Patacrep !

LaTeX Songbook for Guitar tabs

Documentation

Date : 4 novembre 2011
Authors : Crep (R. Goffe) & Lohrun (A. Dupas)

Contents

Introduction

In this document, we detail how to use the tools that are available on http://www.patacrep.com in order to produce customized songbooks. Following is a brief insight of the two main projects:

Songbook

This project corresponds to a set of scripts to build a songbook. It also contains the set of songs/tabs that have already been written. You may download the whole pdf at http://www.patacrep.com/data/documents/songbook.pdf as an example.

Songbook-client

This application is a graphical interface to the songbook.

Technologies and upstream projects

The Patacrep songbook derives from the *Songs LaTeX Package*[1].

- LaTeX : songs are written according to a set of simple rules that allow to use LaTeX for text rendering. As a result, a songbook is a document with high typographical quality ;
- Lilypond : free software to write high quality music sheets;
- Makefile : the production chain of a songbook is fully automated. Just type *make* in a terminal.
- Python : several scripts, for example authors/titles indexes' generation, are written python.
- C++/Qt4 : language/toolkit for the songbook-client application.

Licenses

Software is distributed under the GPLv2 license[2]. All documents and graphics are distributed under a free Creative Commons license CC-By-Sa[3].

[1] http://songs.sourceforge.net/

[2] http://www.gnu.org/licenses/gpl.html

[3] cc BY SA http://creativecommons.org/

1 Songbook

Sommaire

1.1 Setup

Globally, avoid any file/directory name with spaces or special characters. In this guide, we consider that the working directory is: `$HOME/songbook` which may also be referred to as `/home/user/songbook` or `∼/songbook`.

1.1.1 Linux

A songbook uses LaTeX to render the songs' layout and Python to generate indexes (required dependencies). Music sheets can be included if written with Lilypond and source code is hosted/distributed as a Git repository (recommended dependencies). Finally, a pdf reader application is required to open produced songbooks. For Debian/Ubuntu users :

Dependencies

Install required dependencies.

```
sudo apt-get install texlive-base texlive-latex-extra
sudo apt-get install texlive-lang-french python
```

Install recommended dependencies.

```
sudo apt-get install texlive-fonts-recommended
sudo apt-get install lilypond
```

Sources

Sources can be retrieved as a `.tar.gz` archive

```
wget http://www.patacrep.com/data/documents/songbook.tar.gz
tar xzvf songbook.tar.gz
```

or as a Git repository

```
git clone git://git.lohrun.net/songbook.git
```

Compile and run

The compilation chain is automated through a *makefile*. In your songbook's directory:

```
make all
```

Some makefile's options:

- `$ make clean` : removes logs and temporary files produced during the compilation step, except `.pdf` files.
- `$ make cleanall` : removes all temporary files and produced `.pdf` .
- `$ make lilypond` : generates music sheets corresponding to Lilypond sources (`.ly` files) as `.pdf` in `./lilypond` . The Lilypond software is required.
- `$ make book.pdf` : since `make` or `make all` may take some time, if you are only interested in a single songbook, you can specify the book (`.ext`) you want. Instead of `book.pdf` , you may choose:

make	description
`songbook.pdf`	: full version with the whole set of songs and music sheets
`volume-1.pdf`	: volume 1 (171 songs, 152 pages)
`volume-2.pdf`	: volume 2 (180 songs, 157 pages)
`naheulbeuk.pdf`	: a songbook for the *Donjon de Naheulbeuk*
`english.pdf`	: english songs only
`french.pdf`	: french songs only

Note that the `make` command without any option corresponds to `make songbook.pdf` .

1.1.2 Windows

Dependencies

Install required dependencies.
– Miktex : `http://miktex.org/`
– Python : `http://www.python.org/download/releases/2.7.2/`

⚠ `The compilation process under Windows has been tested with MikTeX 2.9 and Python 2.7.2.`

Install recommended dependencies.
– Lilypond : `http://lilypond.org/install/`

Modify the `Path` variable

Restart your computer after installing MikTeX and Python. The `Path` variable now needs to be modified.
– Right-click on My Computer, then Properties.
– In `Parameters > Advanced Settings`, click on Environment Variables.

In the lower part (System Variables), scroll the list until the `Path` command, then double-click to modify. The `Path` contains a set of paths that are separated by semicolons. The following paths must be present:
– Python : `C:\Python27`
– Miktex : `C:\Program Files\MiKTeX 2.9\miktex\bin`
– Lilypond : `C:\Program Files\LilyPond\usr\bin`

⚠ `Pay attention to version numbers. These directories are just examples that may need to be adapted according to your downloaded software version.`

Restart your computer once your `Path` is modified.

Sources

Sources can be retrieved as a `.tar.gz` archive at:

`http://www.patacrep.com/data/documents/songbook.tar.gz` .

Its content can be extracted with third-party softwares such as 7zip[1] or Winrar[2]. Alternatively, in the case you have installed a Git[3] client, for instance Mysysgit[4], you can download the sources with:

[1] `http://www.7-zip.org/`
[2] `http://www.win-rar.com/`
[3] `http://git-scm.com/`
[4] `http://code.google.com/p/msysgit/`

```
git clone git://git.lohrun.net/songbook.git
```

Compile and run

1. Open the command prompt : Windows+r, enter `cmd` and validate with Enter (Figure ??).
2. Go to the songbook directory using `cd`. For instance,

   ```
   cd C:\
   cd songbook
   ```

3. Use the file `make.bat`, giving a `.sb` file as parameter. For instance,

   ```
   make.bat songbook.sb
   ```

MikTeX is a LaTeX distribution that allows to install missing packages “on the fly”. A songbook usually requires a few additional packages (Figure ??).

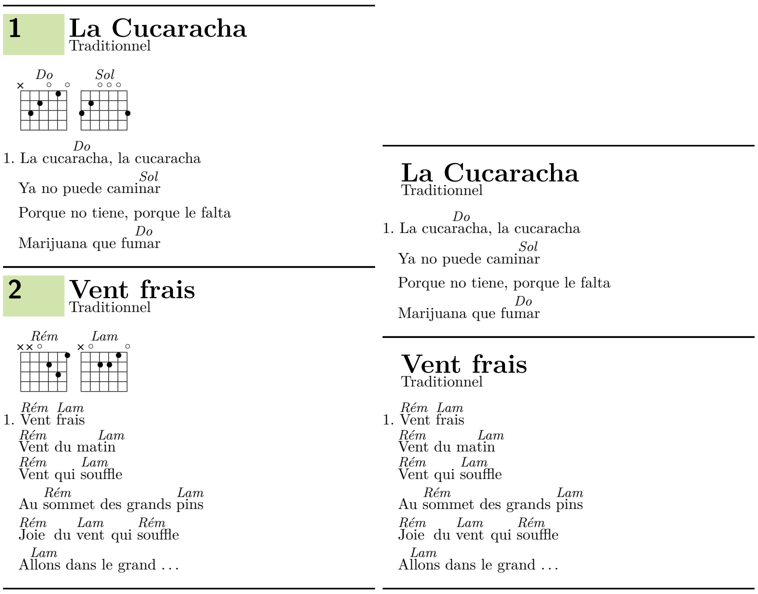
1.2 Content description

`./utils`

Contains a set of scripts. To run a script, enter the following command in a terminal:

```
./utils/script.sh
```

- `resize-cover.sh` : automatically resize `.jpg` files of the `songbook/songs` directory. Those files correspond to songs’ covers. Run this script after introducing a new jpg cover.
- `rules.py` : apply a set of rules for LaTeX specific notations or chords. Run this script after adding a new song or after modifying an existing one in order to preserve global songbook standards.
- `typo.sh` : apply a set of typographical rules such as the removal of double/trailing spaces or language specific rules. Run this script after adding a new song or after modifying an existing one in order to preserve global songbook standards.
- `new-songs-list.sh` : allows to retrieve the list of songs that have been added since the last release.
- `make-html.sh` : builds an html index for the whole set of songs.



1 La Cucaracha
Traditionnel

Do
1. La cucaracha, la cucaracha
Sol
Ya no puede caminar
Porque no tiene, porque le falta
Do
Marijuana que fumar

2 Vent frais
Traditionnel

Rém Lam
1. Vent frais
Rém Lam
Vent du matin
Rém Lam
Vent qui souffle
Rém Lam
Au sommet des grands pins
Rém Lam Rém
Joie du vent qui souffle
Lam
Allons dans le grand ...

(a)

La Cucaracha
Traditionnel

Do
1. La cucaracha, la cucaracha
Sol
Ya no puede caminar
Porque no tiene, porque le falta
Do
Marijuana que fumar

Vent frais
Traditionnel

Rém Lam
1. Vent frais
Rém Lam
Vent du matin
Rém Lam
Vent qui souffle
Rém Lam
Au sommet des grands pins
Rém Lam Rém
Joie du vent qui souffle
Lam
Allons dans le grand ...

(b)

Figure 1.1: Different templates may be used for a same songbook. (a) Template `patacrep.tmpl`; (b) Template `minimal.tmpl`.

`./templates`

Contains some files corresponding to different songbook styles. A template contains some parameters such as the font size, color settings etc.

- `patacrep.tmpl` : default template.
- `patacrep-en.tmpl` : english version of the default template.
- `minimal.tmpl` : a more compact version without front page nor indexes.

`./songs`

Contains the set of available songs. Each song is a text file with the `.sg` extension. A song is found in a sub-directory named by the corresponding artist. For instance: `songs/Pornophonique/Sad_robot.sg`.

`./lilypond`

Contains the Lilypond music sheet sources as `.ly` files.

`./tex`

Contains LaTeX files such as packages, licenses, classes etc.

`./books`

Contains all `.sb` files that correspond to the different pdf books available at `www.patacrep.com` .

`./img`

Contains the images that may be inserted in a songbook (except album covers). To add the image `my_image.jpg` in a song, use the `\image` macro inside a `.sg` file. This macro behaves like the LaTeX macro `\includegraphics`. For example:

```
\image[width=4cm]{my_image}
```

1.3 Start a new songbook

In order to build a songbook, a `.sb` file is required whose content is detailed in the following. A songbook file (`.sb`) contains a set of values for the options that are provided by a template and a list of songs. A `.sb` file must be saved into the `./books` directory. Following is an example of a songbook file based on the minimal template and containing two songs.

```
{
"template" : "minimal.tmpl",
"bookoptions" : [
    "diagram",
    "lilypond",
    "pictures"
  ],
"songs" : [
    "Le_Donjon_de_Naheulbeuk/10_sous_dans_ma_poche.sg",
    "Le_Donjon_de_Naheulbeuk/Bugger_off.sg"
  ]
}
```

Available options are defined in the template file. In this case, the value *["diagram", "lilypond", "pictures"]* has been set for the option *bookoptions* of the `minimal.tmpl` template. Using the `patacrep.tmpl` template, the following options are available:

Option	Description	Type
title	songbook title	character string
author	songbook authors	character string
booktype	whith/without chords	`chorded` or `lyric`
bookoptions	display elements	`diagram`, `lilypond`, `pictures`, `tabs`
version	current pdf version	character string
subtitle	songbook subtitle	character string
web	Internet website	character string
mail	contact email	character string
picture	front page picture	path to image file `.png`, `.jpg` or `.pdf`
picturecopyright	picture copyright	character string
footer	front page footer	character string
license	document license	character string
mainfontsize	document font size	10, 11 or 12 pt
songnumberbgcolor	songs number background color	hexadecimal color code
notebgcolor	textnote background color	hexadecimal color code
indexbgcolor	color of indexes' links	hexadecimal color code

Once your songbook `mybook.sb` is written, produce the corresponding pdf with:

```
make mybook.pdf
```

1.4 Writing a song

1.4.1 Main elements

A song is a plain text file `song.sg` that is saved in a `songs/Artist` directory. Remember that filenames do not accept spaces (replace them with underscores). A `.sg` file header is composed of the following elements :

```
\beginsong{Title}
  [by=<Artist>,cov=<album-cover>,album=<Album>]
```

The `Title`, `Artist`, `album-cover` and `Album` parameters must be set for each new song. The field `album-cover` denotes a file `album-cover.jpg` that is in the same directory of the corresponding `.sg` song.

A song is a sequence of verses and chorus. Use the macros `\beginverse` and `\endverse` to determine the text of a same verse. Similarly, use the macros `\beginchorus` and `\endchorus` for the text of a same chorus.

To insert a guitar chord in the text, a special command is used. For instance, the `\[E]` command will print the chord `E` above the following syllable in the pdf.

```
\beginverse
  His \[Dm]steely skin is covered
  By \[F]centuries of dust
  \[C]Once he was a great one
  \[Dm]Now he's dull and rust
\endverse
```

A song ends with the macro `\endsong`.

1.4.2 Style guidelines

Chords notation

Chords follow the British/American notation (A, B, C, D, E, F, G). By default, the chord is major (C refers to C major). Minor chords are written with a lower-case `m`. The flat symbol (♭) is obtained with the symbol `&`. The sharp symbol (♯) is obtained with the character `#`. For example, the `A flat minor` chord is written `\[A&m]`.

Repeating chords

In order to maintain an easy to read and compact document, chords in verse and chorus are only specified once. This may be harder to decipher the first time but in the end, the text is much more easier to read and that's a lot less of pages to print.

Typography

- Each line begins with a capital letter;
- No punctuation (except "! or ?") at the end of lines;
- Symbols such as "; : ! ?" require an unbreakable space before in French but not in English ;
- French capital letters are accentuated (e.g "À bientôt" and not "A bientôt").

Verse numbering and line breaks

The numbering is automatically performed and increased with each new `\beginverse` environment. However, it is often convenient to decompose a same verse in two parts, without numbering the second part. To achieve this, use the `\beginverse*` macro. For instance, a verse composed of eight verses (lines) may usually be decomposed in two verses of four lines. For example:

```
\beginverse
  His \[Dm]steely skin is covered
  By \[F]centuries of dust
  \[Do]Once he was a great one
  \[Dm]Now he's dull and rust
\endverse

\beginverse*
  An oily tear he's crying
  Can you feel the pain
  Of the sad, sad robot
  And it's driving him insane
\endverse
```

Columns layout

The macro `\songcolumns` determines the number of columns of the song in the final pdf. Use this macro right before the `\beginsong` command. Usually, a song is presented on 1, 2 or 3 columns. By default, specify two columns.

```
\songcolumns{2}
\beginsong{Title}
```

Special characters

A few characters must be written according to LaTeX commands for a better rendering in the pdf. For example, ellipsises should be written by the command `\dots`. See also Section 1.2.

Echo and repeat

When a sentence is repeated several times, use the command `\rep` instead of writing *bis* or (x4). For example, if the word *Hallelujah* is repeated four times, write:

```
Hallelujah \rep{4}
```

The command `\echo` refers to choirs or alike.

```
Hallelujah \echo{Hallelujah}
```

Chords diagrams

Given that the same guitar chord may be played differently and that one position may be better than another, the songbook allows to represent the chord as a diagram at the beginning of the song through the command `\gtab`, right before the first verse of the song. Following are a few examples:

```
\gtab{C}{3:002220}
\gtab{D}{XX0232}
\gtab{E}{022100}
\gtab{Am}{X02210}
```

```
X is the capital letter x. The lower-case variant is not supported.
0 is the figure zero and not the capital letter O.
```

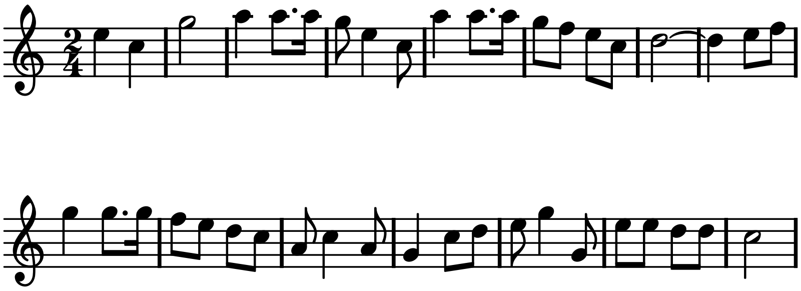
– the six figures correspond to the six guitar strings (`E`, `A`, `D`, `G`, `B`, `E`) ;
– the value indicates the fret to be played;
– 0 denotes an open string;
– X indicates that the string should not be played;
– a value before ":" denotes a barre (*3:* denotes a barre chord on third fret).

1.4.3 Lilypond music sheets

If you want to add a melodic pattern in a song, you may use Lilypond syntax to write the score. In this regard, open a new file `score.ly` in the repertory `songbook/lilypond`. Import the file `header` at the beginning of your file and define the option `paper-height` to ensure that the pdf score fits one single page with minimum whitespace. A first approach is to count 1.6 cm for a single line. Then, write your score between brackets (see Figure 1.2).

Finally, your file `sheet.ly` can be inserted in a song `.sg` through the `\lilypond` command:

```
\lilypond{sheet}
```



```
\include "header"
\paper{paper-height = 3.3\cm}
{
  \key c \major
  \time 2/4
  \relative c’’
    {
      e4 c g’2 a4 a8. a16 g8 e4 c8
      a’4 a8. a16 g8 f e c d2~ d4
      e8 f g4 g8. g16 f8 e d c a c4 a8 g4
      c8 d e8 g4 g,8 e’ e d d c2
    }
}
```

Figure 1.2: Source code for a Lilypond music sheet.

1.4.4 Example

Here is a complete example of the song *Sad robot* by *Pornophonique*[5] that may be used as a reference for a new song.

```
\selectlanguage{english}
\songcolumns{2}
\beginsong{Sad robot}
  [by=Pornophonique,cov=8-bit-lagerfeuer,album=8-bit lagerfeuer]

  \cover
  \gtab{Dm}{XX0231}
  \gtab{F}{1:022100}
  \gtab{C}{X32010}

  \lilypond{Sad_robot}

  \beginverse
    His \[Dm]steely skin is covered
    By \[F]centuries of dust
    \[C]Once he was a great one
    \[Dm]Now he’s dull and rust
  \endverse

  \beginverse*
    An oily tear he’s crying
    Can you feel the pain
    Of the sad, sad robot
    And it’s driving him insane
  \endverse

  \beginverse*
    He can’t turn back time nor history
    So his life became a misery
    He has to face the destiny
    Nobody cares anymore
  \endverse

  \beginchorus
    Sad, sad robot
    Sad, sad robot
    Sad, sad robot
    All alone
  \endchorus

  \beginchorus
```

[5] cc BY NC ND http://www.jamendo.com/fr/track/81740

```
    He’s a sad, sad robot \rep{3}
    He’s so alone
  \endchorus

  \beginverse
    Me steely skin is covered
    By centuries of dust
    Once me was a great one
    But now I’m dull and rust
  \endverse

  \beginverse*
    An oily tear I’m crying
    Can you feel me pain
    I’m the sad, sad robot
    Driving me insane
  \endverse

  \beginverse*
    I can’t turn back time nor history
    So me life became a misery
    I have to face me destiny
    That I’m all on me own
  \endverse

  \beginchorus
    Red, red robot
    I’m a red, red robot \rep{2}
    And so I shall return
  \endchorus

  \beginchorus
    I’m a red, red robot \rep{3}
    So I shall return
  \endchorus
\endsong
```

2 Songbook-Client

Sommaire

The *Songbook-Client* application is a graphical interface to build customized songbooks[1].

Independently of your operating system, the songbook's dependencies must have been installed first (mainly LaTeX and Python) in order to build a pdf songbook. See Section 1.1 for instructions.

Downloads are available at `http://www.patacrep.com/static1/downloads`

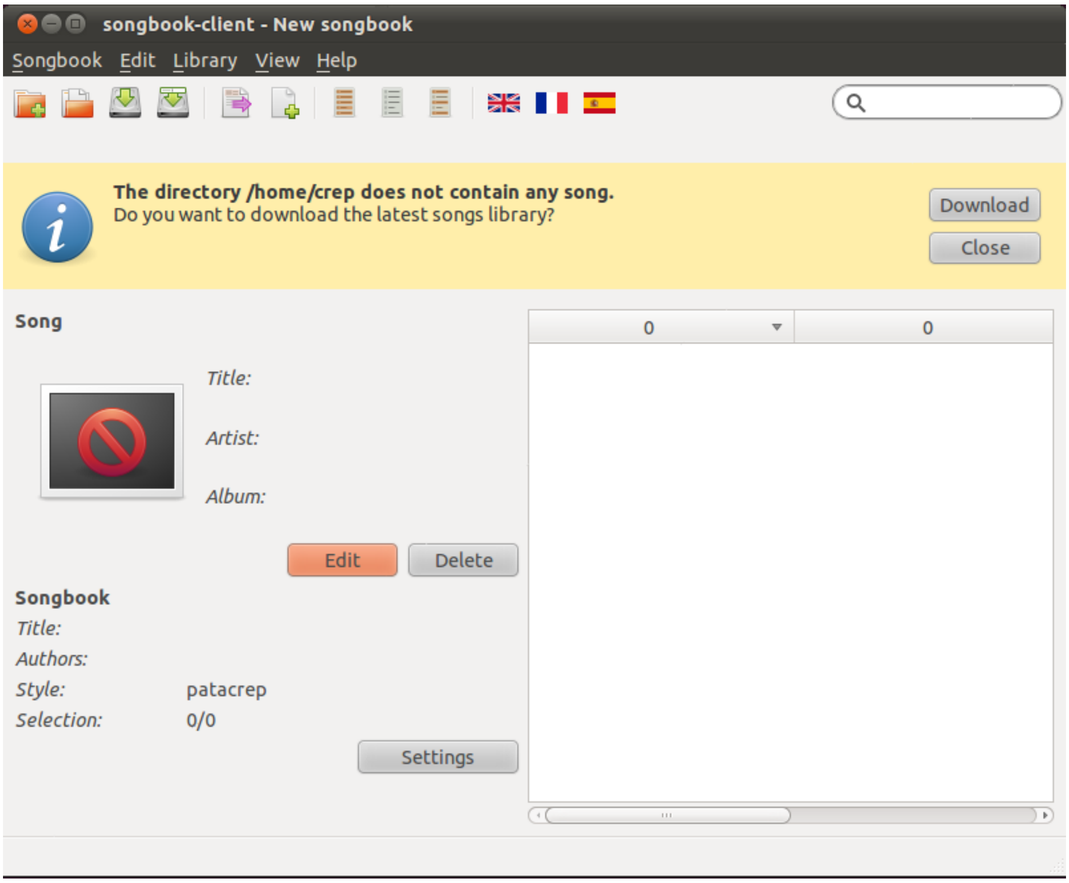
2.1 Interface

2.1.1 First launch

By default, the interface presents an empty list (Figure 2.1). The *Songbook-Client* must be linked to an existing *Songbook*. There are two options:

1. Indicate the path of a `songbook` directory that is on your hard drive from the menu `Edition > Preferences` (Figure 2.2(a)).
2. Download the latest version from the Internet from the menu `Library > Download` (Figure 2.2(b)).

[1] `http://www.ohloh.net/p/songbook-client`, `http://github.com/crep4ever/songbook-client`

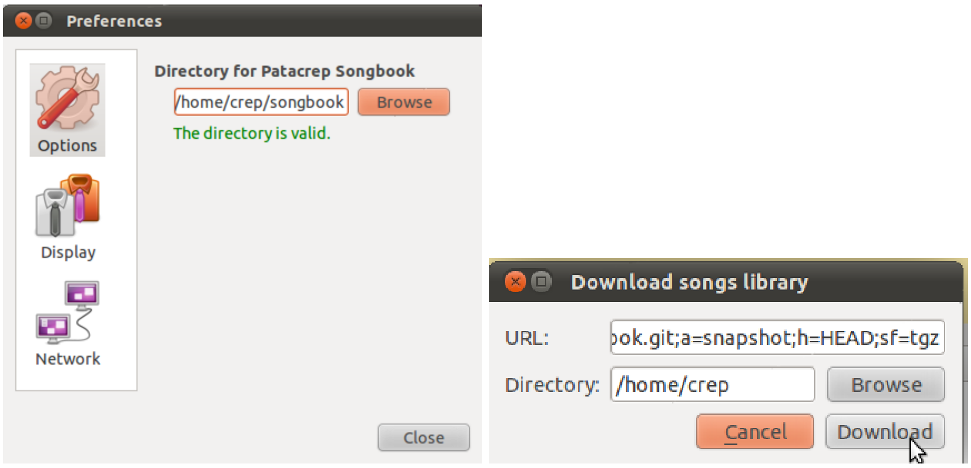


The directory /home/crep does not contain any song.
Do you want to download the latest songs library?

Songbook
Title:
Authors:
Style: patacrep
Selection: 0/0

Figure 2.1: First launch of the application.

]



Directory for Patacrep Songbook

/home/crep/songbook

The directory is valid.

URL: ɔok.git;a=snapshot;h=HEAD;sf=tgz

Directory: /home/crep

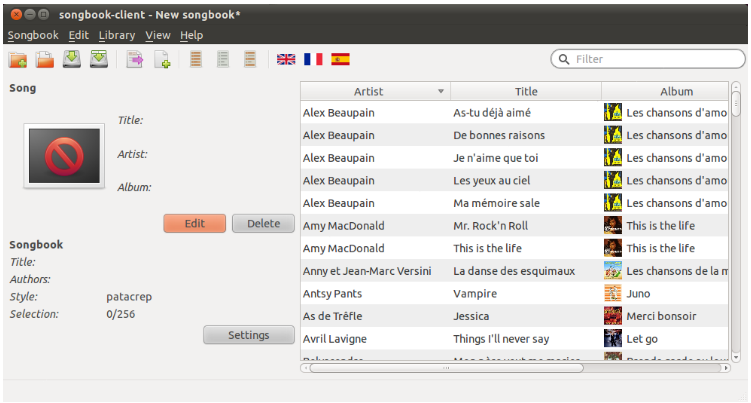
(a) (b)

Figure 2.2: Two options allow to link the *Songbook-Client* to a *Songbook*. (a) Indicate the path to a songbook directory;(b) Download from the Internet.

Once the *Songbook-Client* is linked to a *Songbook*, the *songs' library* is built from the `.sg` files that have been found in the `songs` sub-directory of the *Songbook*.

2.1.2 The songs' library

The set of `.sg` songs found in the `songs` sub-directory is displayed as a list. Columns may be hidden/displayed in the tab `Display` from the menu


songbook-client - New songbook*
Songbook Edit Library View Help
Filter
Edit
Delete
Settings

Song
Title:
Artist:
Album:

Songbook
Title:
Authors:
Style: patacrep
Selection: 0/256

Artist	Title	Album
Alex Beaupain	As-tu déjà aimé	Les chansons d'amo
Alex Beaupain	De bonnes raisons	Les chansons d'amo
Alex Beaupain	Je n'aime que toi	Les chansons d'amo
Alex Beaupain	Les yeux au ciel	Les chansons d'amo
Alex Beaupain	Ma mémoire sale	Les chansons d'amo
Amy MacDonald	Mr. Rock'n Roll	This is the life
Amy MacDonald	This is the life	This is the life
Anny et Jean-Marc Versini	La danse des esquimaux	Les chansons de la m
Antsy Pants	Vampire	Juno
As de Trêfle	Jessica	Merci bonsoir
Avril Lavigne	Things I'll never say	Let go

Figure 2.3: The songs' library.

`Edition > Preferences`. By default, the only visible columns are `Title`, `Artist` and `Album`.

The songs in the library can be selected with a simple click. A selected song is highlighted. Click on a selected song to deselect it.

2.2 Adding a new song

The menu `Library > New song` allows to add a new song to the library (Figure 2.4(a)). The dialog allows to set song's meta-data such as its title, author etc. These fields allow to generate the skeleton of the new file `.sg` corresponding to the song (Figure 2.4(b)). The third step (Figure 2.4(c)) consists in writing the guitar tab. Finally, the song can be selected in the library to obtain the pdf from the menu `Songbook > Build PDF` (Figure 2.4(d)).

2.3 Making a customized songbook

Save/Open

The `.sb` file format saves the list of selected songs and the options that define the appearance of a songbook (see Section 1.3).

Style and options of a songbook

A dialog box allows to quickly select the main style (template) of the songbook and its options (Figure 2.5).

2.4 Compiling from sources

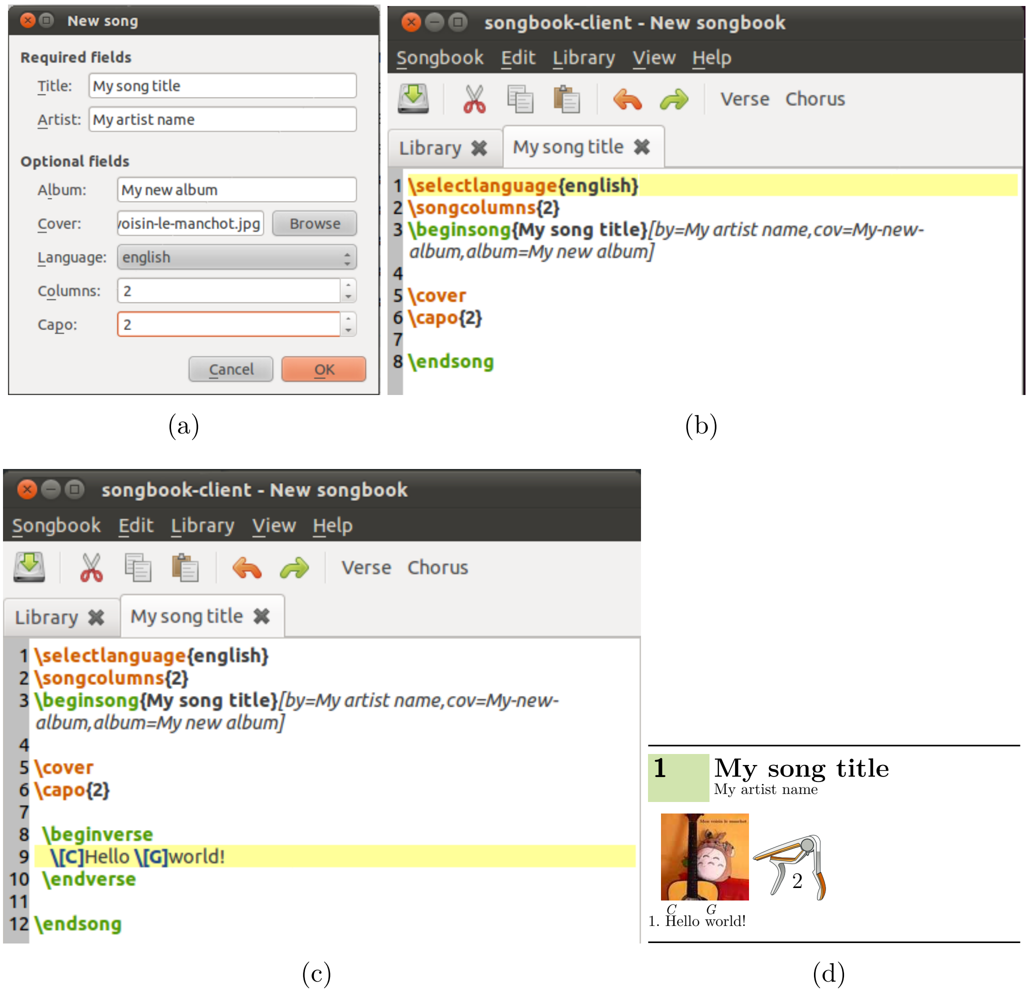
2.4.1 Linux

Dependencies

```
sudo apt-get install build-essential cmake libarchive-dev
sudo apt-get install qt4-qmake qt4-dev-tools libqt4-sql-sqlite
```

Download

```
git clone git://github.com/crep4ever/songbook-client.git
```



New song
Browse
Cancel
OK
songbook-client - New songbook
Songbook Edit Library View Help
Verse Chorus
Library ✖
My song title ✖

Required fields

Title: My song title

Artist: My artist name

Optional fields

Album: My new album

Cover: oisin-le-manchot.jpg

Language: english

Columns: 2

Capo: 2

(a)

```
\selectlanguage{english}
\songcolumns{2}
\beginsong{My song title}[by=My artist name,cov=My-new-album,album=My new album]

\cover
\capo{2}

\endsong
```

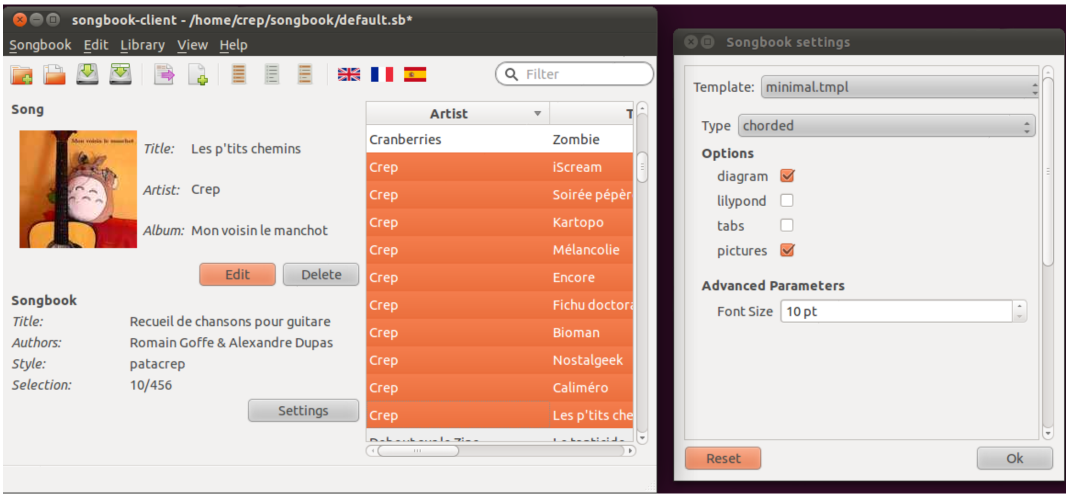
(b)

```
\selectlanguage{english}
\songcolumns{2}
\beginsong{My song title}[by=My artist name,cov=My-new-album,album=My new album]

\cover
\capo{2}

 \beginverse
  \[C]Hello \[G]world!
 \endverse

\endsong
```

(c)

(d)

Figure 2.4: Add a new song. (a) Meta-data information; (b) Auto generation of the `.sg` skeleton; (c) Write the guitar tab; (d) Build the pdf.



Song

Title: Les p'tits chemins
Artist: Crep
Album: Mon voisin le manchot

Songbook

Title: Recueil de chansons pour guitare
Authors: Romain Goffe & Alexandre Dupas
Style: patacrep
Selection: 10/456

Artist	T
Cranberries	Zombie
Crep	iScream
Crep	Soirée pépèr
Crep	Kartopo
Crep	Mélancolie
Crep	Encore
Crep	Fichu doctora
Crep	Bioman
Crep	Nostalgeek
Crep	Caliméro
Crep	Les p'tits che

Template: minimal.tmpl

Type chorded

Options

diagram ☑
lilypond □
tabs □
pictures ☑

Advanced Parameters

Font Size 10 pt

Figure 2.5: Customizing a songbook.

Compile/Run

```
cd songbook-client
make && sudo make install
```

2.4.2 Windows

2.5 FAQ

How to report a bug ?

Through Github[2] or through a post on the forums[3].

Sqlite error at startup

If a warning dialog pops up at startup with some sqlite error check that your operating system supports the Qt's sqlite driver (for Debian/Ubuntu users, install the package `libqt4-sql-sqlite`).

Music scores are not visible

Check that the lilypond checkbox is checked in the songbook tab from the menu `Edition > Preferences`. If the scores are still not visible check that Lilypond is correctly installed on your system.

[2] http://github.com/crep4ever/songbook-client/issues
[3] http://www.patacrep.com/forum/

The songs' library is empty

Check that the path to your *Songbook* is correct in the menu `Edition > Preferences`. The path must contain a makefile and the `songs/` sub-directory.

Errors after renaming/removing a song

Run `make clean` in a terminal or `Songbook > Clean` from the *Songbook-Client*. If you still have an issue you may manually remove all `.d` files within the `~/songbook` directory.