

## Rosegarden Codicil User Guide

This document provides instructions on using the Rosegarden Codicil. For a thorough tutorial covering all aspects of Rosegarden please visit this site:

<http://rosegarden.sourceforge.net/tutorial/en/chapter-0.html>

The Rosegarden Codicil is still in early development. Please feel free to contact the project email address [codicil@n-ism.org](mailto:codicil@n-ism.org) if you experience problems using the software. Feedback from users is essential to ensure the product is reliable and user-friendly.

### Compiling the Rosegarden Codicil

In addition to the standard package dependencies, the Rosegarden Codicil requires the FFTW3 library and development package. Installation can be done by executing the following commands:

```
./configure  
make  
make install (as root)
```

### Extensions and Limitations

The Codicil has added new functionality to Rosegarden, but there are also certain limits due to the developmental stage of the project.

The only part of Rosegarden which supports microtonal tunings is the pitch tracker. The addition of quarter-tone accidentals only affects the graphical representation of notes in a notation window and the pitch tracker (as long as the accidentals occur in the tuning definition).

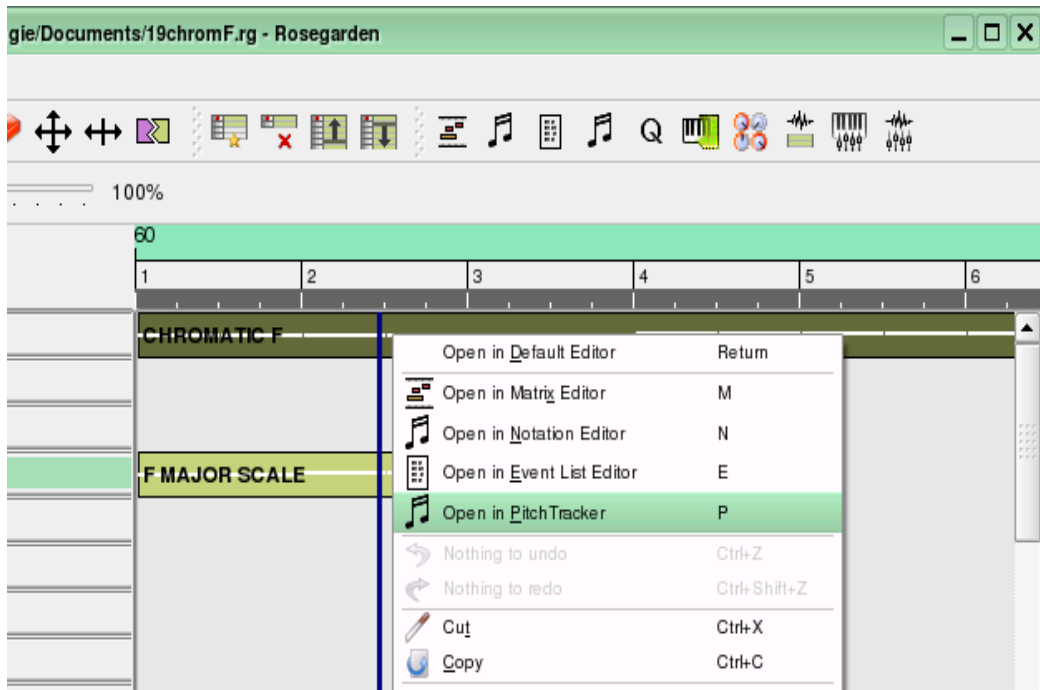
The nature of Rosegarden's representation of pitch means that a the accidental of a note does not need to be explicitly defined. This has serious implications for the translation to frequency in certain microtonal systems. Therefore, when entering a microtonal score for the pitch tracker – notes should be entered entirely by hand (instead of using keyboard shortcuts or transpositions) to ensure that accidentals are explicitly defined. (A solution to this problem is being implemented.)

### Using the Codicil

The main window will appear when Rosegarden starts. This contains the 'Segment View' which displays midi and audio segments arranged by track and time.

Open an existing score file. ( Menu Item: File->Open )

To open the a pitch tracker, **Right-click** on a Segment (green boxes) and select “**Open in Pitchtracker**”.



A tracker view will appear, and the preferences dialog will open automatically.

The **Input Source** determines what source will be tracked. To track from a microphone select `alsa_pcm:capture_1` or `alsa_pcm:capture_2`.

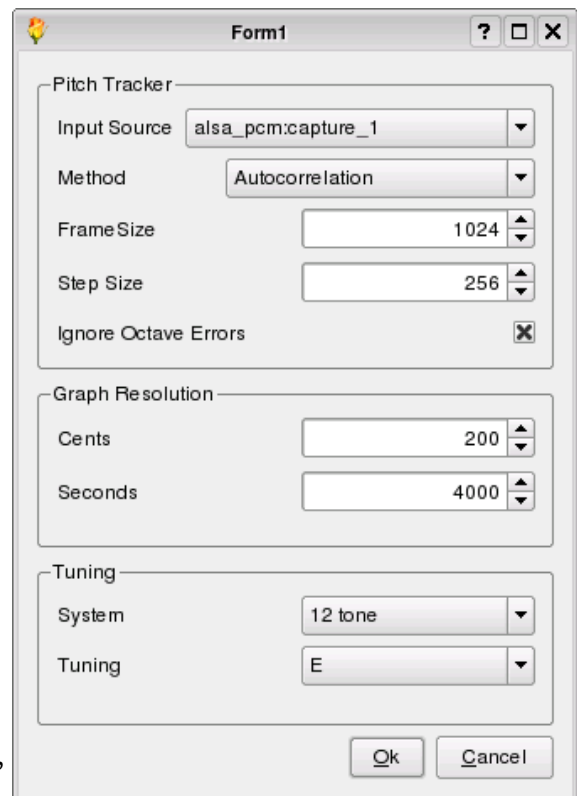
The **Method** selects the pitch tracking algorithm. Different algorithms may provide more reliable results on certain instruments.

The **Frame Size** and **Step Size** affect the pitch tracker performance. High values increase accuracy and reliability at the expense of performance. The default values should be adequate for most situations.

The most common error in pitch tracking is the 'octave error'. If **Ignore Octave Errors** is selected, the pitch tracker will ignore octave differences. This increases stability, and also allows the performer to play in a different register.

The **Graph Resolution** is denoted in cents (vertical) and milliseconds (horizontal).

To choose a **Tuning System**, first select a **System** eg. 19 tone ET. Then select a **Tuning**. The tuning determines the reference note for the tuning. The pitch class selected in the tuning



box will be the same frequency in the current system as it is in 12-tone ET (A=440Hz).  
New tuning systems are entered into the 'Tunings' file see Section X.  
**NB.** This area is still under heavy development

### ***The Pitch Tracker Window***

The Pitch Tracker Window combines the Notation Editor with a real-time pitch tracker. The vertical axis represents cents with the target frequency in the middle. The horizontal axis represents time with the current time being the right edge of the graph. Vertical grey lines traverse the graph to indicate note on/off events. A red line indicates that the amplitude is below the minimum threshold.

To begin tracking, simply press **PLAY** in the Pitch Tracker Window.

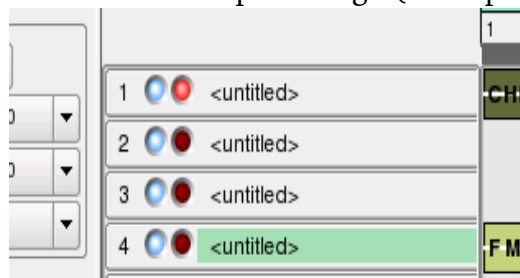
## **Useful Operations**

### ***Changing the tempo:***

Menu Item: Composition -> Tempo & Time Signature -> Add Tempo Change (a tempo change can be applied to a whole composition)

### ***Muting/Unmuting a track***

The blue button to the left of a track's name will mute/unmute the track.



### ***Troubleshooting:***

Many problems can be identified by running Rosegarden from the command line and examining the output.

Pitch Tracker Window will not open

1. There must be a tunings file in the current working directory (copy example from source tree to Home directory)
2. Jack must be active.
3. No other instances of a pitch tracker may be open.
4. Only one track may be selected for tracking.

No audio input (flat red line in tracker)

1. Ensure that the pitch tracker is recording from the correct source. (alsa\_psm:capture for microphone)
2. Make sure that microphone record is enabled and turned up in the mixer.

Unreliable pitch tracker output

1. Try another pitchtracker method.
2. Check “Ignore Octave Errors”
3. Make sure the input levels are not clipping.

Incorrect conversion from notated pitch to frequency **Or**  
 Certain notes are not being tracked

1. Re-enter the note manually into the score.
2. Make sure the note is included in the tuning definition

## Tuning Definition

The tunings file defines the list of all tunings for the Codicil. The syntax of the tuning file was originally based on the Scala scale file.

An example tunings file can be found in the root of the source tree. The Codicil currently requires the tunings file to be in the current working directory when RG is run. It is therefore advised that the file is copied into your home directory unless you wish to use the command line to run Rosegarden.

The following shows an extract from a tunings file. The System is given the name '19-tone'. A system can contain many tunings. The note following the 'Tuning' keyword defines the reference note for the scale. (Eg. An 'A' in this tuning will be exactly equal to an 'A' in 12-tone ET where A=440Hz.)

```
System
19 tone

Tuning A
#cents,  spelling
0.0,     A
63.158,  A2
126.316, B-2
189.474, B
252.632, B2

...

endTuning

endSystem
```

The first column in the Tuning section defines the interval list. These are defined as distance from tonic in cents or an interval ratio eg 3/2 . A cents value **MUST** contain a decimal point. This is consistent with the Scala tuning file.

Each interval is followed by a comma-separated list of enharmonic spellings. The spellings are written as a pitch class optionally followed by an accidental. The accidentals are defined as follows:

Double Sharp	4
Three-Quarter Sharp	3
Sharp	2
Quarter Sharp	1
Quarter Flat	-1
Flat	-2
Three-Quarter Flat	-3
Double Flat	-4

Tunings and Systems must be terminated with the keywords endTuning & endSystem respectively.