Creating Song Books

This README file provides an introductory explanation of how to use the Songs LaTeX package on Windows. It is intended as a user-friendly tutorial for those not already familiar with the LaTeX document-publishing system. If you are already familiar with LaTeX or you are a Unix user, you will probably find the official package documentation more helpful than this document. The official package documentation can be found online at songs.sourceforge.net.

1 Installing

To install the song book software on Windows, run the `setup` self-installer available for download at songs.sourceforge.net.

2 Compiling the Song Book

After you've completed the installation, locate the `Songbook` folder on your desktop (or elsewhere if you changed the default directory during installation), and open the `Sample` subfolder inside. Double-click the `songs.sbd` icon. The Vim file editor window will open.

Near the top of the window you will see a row of icons. One of them (usually the fourth one from the left) will show a green arrow pointing rightward to a reddish document with the tiny letters "PDF" in it. If you hover the mouse pointer over it, the words "Save file and generate a song book pdf" will appear over the icon. Click (with the left mouse button) on that icon. (If you can't find the proper icon, you can alternatively use the menus. Open the "SongBook" menu and select "Generate".)
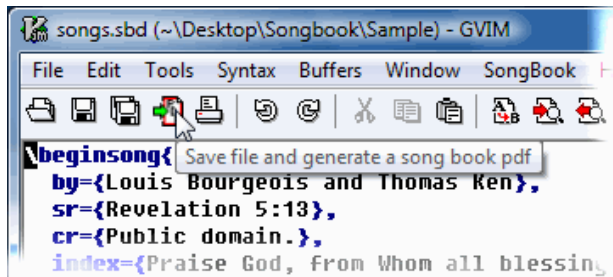
Once you've either clicked on the correct icon or selected the correct menu item, a black DOS window will open and lots of text will flow by. Hopefully it will conclude with the words "Completed successfully!" and prompt you to hit any key to continue. (If it doesn't, then that means you have edited one of the songbook files incorrectly and caused an error. See Section 5 for instructions on how to diagnose and correct errors.) Press any key to close the black DOS window and then close the Vim file editor window (by clicking on the X-button in the top-right corner of the window).

In the `Sample` folder you'll now see new files named `chordbook.pdf`, `lyricbook.pdf`, `slidebook.pdf`, and `transparencies.pdf`. Double-click on any and Adobe Acrobat Reader will open, displaying each newly created song book.

3 Common Tasks

3.1 Creating Handouts

To create handouts containing only some songs in some particular order, you must edit the `chordbook.tex`, `lyricbook.tex`, `slidebook.tex`, or `transparencies.tex` file in the `Sample` folder, depending on whether you want a handout for musicians, a handout for singers, a set of digital projector slides, or a set of overhead transparencies. Here's how:

Start by double-clicking on the icon for whichever of the above files you wish to modify. The Vim file editor program will open and you'll see a lot of LATEX programming. Find the blue-colored line near the top of the file that starts with the text "`% \includeonlysongs`". The "`%`" in front of that line tells the song book generator to ignore that line, so right now it doesn't do anything. Move the cursor down and remove the "`%`" and the space after it. The line will probably change color to brown to indicate that it is now active. Immediately after the "`\includeonlysongs`" part, you'll see a pair of braces "`{ ... }`" enclosing a comma-separated list of song numbers. Replace that list with your own list of song numbers in the order you want them to appear in the handout. Song numbers should match whatever numbering scheme your song book uses (e.g., "`85,32,128`" if it uses standard arabic numbering). Separate the numbers with commas and no spaces. For example:

> `\includeonlysongs{85,32,128}`

Next, recompile the songbook (see Section 2), and when you open the pdf file for the tex file that you modified, you'll find a handout including only the songs you listed. To go back to a full song book, reinsert the "`%`" in front of the "`\includeonlysongs`" line.
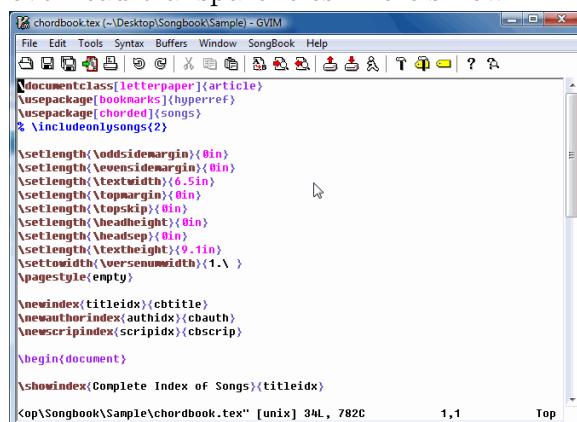
3.2 Adding New Songs

To add a new song, go to the `Sample` folder and double-click on the icon for `songs.sbd`. The Vim file editor will open and you'll see how the sample songs were entered. If you want to just jump in and experiment, you can scroll to the bottom of that file and start typing in a new song the same way you see the other songs in the file entered.

Vim will automatically color the things that you type in various ways to indicate whether it thinks they are correct or not. For example, when you first start typing the word "`\beginsong`",
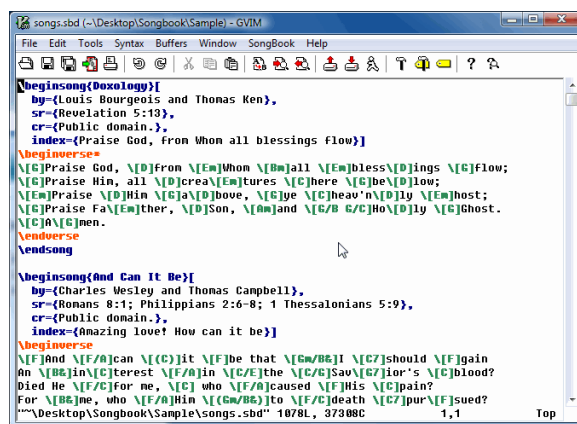
it will be in black because Vim doesn't recognize any command named "\beginso". When you finally hit the last g in "\beginsong", the entire word will change color to dark blue.

From the sample songs already in the `songs.sbd` file, you can see how to write typical songs. For a more systematic discussion of the various available syntaxes, see Section 4.

When you're done entering your new song, click on the icon for compiling the songbook or use the SongBook menu to recompile the song book (see Section 2). If no errors are reported, then open `songbook.pdf` to view your work. Otherwise see Section 5 on errors.

4   Song Entry Syntax

For most people it is easier to learn the proper song entry syntax by example than to read about it in this document. You can view examples for about twenty public domain songs by double-clicking on the icon for `songs.sbd` in the `Sample` folder. However, if you are trying to do something unusual or would like more explanation, read on.

4.1   Structure of a Song

Each song in a songbook begins with a line like:

`\beginsong{Title}[by={Authors},sr={Scripture refs},cr={Copyright}]`

This will produce a song that looks like:

The song number ("1" in this example) is inferred automatically—songs are numbered in the order they appear.

The `by={...}`, `sr={...}`, and `cr={...}` parts may be provided in any order, and any may be omitted. You may also type them on separate lines of the input file if you like; doing so won't affect the appearance of the resulting book. The "Title" part may either be a single song title or multiple song titles separated by double-backslashes (`\\`). For example,

```
\beginsong{Main Title \\ Second Title}[
    cr={\copyright~2018},
    sr={John 3:16}]
```

will produce

3

A separate index entry will be created in each of the book's indexes for each of the song's titles. The scripture index will also be affected by anything you put in the "Scripture refs" field, and the author index will be affected by anything you put in the "Authors" field. If you enter an illegal scripture reference for a song (e.g., a misspelled book of the Bible, a chapter or verse that doesn't exist, etc.) then an error message will alert you to the problem when you try to compile the book.

A song ends with the line:

`\endsong`

Between the `\beginsong` and `\endsong` lines, a number of constructs are possible in any order desired. They are...

4.1.1 Verses

Start a new verse by issuing the line:

`\beginverse`

Proceed to enter each line of text in the verse (see Section 4.2). Then end the verse by issuing the line:

`\endverse`

Any number of verses may be in a song. If you have verse numbering turned on, you can prevent an individual verse from being numbered by using `\beginverse*` instead of `\beginverse`. (Use `\endverse` to end such a verse.)

4.1.2 Choruses

Start a chorus by issuing the line:

`\beginchorus`

Enter each line of the chorus exactly as for a verse. End the chorus by issuing the line:

`\endchorus`

There may be any number of choruses in a song. In the generated song book, a chorus differs from a verse only in that it will be indented and accompanied by a vertical line in the left margin. Choruses are also not numbered.

4.1.3 Musicians' Notes

You can include textual notes that only appear in the musicians' version by entering:

`\musicnote{text of note}`

This command will have no effect upon lyric-only versions of the book, but will cause the musicians' version to contain text in a shaded box:

text of note

Musicians' notes should normally only be placed within verses or choruses or before the first verse/chorus of a song. It is okay to place them elsewhere between the `\beginsong` and `\endsong` lines, but if you do so, they can become separated from the verse or chorus they refer to by a page or column break. Never place them between songs.

There is a special kind of musicnote that provides a capo suggestion for guitarists. If you put

`\capo{1}`

in a song, it will insert the following textual note in the musicians' version, suggesting that guitarists put a capo on fret 1:

capo 1

You can replace "1" with any number up to 11, but it is unusual to put a capo on any fret above the 5th one. The `\capo` musicnote differs from a regular musicnote, though, because in a pianists' version of the book, the `\capo` musicnote will cause the chords in the song to be transposed. See Section 4.8 for how to make a pianists' version that uses this feature.

4.1.4   General Notes

Textual notes to be shown in both the lyric-only and the chorded versions of the songbook are written as:

`\textnote{text of note}`

They are exactly like musicians' notes (see Section 4.1.3) except they also appear in the lyric-only version.

4.1.5   Extra Index Entries

You can add additional index entries for the song in two different ways. To add an index entry corresponding to notable lines from the song's lyrics, add the following to the list of options in the `\beginsong` line (see Section 4.1):

`index={some short lyrics go here}`

To add an index entry corresponding to an alternate title for the song, add:

`ititle={the alternate title goes here}`

These lines will add an extra entry to every title index related to the book section in which the song is located.

4.1.6   Licensing Information

For your song book to be legal, each song that is not part of the public domain needs to be used under the auspices of a license purchased by your organization. Information about the license under which the song is being used is generally printed after the copyright information on the copyright line. You may specify licensing information for the current song by adding the following to the list of options in the `\beginsong` line:

`li={text to be added after the copyright info}`

Usually the same licensing information is used for many songs, so you will want to define a macro that abbreviates the licensing info. For example, if you own a CCLI license, then in the preamble of the `songbook.tex` file you could put

`\newcommand{\CCLI}{(CCLI \#123456)}}`

and then in each `\beginsong` line for a song covered by CCLI, you could just put
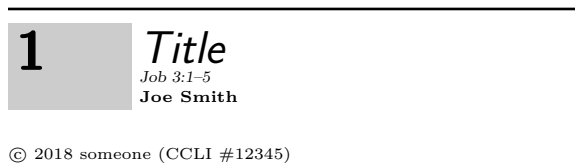
```
        li=\CCLI
```
to cause the CCLI license number for your organization to be printed.

For example, the code
```
\beginsong{Title}[by={Joe Smith},sr={Job 3:1-5},
                  cr={\copyright~2018 someone},
                  li=\CCLI]
\endsong
```
produces a song like

1　*Title*
Job 3:1–5
Joe Smith

© 2018 someone (CCLI #12345)

4.2   Contents of Verses and Choruses

The contents of each verse or chorus consists of a series of lines, each of which contains both the lyric information and the chord information for the line. The chords are the most prominent part, so let's start with those.

4.2.1   Chords

A chord is created by typing
```
\[chordname]
```
where "chordname" is the text that is placed above the line of lyrics to indicate which chord should be played. The simplest valid "chordnames" are just A, B, C, D, E, F, or G. For example,
```
\[A]
```
produces an A-chord. A-sharp and A-flat chords have the names A# and A& respectively. For example,
```
\[A&]
```
creates an A-flat chord.

In general, just about any text can go in a "chordname". Common chord names include but are not limited to:
```
\[Gm]       — G minor
\[Gdim]     — G diminished
\[Gsus4]    — G suspended-4th
\[Gmaj7]    — G major-7th
\[Gadd11]   — G with the 11th added
\[G+]       — G augmented
\[Gmaj7sus4] — G major-7 with a suspended 4th
```
It is also standard to put a slash (/) after a chord name in order to specify a bass line. For example,
```
\[G/B]      — G with a B in the bass
```

`\[Gm/B&]` — G-minor with a B-flat in the bass
`\[F#/C#]` — F-sharp with a C-sharp in the bass
`\[Cmaj7/B&]` — C major-7th with a B-flat in the bass

You can indicate that a chord is "optional" by enclosing it in parentheses. For example,

`\[(E&)]` — an optional E-flat chord
`\[(E&sus4/B&)]` — an optional E-flat suspended-4th with a B-flat bass

4.2.2 Chord-Lyric Pairing

The text that you type immediately after the closing bracket (`]`) of a chord determines what the chord is placed above in the resulting document. If lyric text immediately follows the closing bracket, then that lyric text is placed under the chord. For example,

`\[Cmaj7sus4]love, and` *produces* love,^{Cmaj7sus4} and

Notice that a long chord name causes the first space appearing after the chord to be lengthened to accomodate the chord's length.

If you type a space immediately after the closing bracket, then the chord appears between the various words that surround it but not directly above any word. For example:

`love, \[Cmaj7sus4] and` *produces* love, Cmaj7sus4 and

This can be used to indicate that the chord is to be struck between the words of the lyrics rather than on a particular word.

You can put a chord in the middle of the word to indicate that the chord should be struck on a particular syllable instead of at the start of the word. For example,

`righteous\[F]ness.` *produces* righteousness.

You can even put different chords over different syllables within the same word:

`\[Cmaj7sus4]kind\[Dm]ness` *produces* kind - ness (chords: Cmaj7sus4 Dm)

Notice that when a word needs to be split into segments to accomodate the chords, hyphenation is automatically added.

The above describes the default pairing of chords to lyrics, but sometimes you might want to override the defaults to achieve certain effects. First, you can force multiple chords in sequence to appear over a single word or syllable by using a "chord name" that has spaces in it. For example,

`\[C D7 G]over me` *produces* over me (chords: C D7 G)

Second, you can use braces (`{}`) to restrict a chord or chord sequence to a single syllable of a word, rather than allowing it to range over the whole word. For example:

`{\[C D7 G]o}ver me` *produces* o - ver me (chords: C D7 G)

7

This might be used to indicate that the chord changes are to occur while the first syllable is being sung, rather than on the next syllable as would be more typical.

Third and finally, you can also use braces ({}) to force lyric text to be included under a chord when it would otherwise be pushed out away from the chord. For example,

\[Cmaj7sus4]{th' eternal}   *produces*   Cmaj7sus4 th' eternal

In this case the words "the eternal" are intended to be sung together as three syllables instead of four. That means that the Cmaj7sus4 chord should span both words rather than pushing the second word out away from the chord as would happen by default.

4.2.3   Echo Parts and Repeated Lines

Aside from chords, there are a couple of other things in chorus and verse bodies worth mentioning. The following two methods can be used to indicate that certain lyrics should be echoed or repeated.

To italicize and parenthesize a section of lyrics, type:

    \echo{put the lyrics (and chords) here}

This is generally used for echo parts. For example:

\[G]Alleluia \echo{\[C]Alleluia} *produces* G Alleluia C *(Alleluia)*

To indicate that a line should be sung and played multiple times in a row, add the following to the end of the line:

    \rep{n}

where n is the number of times to repeat. For example, \rep{2} will produce the text "(×2)".

4.2.4   Line breakpoints

You may also want to dictate where LaTeX breaks the line if it is too long to fit. If you want the line broken in a particular place, put

    \brk

at that spot. For example,

This line \brk ends here.   *produces* This line ends here.

Note that this is different than just hitting return and starting a new line of text because \brk will indent the second half of the broken line in the generated song book to indicate that it is a continuation of the line above.

4.2.5   Measure bars

You can insert a measure bar by typing the "|" symbol within a verse or chorus. The very first "|" that appears in a song will have meter numbers placed above it. By default, the meter numbers will be "4/4" (indicating four quarter notes per measure). For example,

`|The Lord a|bove`   *produces* $^4_4$| The Lord a|bove

If you wish to change the default meter for a particular song, somewhere before the first measure bar of the song put:

`\meter{6}{8}`

for 6/8 time, etc. You can also change meters mid-song. To do so, instead of typing a "|" in the verse or chorus text, type

`\mbar{2}{4}`

to switch to 2/4 time. For example,

`\meter{6}{8}`
`|The Lord a\mbar{2}{4}bove`   *produces* $^6_8$| The Lord a$^2_4$|bove

Measure bars (produced with either "|" or `\mbar`) can appear anywhere within chorus or verse lyrics except that they should not follow a chord without an intervening space. For example:

INCORRECT:	`\[G]| above`
CORRECT:	`\[G]	above`
CORRECT:	`|\[G]above`

4.3   Conditional Text

You can designate certain verses, certain choruses, or certain parts of verses and choruses to only be displayed in the chorded version or only in the lyric-only version. To do so, write

`\ifchorded`
material only included in chorded books
`\else`
material only included in non-chorded books
`\fi`

4.4   Scripture Quotations

Between songs you can also insert scripture quotations. Scripture quotations start with a line that reads:

`\beginscripture{John 3:16}`

(or with whatever reference is desired in place of John 3:16). After that, type the text of the quotation, and end it with the line:

`\endscripture`

For example,

`\beginscripture{James 5:13}`
`Is any one of you in trouble?`
`He should pray. Is anyone happy?`   *produces*
`Let him sing songs of praise.`
`\endscripture`

> *Is any one of you in trouble? He should pray. Is anyone happy? Let him sing songs of praise.*   James 5:13

Within a scripture quotation, several constructs are available:

4.4.1 Poetry

Biblical poetry typically consists of tuplets (e.g. couplets or triplets), each line of which is called a "colon". The first colon of a tuplet, called the "A-colon", is typically typeset flush with the left margin, while the remaining colons are typeset indented. If any colon is too long to fit in a single line, the wrapped portion is doubly-indented.

You can typeset poetic verse in this style by beginning each A-colon with `\Acolon` and beginning each subsequent colon of the tuple with `\Bcolon`. For example, Psalm 4:5, which consists of one couplet, could be rendered with:

```
\beginscripture{Psalm 4:5}
\Acolon Offer right sacrifices
\Bcolon and trust in the LORD.
\endscripture
```
produces

> *Offer right sacrifices*
> *and trust in the LORD.*   Psalm 4:5

Biblical poetry is also often divided into blocks called "strophes". Each strophe is separated from the next by a small vertical space. This vertical space can be produced by `\strophe`. For example, Psalm 88:2–3 consists of two strophes, and could be rendered with:

```
\beginscripture{Psalm 88:2-3}
\Acolon May my prayer come before you;
\Bcolon turn your ear to my cry.
\strophe
\Acolon For my soul is full of trouble
\Bcolon and my life draws near the grave.
\endscripture
```
produces

> *May my prayer come before you;*
> *turn your ear to my cry.*
>
> *For my soul is full of trouble*
> *and my life draws near the grave.*
>   Psalm 88:2–3

4.4.2 Indented Blocks

Scripture quotations can also contain indented blocks of text. Within a scripture quotation, the current indentation level can be increased with `\scripindent` and decreased with `\scripoutdent`. For example, Hebrews 10:17–18 could be rendered with:

```
\beginscripture{Hebrews 10:17-18}
Then he adds:
\scripindent
\Acolon ''Their sins and lawless acts
\Bcolon I will remember no more.''
\scripoutdent
And where these have been forgiven, there
is no longer any sacrifice for sin.
\endscripture
```
produces

> *Then he adds:*
> *"Their sins and lawless acts*
> *I will remember no more."*
> *And where these have been forgiven,*
> *there is no longer any sacrifice for sin.*
>   Hebrews 10:17–18

4.5 Manual Column-breaking

To force a column break to occur in a particular place, enter the line:

```
\nextcol
```
Column breaks may only be forced between songs or scripture quotations, not within a song.

Use of `\nextcol` is not recommended, however. A song book looks better if you can fill each page with enough material so that no forced column breaks are necessary. Try to order songs and insert scripture quotations in such a way that each column is filled with material and column breaks occur naturally in aesthetically pleasing places.

4.6   Comments

Any line that begins with a percent symbol (`%`) will be ignored. Vim signals this by coloring the line blue. These comment lines can appear anywhere, including between songs, within songs, within verses, and within choruses.

4.7   Guitar Tablatures

You can put guitar tablature diagrams anywhere you can put normal text. To do so, type a line like the following:

`\gtab{C}{X32010}`   *produces*
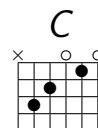
The first pair of braces should contain the name of the chord, which will be written above the tablature diagram. As with other chord names, use "`#`" for sharp and "`&`" for flat.

The second pair of braces contains a description of the contents of the tablature diagram. For each string of the guitar, from lowest pitch to highest pitch (left to right), type an "`X`" (don't play that string), "`0`" (zero or the letter O, to play that string open), "`1`" (1st fret), "`2`" (2nd fret), "`3`" (3rd fret), or "`4`" (4th fret).

You can put a fret number to the left of the diagram by putting a single digit followed by a colon at the beginning of the second argument. For example,

`\gtab{Cm}{3:X13321}`   *produces*
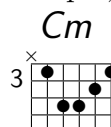
If you want to specify fingering for the chord, you can do so by adding a colon to the end of the second argument followed by six digits: a number from 1 to 4 to indicate which finger should be placed on that string, or "`\0`" (zero or the letter O) for strings on which no finger should be placed. These fingering numbers will appear below each string in the tablature diagram. For example,
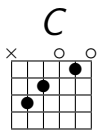
`\gtab{A}{X02220:001230}`   *produces*
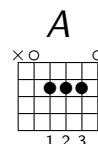
This tablature diagram shows an *A* chord in which the first string isn't played, the second and sixth strings are played open, and the third, fourth, and fifth strings are played on fret 2 and with the guitarist's first, second, and third fingers respectively.

Tablature diagrams can be placed above lyrics by using a construction like:

`\[\gtab{C}{X32010}]God loves`   *produces*   God   loves

4.8   Transposing a song

You can transpose the chords in a song automatically by placing the line

`\transpose{n}`

within the song, where n is a positive or negative integer denoting how many half-steps to transpose the song up or down. All chords in the song after the `\transpose` line will be transposed up or down by n half-steps.

You can also conditionally transpose a song by using the `\capo` line as described in Section 4.1.3. Putting the line

`\capo{n}`

at the beginning of a song normally just prints a music note that says "capo n", suggesting to guitarists that they place their capo on fret n. However, if you insert the option `transposecapos` into the list of options in the `\usepackage[...]{songs}` line near the beginning of the `chordbook.tex` file, then all capo notes are discarded in favor of actually transposing all the chords in songs that would have had capo notes. That is, if you want capo notes to be omitted and the chords of each song transposed instead, then you should change the second line of the `chordbook.tex` file to look like:

`\usepackage[ ...  chorded,transposecapos]{songs}`

This allows you to generate a songbook for pianists or other musicians who don't have capos but who wish to play along with guitarists who do.

4.9   Ligatures

For most users, the various methods described in Section 4.2.2 for creating chords will suffice to produce reasonably accurate and aesthetically pleasing song books. However, users that wish to create perfectly typeset books may want to devote extra attention to chord and measure bar placement within special character combinations called ligatures. A ligature is a sequence of letters or symbols that is grouped together and rendered as a single font character instead of a sequence of font characters. For example, in the word "difficult", the letters "ffi" form a ligature. Notice that these letters are packed so close that they almost overlap, forming a single symbol. The result is a very clean, professional-looking document. Documents written in English typically only use five ligatures: ff, fi, fl, ffi, and ffl. Other languages sometimes use other ligatures such as æ and œ.

12

Normally, LaTeX detects and creates ligatures automatically. For example, if you type "`difficult`" in your document, LaTeX will automatically detect the letters "`ffi`" occurring in sequence and transform them into an "ffi" ligature. However, if a chord happens to occur in the middle of the ligature, that process breaks down. For example, if you type "`dif\[G]ficult`", then LaTeX will produce "difficult" in the lyrics instead of "difficult". (The difference between the two is subtle, so you have to look closely to see it.) This small typesetting error will appear not only in the chorded version of the book, but in the lyric-only version as well, which is particularly unpleasant.

To work around this problem, there is an alternative method for producing chords that fall in the middle of ligatures. To typeset a chord that falls within a ligature, use the following macro:

   `\ch{chordname}{pre}{post}{lig}`

where "`pre`" is whatever part of the ligature should appear before the chord, "`post`" is whatever part of the ligature should appear after the chord, and "`lig`" is the full text that produces the ligature. For example, to typeset the word "difficult" with a G-chord above the second syllable, type:

   `di\ch{G}{f}{fi}{ffi}cult` *produces* di fficult

（上に G が付く）

Use of the `\ch` macro instead of the `\[]` macro is necessary in the above example because the word "difficult" is properly hyphenated "dif-fi-cult", so a chord on the second syllable should fall on the second "f", but that position falls within the ligature "ffi". To avoid breaking the ligature, we use `\ch` with "`f`" as the first argument (since the "f" in "ffi" falls before the chord), "`fi`" as the second argument (since the "fi" in "ffi" falls on or after the chord), and "`ffi`" as the third argument (since that text produces the unbroken "ffi" ligature).

Sometimes both a chord change *and* a measure bar will fall within a ligature. You can handle that case by doing exactly the same thing except using `\mch` instead of `\ch`. For example, to place an A-minor chord along with a measure bar on the second syllable of "affliction", you would type:

   `a\mch{Am}{f}{fl}{ffl}iction` *produces* affl - iction (with |Am above)

In the extremely unusual case that a meter change should appear on a measure bar that falls within a ligature, you can use the "`\meter`" macro (see Section 4.2.5) before the `\mch` to set the new meter. For example, to repeat the above example but with a meter change to 6/8 time, type:

   `a\meter{6}{8}\mch{Am}{f}{fl}{ffl}iction` *produces* affl - iction (with Am above)

It is worth noting that measure bars produced with "𝄋" do *not* need any special treatment when they fall within ligatures. For example,

   `af|fliction`

will work perfectly fine and will not break the "ffl" ligature except when the measure bar is actually displayed.

Remembering when you need to use `\ch` or `\mch` to avoid breaking a ligature is difficult, so the song book software provides two forms of help:

1. While editing `.sbd` files in Vim, Vim will highlight broken English ligatures in inverted blue, alerting you to the problem. For example, you will see something like

   `dif\[G]ficult`

   if you tried to use the `\[]` macro in the previous example.

2. While editing `.sbd` files in Vim, Vim's "SongBook" menu will contain the item "Correct Ligatures". Selecting this menu item will scan through your entire document and convert any broken English ligatures into proper uses of `\ch` or `\mch` automatically.

5 Error Messages

Because the syntax for song-entry is so exacting, you're almost sure to make some mistakes as you add songs. If you are using the Vim file editor to enter new songs, the vast majority of errors become immediately apparent because Vim will highlight them in inverted red, yellow, or blue. Here are what the default Vim colors mean:

violet = a scripture quotation
blue = a comment; blue text will be ignored when compiling to PDF
dark blue = the command to begin or end a song
light blue = a song command used with correct syntax
orange = the command to begin or end a verse or chorus
black = regular text (or something Vim doesn't know about)
green = a known chord used with correct syntax
brown = a measure bar
red = the beginning or end of a lyric-only or chorded-only section
`inverted red` = Vim thinks this is an error
`inverted yellow` = Vim thinks this isn't yet complete
`inverted blue` = a broken ligature (see Section 4.9)

Inverted red is usually a very good sign that you've mistyped something, although if you're doing something extremely unusual, such as using non-standard notation for chord names, you might occasionally get Vim to highlight something you've typed in inverted red even though it will produce a perfectly legal song book with no errors when the song book is generated.

Another good indication that you've mistyped something is color "bleeding". For example, if you forget to put `\endscripture` at the end of a scripture quotation, everything you type will continue to be violet, indicating that something is wrong.

However, it is possible to type errors that Vim won't catch. If this happens, then when you compile to PDF according to the directions in Section 2, LaTeX will report an error and prompt you to type something in response. Unfortunately, LaTeX errors can look very confusing at first. Let's take a look at one as an example. If you forgot to put a closing "}" brace at the end of an `\echo{...}` command, then when you performed compilation process in Section 2, you would be greeted with the following error message:

```
Runaway argument?
{Alle\D /A[lu]\A [ia!]
!  Forbidden control sequence found while scanning use of \echo.
<inserted text>
                \par
l.100 ...\A[ia!]  \echo{Alle\D/A[lu]\A[ia!]

?
```

The most important part of this error message is the part that says "`l.100`" at the beginning of the last line. This means that the error occurred on line 100 of the file. Since Vim reports the current line number of the file you're currently editing (in the bottom right corner of the editor window), this makes it easy to pinpoint where things went wrong. You can often ignore the rest of the error message and just use Vim to look around for anything amiss and correct the problem. Don't limit your attention just to the named line, however. Sometimes a typo one or two lines above the named line can cause an error to be reported on a later line.

If you can't figure it out just by looking, the error message may sometimes help. In the above case, LaTeX is reporting that something went wrong while it was looking at an `\echo` command in line 100. That's a good hint because the typo that caused the error is the absense of a "}" at the end of the echo text.

If all else fails, you can often locate an error by deleting lines from a newly added song until the error goes away. When the error disappears, you know that the last line you deleted must have been causing it, and you can focus your attention there. (Naturally you should make a backup copy of the file before doing this so that you can restore it after you've deleted enough lines to determine where the error lies.)

Whatever your approach, after an error has occurred, you must remember to do several things before you try compiling the songbook again:

1. Type "quit" and hit enter in response to the prompt. When the compilation is finished, close the black compilation window when prompted before you try to edit the file. You won't be able to save any changes in the Vim file editor window while the compilation window is open because Vim can't write to the file while LaTeX is still reading it.

2. After closing the compilation window, edit the file containing the error using Vim.

3. After correcting the file, try recompiling as per the instructions in Section 2.

4. If the error recurs, repeat this process.

6 Modifying the Songbook Structure

The high-level structure and format of each song book is defined in the files `chordbook.tex`, `lyricbook.tex`, `slidebook.tex`, and `transparencies.tex`. By understanding the format of those files, you can move sections around, add new sections, and introduce new indexes. Modifying the song book structure is not recommended unless you are reasonably familiar with the LaTeX typesetting language. For the most part the provided sample files should serve as an adequate example for anyone who has such familiarity. For a comprehensive description of all the features offered by the `songs` package, see the `Reference Manual` pdf file that accompanies this `README` file in the `Songbook` folder.

7 Uninstalling the Song Book

Installing the song book software places an uninstaller program named `uninstall.exe` in the `Songbook` directory. Executing this program (by double-clicking on it) will delete all files in the `Songbook` directory that have not been modified by the user, as well as all auxiliary files and system registry keys installed elsewhere. After uninstalling, the `Songbook` directory can be deleted manually to eliminate even the user-modified files, if desired. The uninstaller will *not*, however, uninstall MiKTeX, Vim, or Adobe Acrobat Reader. If you installed those software packages as part of installing the song book software and you wish to uninstall them, you can do so by using the "Add/Remove Programs" icon in the Control Panel to uninstall each.