where that gets us:



Not too bad. It's looking much better:



Let's see how those changes sound, and then we'll see about turning those 16th note/rest pairs into something a little easier to read...


The quantized notes seem to have fallen into place along sane boundaries without having an undue impact on the rhythm or feel of the piece. Bearing that in mind, let's back up a step and see what would have happened if we had quantized on 8th note boundaries. Hit **ctrl-Z** or use **Edit** → **Undo**, then run through the **Transformations** → **Quantize** dialog again, specifying a base duration of 8th note this time around.

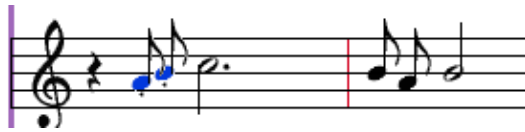
Very nice indeed, and Rosegarden has rendered some 16th notes and other odd bits as required, ignoring the base duration when it really had to. That's one of the best jobs of quantization I've seen in a notation editor. There won't be much clean-up to do from here:



4.4.3 Adding Marks

We've changed the flavor somewhat by lengthening those notes. Let's make them staccato. There isn't currently an icon for this, but you'll find it on the menu. First you'll need to make a selection.

If it's not already the default, choose the  cursor from the **Toolbar**. Click at a point to the left of one of the notes to begin the selection. Drag the blue selection box so that it includes the first two 8th notes. When they're selected, they'll turn blue. Now use **Marks** → **Add Staccato**. The dots appear:



4.4.4 Auto-Beam

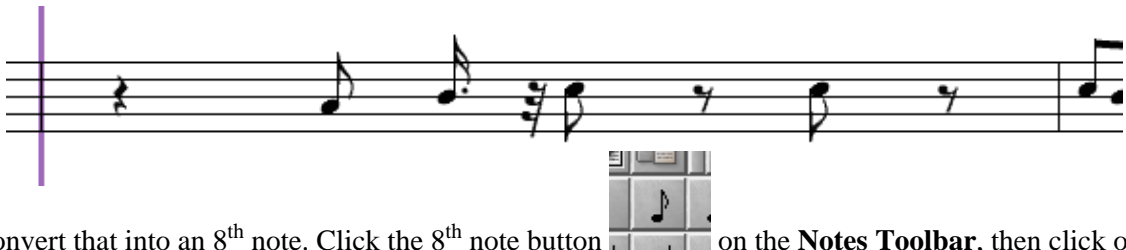
Now let's add beams. **Edit** -> **Select Whole Staff**, then **Group** -> **Auto Beam** to beam these notes into logical groupings:



4.4.5 Changing Note Durations

One measure further into the segment, there are some rhythms that could probably be written so that they're less awkward to count. Let's see about changing them.

There's an odd dotted 16th with 32nd rest combination that should probably be another staccato 8th note:



Let's convert that into an 8th note. Click the 8th note button  on the **Notes Toolbar**, then click on the

offending dotted 16th note's head. It changes into an 8th note, and the rest disappears. *Note that Rosegarden pads out the rests as necessary, to keep the correct time in the measure.*

4.4.6 Let's Add Some Slurs

Let's add some slurs as well. Highlight the first two 8th notes in this measure, then use **Group** -> **Add Slur**. You should be looking at something very like this:



4.4.7 Manual Beaming

Now let's do two more things to enhance readability. I'm used to reading beamed groups that count out to one beat. Let's separate those two groups, and add another slur. Select the first two 8th notes in that four-note

group, then use **Group** → **Beam Group** to fuse those two together. The remaining two notes adapt automatically, and remain beamed. Now select all four of the 8th notes plus the half note and add a slur. Finally, notice that the note stems are in the way of the slur. Let's select the second pair of 8th notes and use **Notes** → **Stem Direction** → **Stems Down** or else **ctrl-PageDown** to change their direction. You should wind up with a measure resembling this one:



Continue in this fashion until you finish altering the notation so that it's nice and tidy. I think you should have a pretty good idea how to proceed from here, so I'm not going to walk you through the rest of this job.

4.4.8 The Text Cursor

Now let's add some directions. Exchange your **Select** cursor for a **Text**  cursor. Click toward the

beginning of the staff, near the clef. Let's start by adding a tempo and a dynamic of ff, (since my dynamic range is from ff to ffff :). Click the cursor where you want the text to appear, then configure it via the dialog.



It's pretty self-explanatory. My score ended up looking like:



Andante



*NOTE: Did you notice that one of the available text types was called "annotation?" You can use these to make notes on your score, and can control whether or not you see them via the **Settings** -> **Show Annotations** toggle.*

4.5 A Change of Key

(unfinished... sorry...)

This section will talk about ways of transposing a part to a new key, and then about how to write a part for a transposing instrument (eg. a B_b trumpet) so that it sounds in one key and is written in another. I just haven't had time to sit down and write it. Writing documentation to this standard is a very time consuming process. I hope you agree that it's worth the time, and worth the wait.

4.5.1 Multiple Staves

Until now, we've only edited the one part. The **Notation Editor** allows you to edit any number of segments simultaneously. Since we're going to transpose the entire composition, let's use **Edit** -> **Select All Segments** and then **Segments** -> **Open in Notation Editor**.

(unfinished... sorry...)

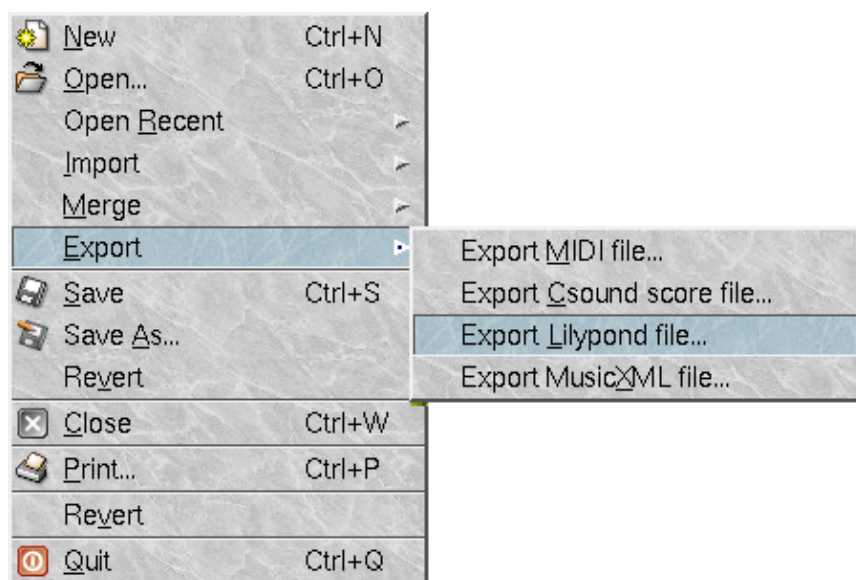
Quick note: you can select multiple segments and edit them in one window. That's what I'll show you here eventually, explaining it in my usual level of detail, by running through and grabbing screenshots as I go, to make sure I get it right.

4.6 Printing

The current design plan calls for a rudimentary print facility incorporated within Rosegarden itself, whose purpose is to allow rough draft printing, combined with off-loading the task onto an external utility for high quality printing. The print engine in CVS works pretty well for draft printing. The print engine in 0.8.5 is totally horrible. If you want to print notation, dump 0.8.5 and get CVS.

The "external utility" of choice is Lilypond, and we've put a lot of effort into making Rosegarden a useful front end for Lilypond. We also support music XML, but the active development is on Lilypond export.

If you have a look at the **File** -> **Export...** menu, you'll see that it's possible to export your compositions in both Lilypond and MusicXML formats:



I've spent many hours on the Lilypond export code myself, trying to make it work as well as possible. I've added a lot to what Hans originally provided, and there have been many improvements. Notably, I've taken advantage of Rosegarden's new "metadata" facility to provide mechanism for exporting user-specified headers. I'm very grateful to Chris and Guillaume for providing some way for me to be able to export these, and for you to enter them. This really deserves a proper explanation, complete with visuals, but I don't have time to do it right. For now, a brief description:

Look at **Edit** -> **Edit Document Properties**, then look at the **Description** tab. Put the header name (eg. "title") in the **Name** field, and the string you want attached to that header in the **Value** field. All 14 possible headers are exported now, but the interface is still a bit weird.

In spite of all this effort, however, Lilypond export is an imperfect thing. Rosegarden and Lilypond have different ideas about how to draw some things, and there's no getting around the need to manually tweak your Lilypond files for the best possible results, or at least to feed the export a set of things that it can translate well.

With careful planning, good results are possible, but if you attach five marks to every note and have all sorts of performance directions and whatnot in the same space, Lilypond will jumble it up. There just isn't any reasonable mechanism for getting Lilypond to render things as far above or below the staff as Rosegarden does, and there's only so much an export filter can do.

If there's something you don't like about the Lilypond export, please let me know. Hans is still the official "owner" of the code, but I'm currently the one actively working on it. There are still some bugs and missing features, and I'm eager to get user feedback; particularly to hear whether you like the way I've implemented the new features, and what you'd like done differently. I want to make this as useful to you as it can be, and I have both the time and willingness to devote a lot of effort to incremental improvements.

Recent improvements in CVS:

- ◆ all accidentals and enharmonic spellings handled correctly
- ◆ text events (Tempo, Local Tempo, dynamics, etc.) exported as well as practicable.
- ◆ marks exported as well as practicable
- ◆ user-specified Lilypond headers available

It doesn't sound like much, but I have hundreds of hours invested in making those improvements. I hope you like them. :)

5 Using Audio

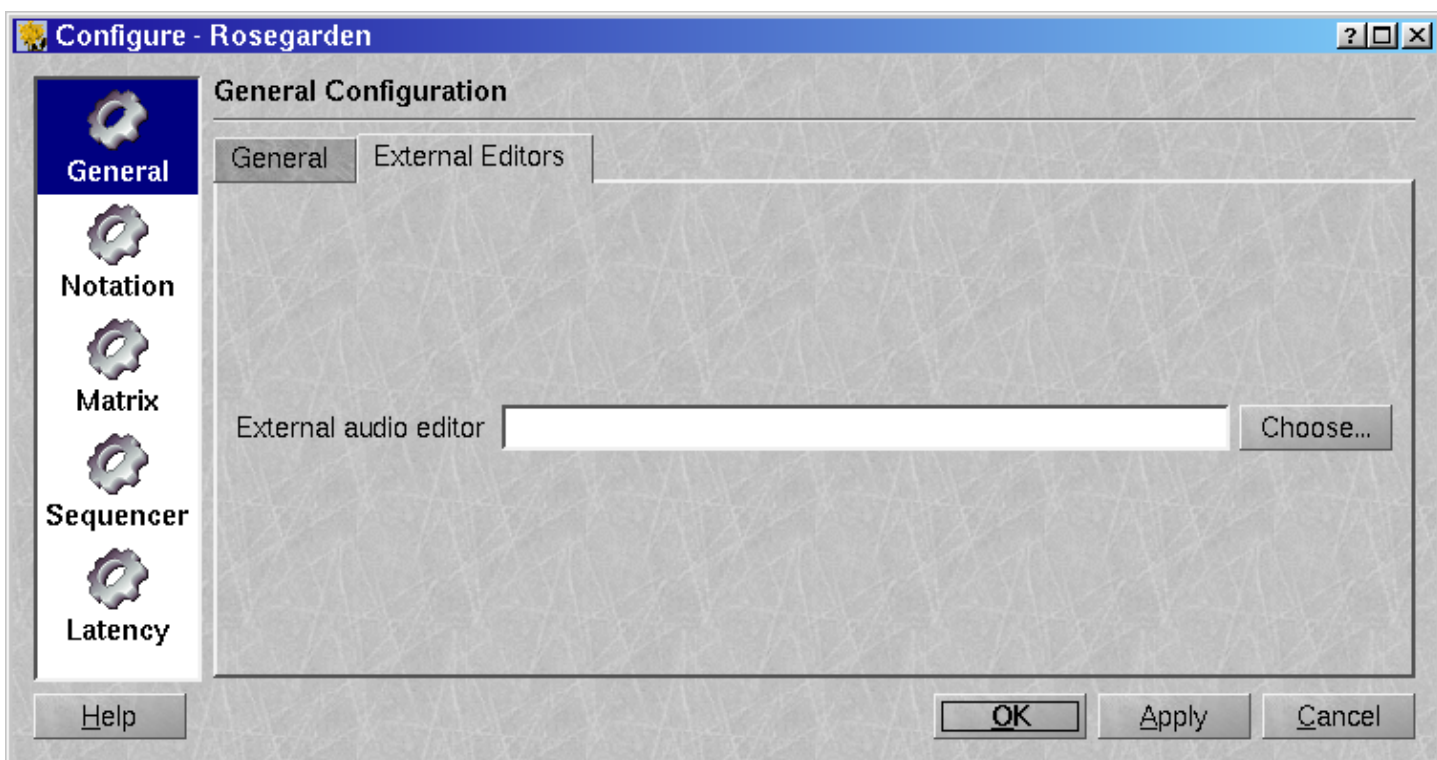
Those of you with a keen interest in matters audio have probably been wondering why it took me so long to get to this. The truth is, I've just never gotten around to trying to do anything serious in this area, and I haven't really done much with these features. As I said in the introduction, I am not an audio guy.

At present, I can't offer the same level of detail as in the rest of this guide, because what I say here is not based on real experience. Still, there is great demand for information on this subject, so I will do what I can to show you how to make use of the features. *(If I'm leaving something important out, or my explanation is really off, feel free to let me know.)*

We'll start by loading the composition we worked on in Project 1, then add some audio to it and look at how to manipulate it.

5.1 Configuring an Audio Editor

Rosegarden has no built-in facility for editing audio. Instead, it provides a mechanism for you to configure an external editor, via **Settings** → **Configure Rosegarden** on the **External Editors** tab:

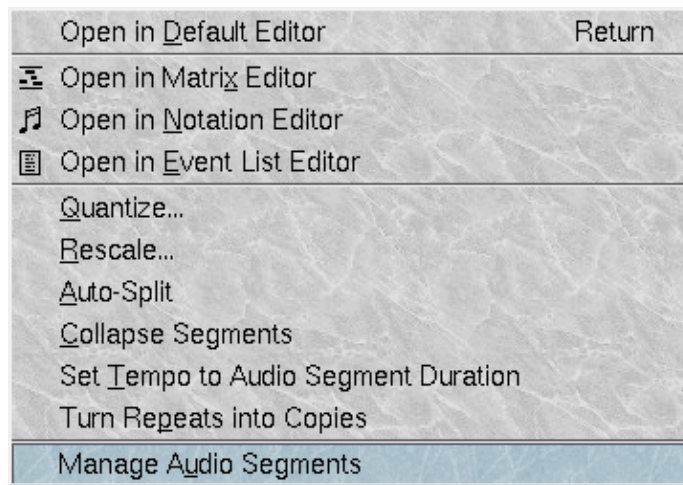


You can browse to your editor of choice via the **Choose...** button, or type the name of the executable. My personal preference is for Audacity (<http://audacity.sourceforge.net/>), which I really like, but there are many suitable editors available.

5.2 Importing Existing Audio

There are two ways to get audio into Rosegarden. If you wish to import projects that you have been working

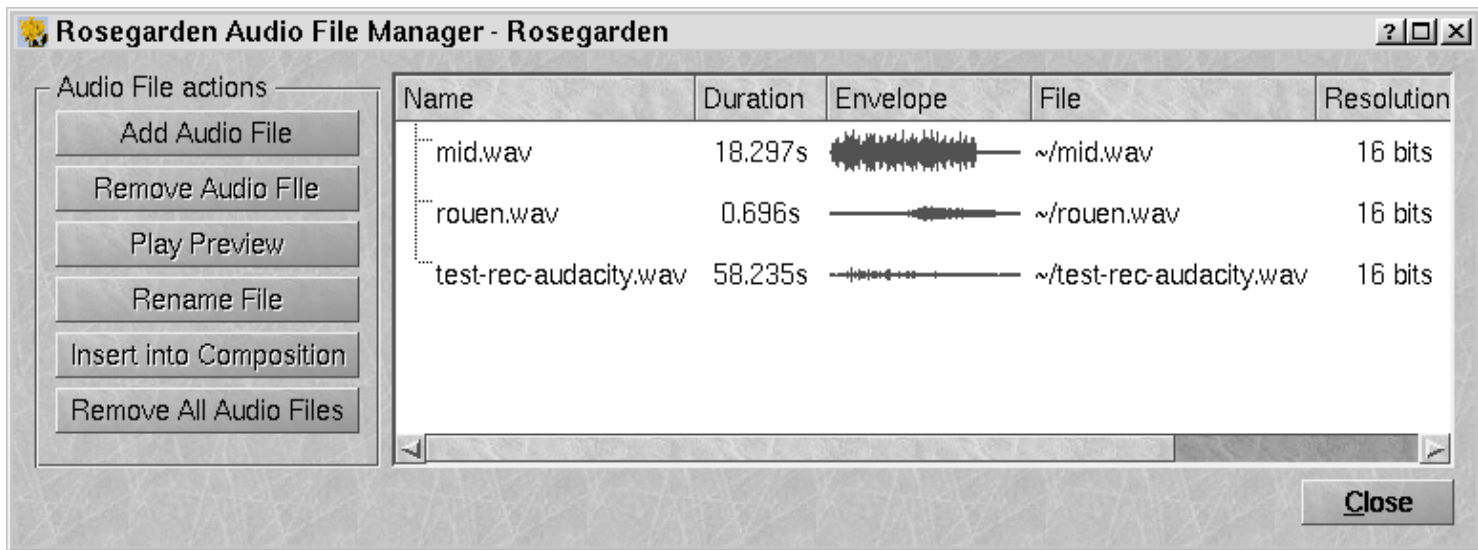
on with some other application, you can import the audio files and insert them into your composition. The heart of this facility is the **Segments** -> **Manage Audio Segments...** menu, tucked away at the bottom, below the divider:



This takes you to the dialog you use to select audio files for insertion into your compositions:



Using this is fairly straightforward. Use the **Add Audio File** button to pick a file, which will then appear in the list. I'll just grab some .wav files that I have handy, since I haven't actually recorded anything related to this composition:



Once the files are loaded, some statistics are printed, and a brief preview appears. You can use the **Play Preview** button to hear what you've got.

*NOTE: If the **Play Preview** button is grayed out, as in the above example, it probably means that either Rosegarden wasn't compiled with audio features enabled, or else your audio server is not running. In my case, I forgot to run Jack before starting Rosegarden.*

Once you're satisfied that you've selected the correct files, you can insert them into your composition one by one with the **Insert into Composition** button. Let's insert "rouen.wav" and make both Dave Barry and Guillaume Laurent proud:

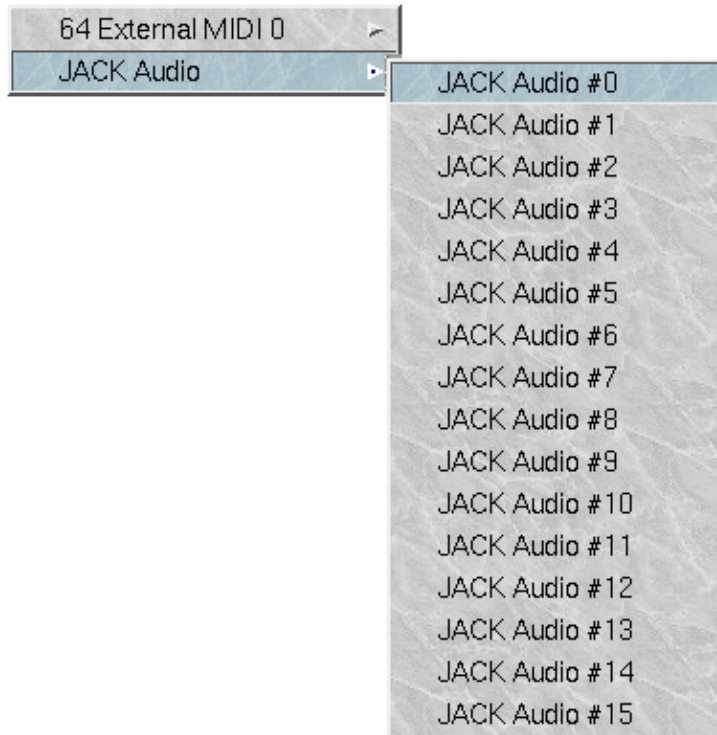


Newly-inserted audio segments go into a new track of their own, and it's always appended to the bottom of the **Tracklist**.

*TIP: If you're working on a newly-created composition, rather than one that is based on a MIDI import, it will start with a large number of tracks. It may appear that nothing happened after you hit the **Insert into Composition...** button. Scroll down to the bottom, and you should see a new track with your audio segment inserted.*

5.3 Recording

It is possible to record directly into Rosegarden. First you'll want to point the track to a Jack source in the same fashion used to assign tracks to MIDI devices.



NOTE: I honestly have no idea what JACK Audio #0 through #15 mean.

Now if the **R** button is toggled for the track, making it the destination for record operations, any signal on the mixer channel you've selected as the record source (via KMix or the like) should be recorded, and should wind up in an audio segment.

NOTE: The one time I tried that, it didn't work very well, and I haven't tried to do it since. I'm not sure whether or not this is actually supposed to work. I need to play with that, but at this writing I am quickly running out of time, and want to get this out the door. I don't currently have Jack working, and don't have time to figure out why it's broken.

5.4 Manipulating Audio Segments

Audio segments behave in the same fashion as segments containing notation data. You can drag them around, drag them from track to track, and so on. You can split them, move bits of them around. I don't have time to show you any of that at this writing. I need to go to work, and I want to get this published before I go.

I plan to:

check on recording

find out what jack #0 15 mean

explain how to use Edit Audio and Auto-Split Audio

do a real project with a mixture of MIDI and audio

demonstrate the use of LADSPA plugins

???

6 That's All, Folks...

That's all I've had time to do to date. There are many more topics I wish to treat eventually, but there are so many developments in the works at the time of this writing that I'm just as well off to hold back, to reduce the amount of material I will have to adapt to suit changing future realities. I've already written about far too many features that are only available in CVS as it is.

I hope what you've seen has given you ideas for projects that **you** can do with Rosegarden. If I've left something that's very important to you out, please feel free to send me a note at dmmcintyr@users.sourceforge.net and I will try to accommodate your need and discuss it sooner.

D. Michael McIntyre