Songwrite documentation

Jean-Baptiste "Jiba" LAMY (jiba@tuxfamily.org)

29th August 2004

Contents

1 Introduction

Songwrite is a tablature editor and player (with Timidity or another Midi player), destined to Linux-user guitarists. It is written in Python, and relies on GNU Lilypond (Lilypond `http://www.lilypond.org`) for printing.

See the README file for installation and requirements.

Songwrite is Free Software. It is available under GNU GPL (doc under GNU FDL).

Pleased the author: do not confound "Free Software" and "Freeware"! In addition to being "Freeware", a Free Software is available as source code; it is allowed to modify and redistribute freely these sources. Thanks to these liberties, Free Softwares are an alternative to capitalism; respect it and join the revolution!

1.1 Supported file formats

Format	read	write	comments
SongWrite format	X	X	XML format
Old format	X		Plain Python serialization (`pickle`)
Midi	X	X	
Rich Midi Tablature		E	Midi with meta event for string numbers, ... EXPERIMENTAL
Ascii tablature	E	X	Import works with SongWrite Ascci tab, but may not work with other...
Guitar pro 3-4	E		
Lilypond		X	No lyrics
LaTeX and LyliPond		X	Tablature with Lilypond, lyrics with LaTeX
PostScript		X	Printable

"E" means EXPERIMENTAL (=not or not enough tested) and "X" fully supported.

The Midi importation has been tested on Midi files generated by Songwrite, GNU Lilypond, Tabledit and NoteWorthy Composer.

Since the 0.9 version, Songwrite uses a new human-readable XML file format (see file `songwrite.schema.xml`, not tested yet). Files saved with the old format can be turned to the new XML format by opening them into Songwrite and then saving them, or with the `convert_to_xml` script. This script takes as arguments the list of file to convert (*e.g.* `convert_to_xml ~/tablatures/*`).

1.2 Mailing list

To subscribe to the Songwrite mailing, send a mail at Songwrite-subscribe@oomadness.tuxfamily.org. To unsuscribe, send a mail to Songwrite-unsubscribe@oomadness.tuxfamily.org.

1.3 Historical stuff about GTablature

Songwrite was previously known as "GTablature". The name has been changed since it no longer belong to the Gnome project. Gtk is now replaced by Tk.

GTablature's users will found Songwrite very similar to GTablature, and the file format is compatible!

To summarize, my reasons for leaving the Gnome project are the following:

- The new version of Gnome (Gnome 2) has lots of incompatibilities with the previous one. In particular, the Python binding for Gtk 1 and 2 cannot be installed at the same time. As i was not ready to re-design the UI part of my application each time the Gnome guys decide to make a new release, i switched to Tk.
- The dark spectrum of .net was floating over Gnome. My favorite language is Python, and contrary to what you may think, Python.net is NOT Python. It is .net, exactely like C++.net is .net and not C++ (have you ever seen a C++ without multiple inheritance?).

1.4 Future direction

See the TODO file for more info.

2 Configuration

The first time you launch Songwrite, the configuration dialog popups. The default configuration should be right for any Linux distributions; you may just be interested by using a different Midi

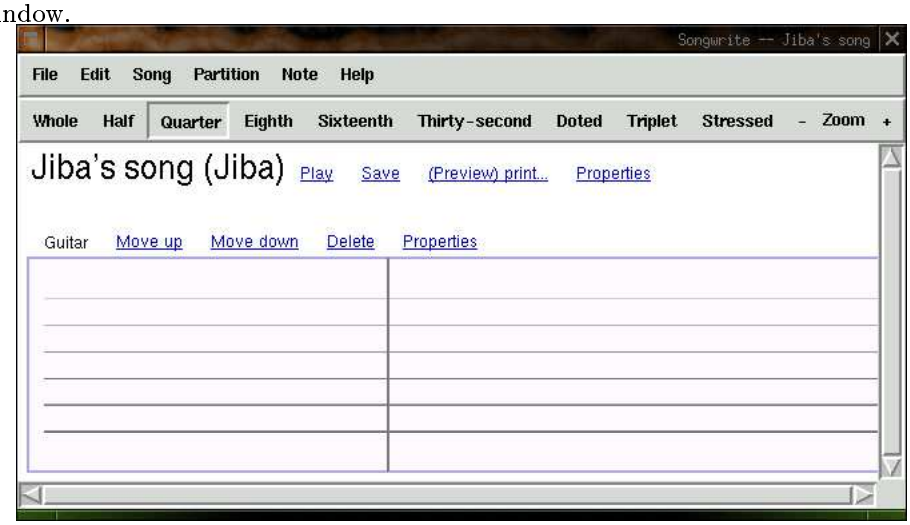
player (such as "playmidi %s" if your sound card supports Midi).

To change the configuration later, use the edit ▷ preferences... menu.

3 Creating a song

3.1 Song properties

The main document you edit in Songwrite is called a "song" (sorry if you just play guitar without singing :-). When Songwrite is launched, it creates a new song for you, and display it in a new window.



A song is composed of many partitions. The word "partitions" should be understood here in a very large meaning: it includes guitar tablatures, but also lyrics, drums,... and other (in the future).

Obviously the song has also some basic information about it, like its title or its authors. You can change them in the song properties dialog box, available through the song ▷ properties menu, or clicking on the blue link on the top of the main window. Notice that the title, the author and the comments are shown at the top of the main window.

The language property of the song is intended for lyrics; it is given to LaTeX when printing. It defaults to your current language.

The song property dialog shows your song properties. The tree view at the top of the window displays the song's structure and the different partitions; it can be used to change the currently edited object and you can click on the violet triangle to show/hide the children items. The bottom part of the window shows the properties of the current object (by default, the song).

3.2 Rhythm

The song menu offers also different playing functions, and the song ▷ rhythm menu can be used to change the tempo and the rhythm of any bar. The syncope picking option is destined to finger-picking song, and consists in playing the first half of each time two time longer than the second one.

If no notes are selected, the rhythm of ALL bars is edited. If some notes are selected, only the bars with at least one selected note in it are edited. Only x/4 and y/8 (where y = 3, 6, 9, 12,...) rhythms are supported.

3.3 Repeat / playlist

Click the song ▷ repeat... menu to get the repeat dialog box. To add an item, select the notes of the bars your want to add (at least one par bar), and click Add.... Use the Remove, Up and Down button to remove or move the playlist's items. You can also modify an item by selecting it, and changing its values, or by selecting new bars and then clicking the Do button. The first bar is #0.

Bug Only playlist which can be translated into musical symbols are correctly shown and printed.

3.4 File menu

The file menu can be used to open or save your work, and to import or export it in other format (Midi, ...). Click on file ▷ create new window to get a new Songwrite window, and edit many songs at the same time. file ▷ new file creates a new song in the current window.

3.5 Edit menu

To finish with, the edit menu allows to cancel or redo any operation, so as you can feel cool when editing tablatures ;-) Multiple cancelments are supported.

The edit ▷ preference... menu displays the configuration dialog.

3.6 Zoom

The - and + button on the right of the toolbar allows you to change the horizontal zoom level.

4 Managing partitions

4.1 Partition properties

Songwrite displays the different partitions vertically, just below the song's title and comments. When launched, Songwrite adds a new tablature to your song.

Each partition has two parts: a header (the "guitar" label) and the partition itself. The header can be used to display some comments or information about the partition, about who plays it, ... i guess you'll find it useful ;-)

Songwrite totally separates the *data* of the partition, *e.g.* the notes, and the *view* that display them, e.g. the tablature. This means that it is possible to change the view of a partition. For example, right click on the partition header, and it uses the hidden view instead of the tablature view, and collapses in a one-line bar. Click again to return to the previous view.

Click the blue "property" link on the partition header to get the partition properties dialog box, or click the partition ▷ properties menu. The dialog box allows to change different parameters, and may differ for the different views:

- volume
- muted
- chorus
- reverb
- header
- instrument: choose the Midi instrument to use for this partition
- tonality
- view: allows to change the partition's view. Mainly used to convert tablature into staff, or staff into tablature.

4.2 Adding new partitions

The partition menu offers to add new partitions for different type of instruments. Add some of them to try!

Just Click on a partition to select it. Songwrite displays a pastel blue box around the selected partition. The partition ▷ move up, partition ▷ move down and partition ▷ delete menus can be used to move or delete the current partition.

The number of partition is not limited, though Midi playing is limited to 16 channels. In Songwrite, each musical partition (*i.e.* not lyric!) takes one channel, or two if it contains notes with special effect (see below). So you can have at least 7 instruments.

5 Using tablatures

Tablatures are probably the most awaited partitions! A tablature can be added by the partition ▷ new string instrument submenus.

For each tablature, Songwrite displays one line for each string on the instrument (see below if you want more or less strings or a custom tuning). As in a real guitar, higher strings are thinner than lower one. The vertical grey line you can see in the middle is a bar. Don't bother about it, Songwrite add them automatically if you add a note after the last bar.

5.1 Adding notes

You can use the mouse (left click) or the cursor's key to move in the tablature.

To add a note on the tablature, just click were you want to add it, and type the corresponding fret number (fret number superior to 9 are OK, or even 158! Just type several figures successively, *e.g.* "1", "2" for 12). If the midi player is well configured, Songwrite will play the notes when you enter them.

You can change the note duration by clicking on the corresponding buttons in the tool-bar. The width of the selection corresponds to the duration. As accentuation is very important in guitar playing, you can also toggle accentuation with the stressed button. Stressed notes are played stronger, and displayed in red.

Notice that, when moving the selected position, Songwrite takes the current note's duration into account. If you want to move with a smaller step, select a shorter note duration.

The note ▷ arrange at fret... menu can be used to organize the selected notes on the strings so as they are played after the choosen fret. Usefull after a Midi importation!

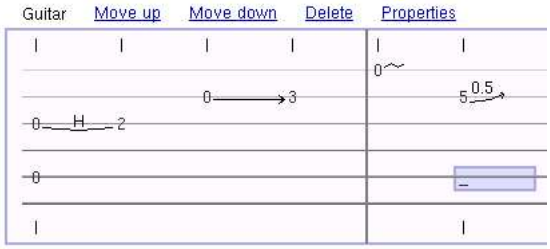
5.2 Notes properties

Double click on a selected note (or a group of selected notes), or click the note ▷ properties menu to get the note dialog box. This window allows to set different properties:

- value: the note's pitch (in semi-tons; the note's name is displayed on the left)
- duration (where 64 is a musical time unit; the note's duration name is displayed on the right)
- volume (the volume can be controlled with the accentuation button too)
- special effects (see below)

5.3 Special effects

Songwrite supports most of the special effects that can be played on a guitar: hammer/pull, slide, dead notes, bend, tremolo, roll... are all here! For effects that link two notes (hammers and slides), the effect must be set on the first note. For rolls, the effect must be set on *all* the note of the corresponding chord, and not only the lower one.

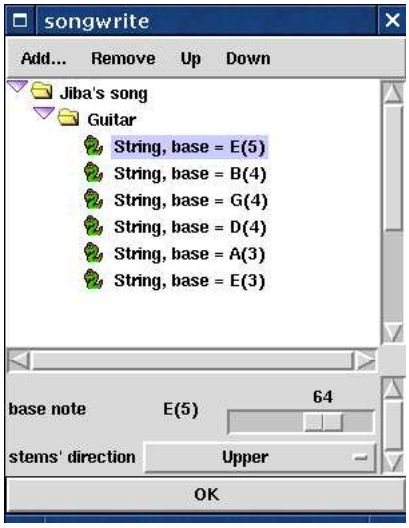


Bug It is not possible to put more than one special effect on a single note :-(

Bug Slide and hammer combination of more than 5 semi-tons cannot be played with Midi.

5.4 Tuning

The partition properties dialog box (or the song dialog box) allows to change the guitar tuning, or to add or remove strings. If you play another string-based instrument (banjo, mandolin, ...), please mail me the tuning so i can add it in the next release!



Songwrite allows also to choose for each string if the stems and the beams are drawn at the top or at the bottom of the tablature. By default, the three upper strings draw their stems and beams at the top, and the three lower ones at the bottom. To add a new string, select the partition and then click the Add... button. To remove a string, select it and click the remove button. The move up and move down button can be used to modify the order of the strings.

5.5 Keyboard shortcuts

cursor keys move the current position

tab, shift tab go the next or the previous note on the same string

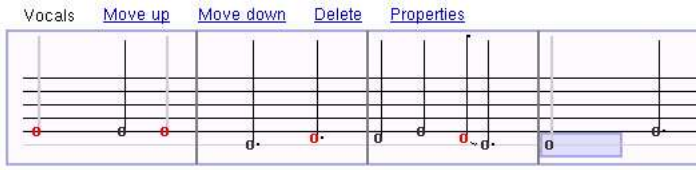
page up, down go to the previous or the next measure

origin, end go to the beginning or the end of the song

+, -	increase or decrease the note pitch
/, *	increase or decrease the note duration
.	toggle doted duration
return	toggle accentuation
del	delete all selected notes
n	normal (remove special effect)
s	slide to the following note
h	hammer, pull or legato
b	bend
t	tremolo
d	dead note
r	roll
space	plays from the current position

6 Staff

In addition to tablature, SongWrite supports staffs too. However, staff support is quite limited; it may satisfy a guitarist but probably not a pianist! The partition ▷ new staff ▷ ... menu allows to add a new staff. The partition properties dialog box allows to choose the tonality.

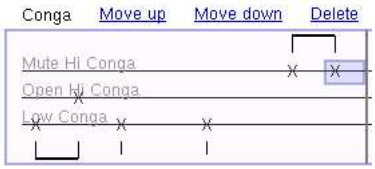


To insert notes, right-click at the right position, or use the "+" or "-" keys. These keys can also move the selected notes up or down. To add more lines above or below the staff, put a note at the top or bottom limit, and press "+" or "-".

7 Using drums

Drums sets can be added with the partition ▷ new drums ▷ new ... menus; different drums instruments are proposed. I DON'T play drums, so any suggestion is welcome. In particular, i'd like to know what are the "patch" provided by a standard drums set!

The current drums view doesn't use any "official" or standard drums notation (sorry i don't them!), but rather a really easy tablature-like system. Songwrite display a line (similar to a guitar string) for each drum patch; the name of the coressponding midi patch is written on the left of the line. Right click (or press "+") on the line to play a drum; drums are visualized as an "X" on the line. Right click again or press "-" to remove the drum.



8 Using lyrics

8.1 Vocals

First, enter the lyrics' vocals, with a staff or, why not, a tablature (if, like me, you don't know how to read a staff ;-).

8.2 Lyrics

Lyrics can be added through the partition ▷ new lyrics menu. They must be placed *just below* the melody. Typically, one would use a different lyrics set for each verse and refrain.

Then, click on the lyrics' rectangle and type the text. SongWrite automaticaly aligns the syllabes on the vocals, and moves them correctly if the melody if changed after that. The following keys has a special function:

space, tab go to the next syllabe

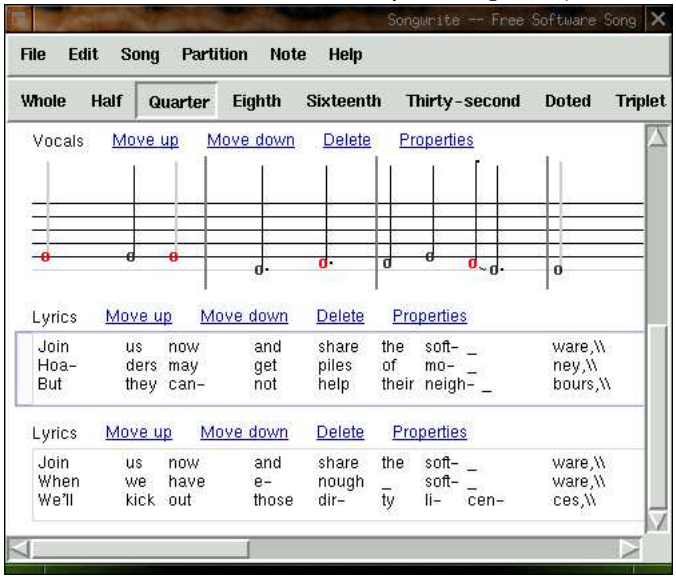
- go to the next syllabe, and cut the current word

_ go to the next syllabe, and continue the previous syllabe on the current note

two backslashes end a line

return new sentence on the same vocals. If several sentences must be sung on the same vocals, don't copy the vocals!

The following example summarizes all of that, and shows two verses of a famous hacker song (you can find it in the demo directory of SongWrite).



Finally, as the final print out is done with Lilypond and LaTeX, you can use LaTeX code in the lyrics!

8.3 Conversion

Since version 0.8, SongWrite has a new lyric system, which allows per-note lyrics instead of per-mesure lyrics. A link (in blue) propose to convert the old lyrics into the new format; however the conversion is not totally automated :-(In particular you need to cut the word manually (by typing "-").

8.4 Drums "tuning"

As guitar strings, the drums lines can be "tuned" between the different patches. The drums's view properties dialog box is very similar to the tablature one, and allows you to change the patch of each line, and to add or remove lines.

Bug Using several times the same drum patch in a drums partition can give strange results.

9 Selection and copy/paste

Songwrite has a pretty advanced copy/paste system. You can select many notes/... by drawing a box with the mouse; "heterogeneous" selection are supported, that is to say you can select notes of different partition, at the same time, in the same selection! Impressive, isn't it? Then click on

one of the selected notes/items to move the selection horizontally or vertically. Right-click when moving if you want to cancel.

Songwrite use a Linux-like copy/paste system: anything that is selected is automatically copied. Then just middle-click to paste the selection; keep the middle button pressed and move the mouse to adjust the exact pasting position.

Notice that Songwrite use the current note's duration to align the pasted notes horizontally.

10 Printing

Printing is still an experimental feature, though it can already yield impressive parts! Tablatures are printed by Lilypond, with the tablature patch i have written for it, and lyrics are printed with LaTeX, but you don't need to know Lilypond or LaTeX to print with Songwrite!

10.1 Configuration

To print your work with Songwrite, you need Lilypond and LaTeX. Lilypond is available at `www.lilypond.org`. Songwrite 0.13 has been tested with Lilypond 2.3.12. At the end of the Lilypond install, don't forget to place the TeX font intialization script, as demanded. See the end of the output of Lilypond's "make install", reproduced here:

```
*** Before running, buildscripts/out/lilypond-{profile,login}
*** must be run. You're advised to source these scripts from your
*** login scripts. For more information, see Invoking LilyPond in the manual.
```

10.2 Print!

The following things are not yet supported while printing:

- special effects other than hammers/pulls. Hammers/pulls are rendered as "normal" slurs.
- triplets without 3 notes in it (linked notes are right).
- more than 2 notes at the same time, but not in a chord (=with different duration). This should never occurs in a real tab.

11 Songbook

A song groups many songs in order to print them together or to access them more easily.

The File▷New songbook... allows to create a new songbook. Then use the Add... button to add song into the songbook (you can only add songs that are saved on the disk). Be carefull, SongWrite does not save the songs inside the songbook, but only their filename, so you must keep your song files!

Click a song in the book to open it in a new window.

To print the songbook, click the (Preview) print button.

12 Songwrite internals

12.1 Intro to dynamic programming

Songwrite is fully Python written, and use the dynamic features of the language as much as possible, as well as some technics comming from aspect-oriented programming[1]. Yes, it's full of amazing hacks, but why should i use a dynamic language if i don't use it fully? In this subsection, i try to explain why dynamic hackish coding is so great[2].

I do believe that politics, as well as the vision of the "ideal" society you have, influences your way of coding. As an anarchist hacker, it is normal for my code to be so hackish! But it does work, and it follows it own logic.

Many people try to have a global vision of what they are coding. They are wrong, because, the bigger your project is, the more difficult this approach will be! The method i propose is the following: just design a society of objects, each of which having a specific role and doing its job. And let the objects be themselves, let them having their own life. When all objects do their job, the work is done! You *do not* need to understand how the global work is performed, but only how each object does its little job. And so, if a project is twice bigger, it just requires twice as many objects, and so twice as many work for you. No more!

[1] `http://aopd.net` E.g. see how the `view` module adds a new mother class to the `song.TemporalData` class, or see the `editobj.eventobj` module (in EditObj).

[2] Hey ? What about a book about that ? Something like "the hacker hackish way of coding" ? It would be a graceful change from all those methodologic books about UML ! And maybe the first book that really learns how to code.

12.2 Data types

The `song.py` pure-Python module defines all the data for song, partitions and notes objects. It is totally independent of the rest of Songwrite and so can be easily re-used in other (GPL'ed) programs.

The `song` object has a list of partitions called `partitions`. This list contains partitions, lyrics, ... anything that can be added to a song; all these classes of objects inherits from the class `TemporalData` (*e.g.* `Partition`, `Lyrics`, ...). Any `TemporalData` is composed of a list of sub-items, typically called `notes` (for `Lyrics`, the list is called `lyrics` and aliased with `notes`); the class of these sub-items depends on the `TemporalData` class: `Note` (or subclasses) for `Partition`, `Lyric` for `Lyrics`, ...

Special effects are treated as different subclasses of `Note`. The `LinkedNote` class is used for any note that is linked with another (as a source or as a destination), and `HammerNote` and `SlideNote` are subclasses of it. Notes that are linked from another one but doesn't have any effect (e.g. the second note of a hammer) are `LinkedNote`.

Drums are just `Partition` and `Note` exactely as other. The only difference is that it use the 128th instruments, which is out the range of the standard midi instruments, and Songwrite maps this value to drums.

12.3 Views

Views are defined in the `view.py` module. Importing this module modify the `TemporalData` class so it now has a `view` property, with the corresponding methods.

The `view.py` module also defines the `View` and `GraphicNote` base classes. Songwrite use a view for each partition, and the view creates a *graphic note* for each note (or lyric, ...) the partition has. Graphic notes typically creates some Tkinter canvas item in the Songwrite canvas to represent the corresponding note.

13 How can I help you with Songwrite?

- Send me your suggestions!
- New intrument tunings are welcome too!
- Translate Songwrite to your native language: Songwrite use the standard GNU Gettext system; just translate the Songwrite.po file in the locale directory!
- Improve this documentation.
- Port Songwrite to other OS. As it now uses Tk, porting Songwrite should be really easy. The only real problem is the dependencies: Timidity and Lilypond. The former can easily be replaced by any other Midi player, but not the latter...
- Set up a web-based database of free partition for Songwrite!
- Write import/export filter for new format: this is relatively easy since you only need to look at the `song.py` module.
- Write new views. For advanced coders!
- See the TODO file for more suggestions.