

# Summercode project plan

Sakari Bergen

March 21, 2008

## Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
1.1	Ardour . . . . .	2
1.2	Project goals . . . . .	2
1.3	About me . . . . .	2
1.4	Mentoring and other sources of help . . . . .	3
<b>2</b>	<b>Motivation</b>	<b>3</b>
2.1	Meta data and tagging . . . . .	3
2.2	Export functionality . . . . .	3
<b>3</b>	<b>Project details and planned schedule</b>	<b>3</b>
3.1	Meta data . . . . .	3
3.2	Export . . . . .	4
3.3	Tagging exported files . . . . .	7
3.4	Fine tuning and testing . . . . .	7
<b>4</b>	<b>Risk management</b>	<b>7</b>
4.1	Meta data and tagging . . . . .	7
4.2	Export . . . . .	8
<b>5</b>	<b>Benefits</b>	<b>9</b>
5.1	Meta data and tagging . . . . .	9
5.2	Export functionality . . . . .	9

# 1 Overview

## 1.1 Ardour

Ardour<sup>1</sup> is a free software hard disk recorder and digital audio workstation application released under the GNU General Public License. It is written in C++ and currently runs on Linux, Solaris, and Mac OS X.

Ardour is currently the most promising open source audio workstation available that strives to meet the needs of professional users. Its development is currently sponsored by The School of Audio Engineering (SAE)<sup>2</sup> to the extent that, together with donations from users, it is possible for the primary author, Paul Davis, to work on Ardour full-time.

## 1.2 Project goals

The goals of my project are to add meta data support and new audio export functionality to Ardour. Meta data support will include tagging exported files in their native tagging format. Achieving both goals will make working on music collaboratively and sharing it a lot easier. Code will be committed to Ardour's SVN-repository and will be included in Ardour version 3.0 or later, depending on when 3.0 is released.

## 1.3 About me

I am a second year student at the Helsinki University of Technology. My major is Communications Engineering and I have a minor in Computer Science and Engineering. I have been an enthusiastic open source user since 2002. Lately I have also started to participate in open source development and published some programming projects of my own as open source on my website<sup>3</sup>.

I have always been interested in programming, and developed my skills through hobby projects and taking courses in school. Most of my practical C++ experience is from our school C++ project work, which received maximum points<sup>4</sup>. My skills have also developed a lot through hacking on Ardour.

I have been an Ardour user for quite some time, and after learning the appropriate programming skills I immediately found myself contributing to the Ardour codebase, ending up fixing a couple of bugs for the next release<sup>5</sup>. Recently I have also started developing some new functionality for Ardour<sup>6</sup>.

---

<sup>1</sup><http://ardour.org> - Ardour website

<sup>2</sup><http://ardour.org/node/976> - SAE Institute Sponsors Ardour Open Source DAW Project

<sup>3</sup><http://beatwaves.net/> - My personal website

<sup>4</sup>[http://beatwaves.net/software/crumbled\\_earth](http://beatwaves.net/software/crumbled_earth) - Crumbled Earth (C++ project work)

<sup>5</sup><http://tracker.ardour.org/view.php?id=1804> and <http://tracker.ardour.org/view.php?id=1681> - Ardour bug tracker entries for bugs I have fixed (username: SaBer)

<sup>6</sup><http://beatwaves.net/node/17> - Ardour session merger project

## 1.4 Mentoring and other sources of help

Sampo Savolainen, who has been an Ardour developer for about four years, has promised to mentor me in the project. In addition to having knowledge of both the engine and UI of Ardour, Sampo, as an Ardour user himself, is also interested in Ardour's usability.

The best part about Ardour is its active community. The most active users and developers are commonly available on IRC at #ardour @ freenode for feedback and advice on issues related to Ardour. Ardour also has user and developer mailing lists and a discussion forum on its website. During the time I have spent coding Ardour, its main developer, Paul Davis, has been a lot of help mainly through IRC discussions, and will most probably continue to do so. However, he was not able to make any promises concerning mentoring next summer, but has promised to be a contact person on Ardour's behalf when necessary.

## 2 Motivation

### 2.1 Meta data and tagging

In a world where there is a lot of information available, proper meta data is very important. Most media files these days have some sort of tagging possibilities, but adding meta data manually to a multitude of files is not the way tagging should be done. Instead, tagging should be done by the program that originally created the file, and this is exactly what the goal of this new functionality is.

### 2.2 Export functionality

A user survey related to Ardour's export functionality<sup>7</sup> revealed many points of improvement in it. There have also been feature requests from users related to export, that need addressing. These problems are addressed in the proposed project.

## 3 Project details and planned schedule

My planned project schedule can be seen in figure 1. Detailed descriptions of each task can be found below.

### 3.1 Meta data

With the huge amount of information available these days, having proper meta data is very important. My goal is to add features to Ardour that make it possible to include important meta data, such as licensing, copyright, name and artist information, into both Ardour sessions and exported files. Having

---

<sup>7</sup><http://lalists.stanford.edu/laa/2007/02/0665.html> - User survey on Ardour's export design

Task	Week	23	24	25	26	27	28	29	30	31	32	33	34	35
<b>1. Meta-data</b>														
1.1. Support for meta data in session files														
1.2. GUI components for meta data														
<b>2. Export</b>														
2.1. Silence trimming and adding														
2.2. Normalization														
2.3. Filename handling														
2.4. Real-time exporting														
2.5. Encoding options														
2.6. Multichannel export														
2.7. Audio format profiles														
2.8. Exporting of multiple files														
2.9. GUI work														
<b>3. Tagging Exported files</b>														
<b>4. Fine tuning and testing</b>														

Figure 1: planned schedule

such functionality in Ardour would encourage people to include proper meta data in their files. Especially specifying a license would broaden possibilities in sharing and reusing openly licensed audio material and thus boost collaborative creativity.

### 3.1.1 Support for meta data in session files

Ardour sessions are stored in one XML-file and a bunch of data files. The XML-file describes the structure of the session and is also a good place to store meta data associated to the session.

- A set of XML-elements for defining relevant meta data will be defined
- Backend functionality for storing and editing this data will be added

### 3.1.2 GUI components for meta data

Metadata within a session will have to be created and edited somehow. A GUI component for doing this will be needed.

- A GUI component for editing metadata will be added

## 3.2 Export

The basic purpose of Ardour's export functionality is to create mixdowns of multitrack arrangements. On the user interface side, this happens through a dialog. In many cases, exporting as it is currently implemented doesn't deliver the desired final result; additional steps and tools are needed. This is inefficient, and is more demanding on the user.

The goal is to streamline and add functionality to Ardour's export dialog. This would make it easier to share and store both ready compositions and

snapshots of work in progress. Industrial designer Thorsten Wilms has done thorough concept work on renewing the export dialog, covering everything from a user survey to GUI mock-ups<sup>8</sup>. This work will provide an excellent basis for my work. New export functionality will include:

- More encoding options (codecs and containers)
- Audio format profiles
- Exporting to multiple files with different formats at once
- A possibility to include a date and time in filenames
- Trimming and/or adding silence to the beginning and end of files
- Normalization of audio
- Multichannel export
- Tagging support
- Allowing real-time exporting for increased compatibility with some applications
- Many usability improvements

### **3.2.1 Silence trimming and adding**

In some cases the exact start and end of a session are not easy to mark for export. Trimming silence automatically saves space and allows gap-less playback. In some cases, it might be desirable to have a specific duration of leading or trailing silence.

- An option for trimming and/or adding silence to exported files will be added

### **3.2.2 Normalization**

Normalization is a process in which the volume level in an audio file is set as high as possible without causing clipping. Clipping causes loss of information and distortion, which in most situations is a bad thing. Normalization of exported files is generally a good thing to do.

- An option for normalizing exported files will be added

---

<sup>8</sup>[http://thorwil.files.wordpress.com/2007/08/export\\_design\\_2007-08-11.pdf](http://thorwil.files.wordpress.com/2007/08/export_design_2007-08-11.pdf) - Ardour Export Redesign

### 3.2.3 Filename handling

When exporting many files, naming the files wisely can be tedious. Naming files automatically is not flexible, so a user controlled semi-automatic file naming mechanism will be implemented. File names may include the session name, a free label, the current date and a revision number.

- A system for semi-automatic file naming will be added

### 3.2.4 Real-time exporting

At the moment Ardour uses a mechanism called Jack freewheeling at export. This is a mode in the Jack Audio Connection Kit (the audio server Ardour uses), that allows running clients faster or slower than real-time. Not all Jack clients support the freewheeling mode, so an option for doing a real-time export in Ardour is needed.

- An option for real-time export will be added

### 3.2.5 Encoding options

At the moment Ardour can export only linear encoded sound files via libsndfile. In practice this means the only available choices are different container formats such as AIFF, WAV and RAW. However, libsndfile supports other formats also. This should make implementing support for those formats straightforward.

- Basic support for exporting with different encodings and containers shall be added to the backend. This will include lossily and losslessly compressed formats such as Ogg Vorbis and FLAC.
- The amount of encodings implemented will depend on how long implementing them takes. The goal is to add a couple of useful options to allow proper implementation of the rest of the project, rather than cover a multitude of encoding options.

### 3.2.6 Multichannel export

Not all Audio projects made with Ardour are stereo or mono. Some support for multichannel mixing already exists in Ardour, so it is logical for one to be able to export multichannel files.

- Support for basic multichannel export will be added

### 3.2.7 Audio format profiles

An audio format profile describes the container, encoding, quality and other options related to an exported file. In addition to pre-defined format profiles, such as 'Audio CD' the user can create and edit his own profiles. Audio format profiles are also the only GUI representation of different encoding options.

- The basic structures for storing and editing audio format profiles will be added to the backend
- A GUI component for editing these profiles will be created

### **3.2.8 Exporting of multiple files**

Sometimes it is necessary to export many different files at the same time or the same audio in multiple different formats. Adding an option for exporting multiple files at a time will enable the user to do multiple exports without opening the export dialog more than once.

- An option for exporting multiple files will be added

### **3.2.9 GUI work**

Not all features described above need extensive GUI reworking. These features will be implemented first into the old GUI. Doing so will make it easier to implement the new GUI components somewhat all at once. However new features are best tested from the GUI, so the GUI will be built concurrent to implementing new features.

- A new GUI for the export dialog will be implemented as described in Thorsten Wilms' document

## **3.3 Tagging exported files**

It should be possible to include meta data from the session into exported files. This will be done using standard tagging features already present in different audio formats: Vorbis comment for FLAC and Vorbis, iXML for BWF etc.

- Tagging support for exported files will be added

## **3.4 Fine tuning and testing**

Making GUI components behave well is a lot of work. This is why I have reserved two weeks for testing and fine-tuning mainly the GUI components. Also, since the export dialog functionality consists of many small functionalities brought together into a few dialogs, it might be necessary to do temporary incomplete implementations of some functionality, to be able to test others. The work needed for finishing up these incomplete parts is also included in this section.

# **4 Risk management**

## **4.1 Meta data and tagging**

- Tagging formats are hard to use

- Probability: Low
- Consequences: Medium
- Description: Some tagging format turns out to be difficult to implement. This could lead to a deviation from the schedule or the specific tagging format could be left out.
- Defining a satisfying set of meta data attributes causes problems
  - Probability: High
  - Consequences: Low
  - Description: A consensus on the set of attributes to store into meta data is not reached quickly enough. This will probably lead to a format change later on.

## 4.2 Export

- The dialog design is flawed
  - Probability: Low
  - Consequences: Medium
  - Description: The design work made by Thorsten Wilms has some fundamental flaw in it. This would lead to more work doing some redesigning.
- The designed dialog GUI components are hard to implement
  - Probability: High
  - Consequences: Low
  - Description: Some designed GUI components are hard to implement. GUI programming in general can be tricky, and schedules including GUI programming are hard to make. The risk is deviating from the schedule.
- Some functionality needs deep changes in Ardour
  - Probability: Low
  - Consequences: Medium-High
  - Description: Trying to implement some functionality reveals problems in the Ardour code base that need fixing. This could lead in a moderate to high increase in workload.



## **5 Benefits**

### **5.1 Meta data and tagging**

Meta data will be available in both the session itself and the files exported from it. This will improve the general quality of media created with Ardour. It also makes it easier to create libraries of sounds or music, which can be searched in many different ways. Encouraging users to include a license in their files would be a step towards having lots of collaboratively created freely licensed music available for both listeners and composers.

### **5.2 Export functionality**

The usability and usefulness of the export dialog will increase. Many tasks that before required the user to use many tools and/or do a lot of manual work can be done easily and efficiently. This will especially make it easy to share ideas and creations with others and also help the user keep his files in order.