LivePrompter Manual

1. What is LivePrompter?

LivePrompter is essentially a Teleprompter designed specifically for musicians. It displays lyrics and chords for a song on screen and automatically scrolls through the song, so that it always displays the current part of the song.

Being designed for live musicians, LivePrompter incorporates some special features:

- Its file format is based on the “ChordPro” format, which allows very easy entry of chords in song lyrics. Also, there is already a large pool of ChordPro files available on the Web, which you can use with LivePrompter
- Songs can be organized in “books” (music collections) and set lists
- LivePrompter is very easy to operate – in live operation, you mostly need only one key (or foot switch) to control the program. The program is also built to be easily operated with touch screens or tablet PCs.
- Scrolling through songs can be controlled very precisely via timer markers within the song text
- Songs can be transposed easily via “transpose” and “capo” functions. In transposing, LivePrompter always attempts to interpret chords in a harmonically meaningful way.
- LivePrompter can connect via MIDI to other software or to MIDI devices in order to trigger sound changes, to set up lighting or even just to synchronize multiple LivePrompter machines

The input files for LivePrompter are simple textfiles (extension “.txt”) which contain lyrics, chords, and metadata – sorry, no graphics or PDF documents, so no score display. But you can use tab or drum notation in text format.

LivePrompter has been developed for the Windows platform – unfortunately no Mac, Unix, Android or iOS version at present.

2. Installation and Configuration

LivePrompter is contained in a ZIP file; simply unpack it into a folder of your choice, and you are (almost) ready to rock! Important: since LivePrompter’s settings are controlled by an .ini file in the program folder, it is important to unpack everything into a folder for which you have write access rights (which isn’t usually the case for the “Program Files” folder). So: best to use your own folder outside the “Program Files” folder (e.g. C:\LiveTools\LivePrompter\). If you really, really want to install into the “Program Files” folder, then make sure that you give yourself write access for the LivePrompter subfolder.

The LivePrompter folder contains:

- LivePrompter.exe – the actual program file
- LivePrompter.ini – the configuration file (see below)
- graphics – a folder with button images. Best don’t change anything here!
- fonts – a folder with font files for use with LivePrompter (DejaVu Sans)

To run the program, you need a song folder – this is the folder that contains the text files for songs. Things won’t work without this, so this is the first thing we need to configure after installation!

Simply open LivePrompter.ini with any text editor (e.g. Notepad), find the line “HomeDirectory=…” and enter the path to your own song file folder without the “\” at the end. This will look like this:

```
HomeDirectory=C:\Users\Torsten\Documents\LivePrompter
```

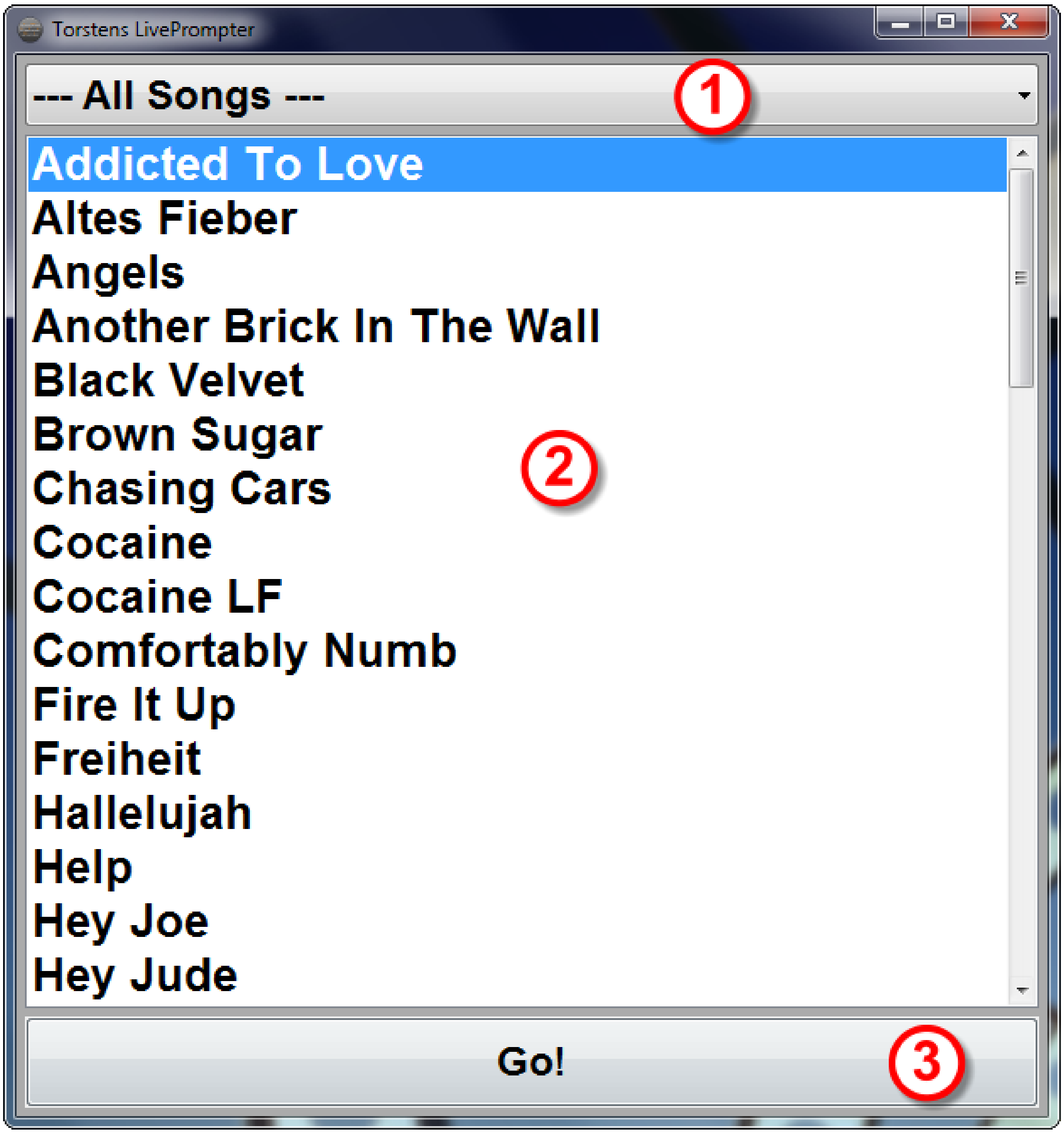
Let's leave the rest of the settings as they are for the moment – we'll deal with them later.

Now you need to install the specific fonts: simply enter the "fonts" directory, select all the font files contained within, and then right-click them and select "Install". This should install the DejaVu Sans font to your Windows fonts folder.

Now you can start LivePrompter by double-clicking the program file LivePrompter.exe – of course you can also create a shortcut to this file on your desktop or in your start menu. Creating a shortcut also allows you to start LivePrompter with different configurations – we'll get to that later as well.

3. The Main Window

After starting, you will first see LivePrompter's main window:



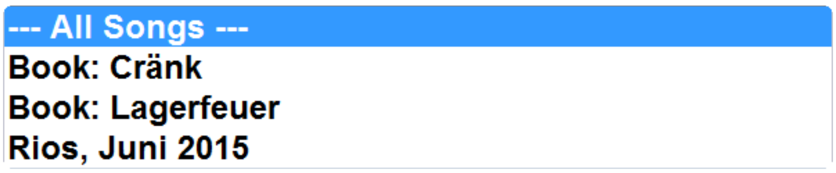
It will probably look somewhat different for you, since you haven't got any songs in your song folder yet, but you get the idea ;-)

The main window has three elements:

- The list selector for books and set lists (1)
- The song list (2)
- The “Go!” button (3)

The list selector (1) allows you to choose which songs are displayed in the song list:

- All songs in the song folder in alphabetical order (“--- All Songs ---“)
- All songs belonging to a specific book (“Book. …”) in alphabetical oder. NB: songs can belong to multiple books.
- All songs from a set list, ordered in the sequence assigned in the set list file



--- All Songs ---
Book: Cränk
Book: Lagerfeuer
Rios, Juni 2015

There will be more information on books an set lists in later chapters.

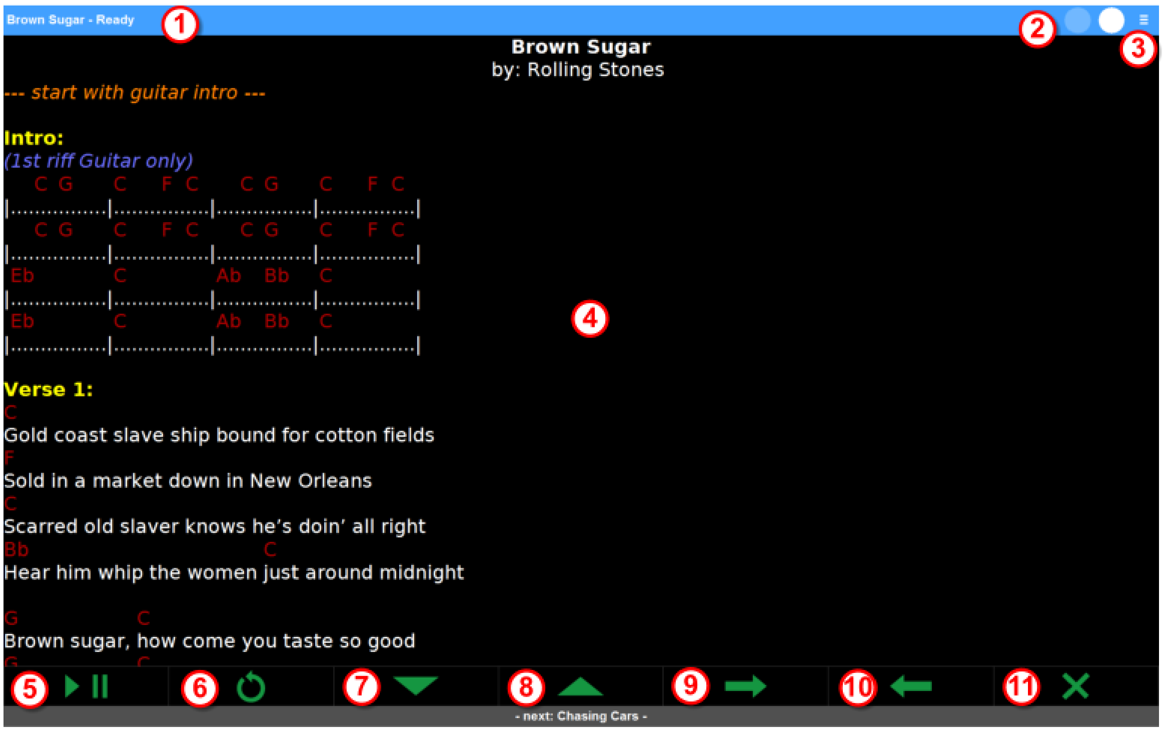
Important: choosing “--- All Songs ---“ also completely re-reads the song folder’s content. If you have created new song or set list files, you don’t need to re-start LivePrompter to display them – simply choose “--- All Songs ---“ and everything will be updated.

The song list (2) displays all songs chosen with the list selector. Clicking a song selects it (marked blue). After this, simply click “Go!” (3) and start display mode, starting with the selected song and proceeding further through the songs in the song list.

Displaying songs is always full-screen – no more program window! After exiting display mode, you’ll get back to the usual Windows display with the main window.

4. The Song Display Screen

The song display screen is all about your song’s lyrics and chords:



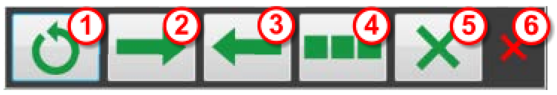
```
Brown Sugar - Ready

                              Brown Sugar
                          by: Rolling Stones
--- start with guitar intro ---

Intro:
(1st riff Guitar only)
   C G     C     F C      C G      C     F C
|...............|...............|...............|...............|
   C G     C     F C      C G      C     F C
|...............|...............|...............|...............|
 Eb             C              Ab    Bb    C
|...............|...............|...............|...............|
 Eb             C              Ab    Bb    C
|...............|...............|...............|...............|

Verse 1:
C
Gold coast slave ship bound for cotton fields
F
Sold in a market down in New Orleans
C
Scarred old slaver knows he’s doin’ all right
Bb                                  C
Hear him whip the women just around midnight

G                  C
Brown sugar, how come you taste so good

                         - next: Chasing Cars -
```

It consists of these elements:

- **The title bar (1):** it displays the name of the current song, as well as LivePrompter's operating mode. It changes colors according to the operating mode:
  - **Ready (blue):** a song has been loaded and is waiting for scrolling to be started with the play/pause button
  - **Playing (green):** a song is being scrolled
  - **Paused (red):** scrolling is currently paused, either by the play/pause button or by a command in the song file, and needs to be re-started manually with the play/pause button
  - **Countdown (orange):** scrolling is currently paused by a command in the song file, but will be continued after a certain time determined by the command. You can also manually continue scrolling by pressing the play/pause button.
  - **Finished (purple):** scrolling has reached the end of the song
- **The metronome (2):** it displays the song tempo in BPM (if entered in the song) by alternatively illuminating the two circles
- **The menu button (3):** this button allows you to call up the mini menu for the display screen. With the mini menu, you can hide or show bottom button row (you don't really need these buttons when operating LivePrompter with keys or foot switches). Also, when buttons are hidden, the mini menu gives you temporary access to the key button functions
- **The song display (4):** this area displays and scrolls the current song
- **The play/pause button (5):** it starts, pauses, and continues scrolling. Also, it allows you to continue scrolling when paused by a command.
  Don't worry if scrolling does not start immediately after pressing the play/pause button at the start of the song: at the beginning of a song, there will always be a certain delay. The reason for this is that the (virtual) song position will immediately start moving down the song from the top of the page, but only when it has reached a certain position on the screen will the actual scrolling start, otherwise the top of the song would be gone from the screen too soon!
- **The re-start button (6):** this button resets scrolling to the head of the song and also sets the operating mode to "Ready"
- **The jump-down button (7):** it scrolls down the song by a certain amount
- **The jump-up button (8):** it scrolls up by a certain amount
- **The next-song button (9):** it skips to the next song in the set list, in the current book, or in the song folder
- **The previous-song button (10):** it skips to the previous song in the set list, in the current book, or in the song folder
- **The exit button (11):** it exits display mode and closes the full screen view

4.1 The Mini Menu

The mini menu (opened by clicking the menu button) allows you to hide the bottom button row, giving you more screen space for your song when operating LivePrompter via keyboard or foot switch. If you still need to use your mouse or touch screen for some functions, you can reach them temporarily in the mini menu:



- **The re-start button (1):** this button resets scrolling to the head of the song and also sets the operating mode to "Ready"

- **The next-song button (2):** it skips to the next song in the set list, in the current book, or in the song folder
- **The previous-song button (3):** it skips to the previous song in the set list, in the current book, or in the song folder
- **The hide/show button (4):** it hides the bottom button row or shows them when hidden. This also causes the song to be reloaded and reset to start, so don't do this while scrolling through a song!
- **The exit button (5):** it exits display mode and closes the full screen view
- **The close button (6):** it closes the mini menu

4.2 Keyboard Operation

All essential functions in display mode can also be operated via the keyboard. One helpful feature is the "universal key" (default: the space bar): for different operating modes it adapts its function::

- **Ready:** start scrolling
- **Playing:** pause scrolling
- **Paused:** continue scrolling
- **Countdown:** continue scrolling
- **Finished:** go to next song

With this, you need only one key to get through normal operation during a gig (start a song, pause for an extended solo, continue scrolling after your guitarist has calmed down again, step to the next song after the end of the current one). When using a USB or Bluetooth foot switch, you can step through your gig with one switch – this allows you to concentrate more on making music and less on operating software!

Key assignments can be configured in LivePrompter.ini; the default configuration is:

- **Play/Pause:** RETURN
- **Restart:** ESCAPE
- **Jump-Up:** PAGE-UP
- **Jump-Down:** PAGE-DOWN
- **Next-Song:** CURSOR-LEFT
- **Previous-Song:** CURSOR-RIGHT
- **Universaltaste:** SPACE

5. The LivePrompter File Format

LivePrompter uses text files (extension: ".txt"). Their file name (without ".txt") is displayed in the song list on the main screen, as well as in the title bar of the song display screen. Therefore it makes a lot of sense to give your files meaningful names (why not the song title?).

ATTENTION: To display correctly, files need to be coded in 8 bit Unicode format (no byte order marker), otherwise you can get strange characters displayed, especially for special characters like "à", "ä" or other national/regional flavors.

My recommended editor is EditPad Pro (http://www.editpadpro.com) – it performs code conversions perfectly and is also a great editor for all kinds of text editing requirements. There is also a free version, EditPad Lite (http://www.editpadlite.com) that is also capable of code conversion.

5.1 Lyrics and Chords

Lyrics and chords format follows the ChordPro standard:

Lyrics are simply entered in the text file line by line. ATTENTION: LivePrompter does not do any line-breaking – anything that is too long for the screen width will be brutally truncated!

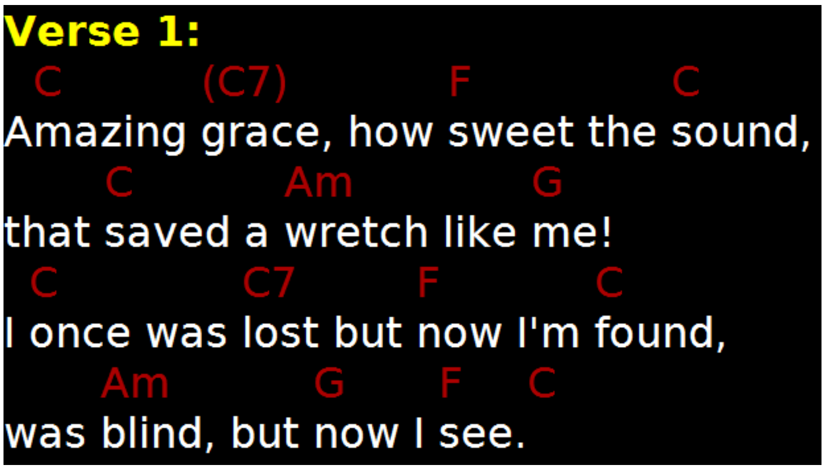
Chords are entered in square brackets at the musically correct place in the lyrics. This will display them above the lyrics at the right place

Section headings (chorus, verse, solo, …) are simply lines that end with a colon (":"). This will display them in bold and in a special color.

An example:

```
Verse 1:
A[C]mazing [(C7)]grace, how [F]sweet the [C]sound,
that [C]saved a [Am]wretch like [G]me!
I [C]once was [C7]lost but [F]now I'm [C]found,
was [Am]blind, but [G]now I [F]see.[C]
```
…

…is displayed as:



```
Verse 1:
 C       (C7)        F          C
Amazing grace, how sweet the sound,
    C        Am          G
that saved a wretch like me!
 C          C7       F        C
I once was lost but now I'm found,
    Am         G      F    C
was blind, but now I see.
```

Some specifics on entering chords, so that things go smoothly when transposing:

- Base notes are always entered in English notation (C D E F G A B C), so there is no "H", as some national conventions (e.g. Germany) notate the English "B". But: you can configure LivePrompter to display a "B" as "H" in LivePrompter.ini!
- Minor chords are entered with "m", sharps and flats with "#" and "b". So A flat minor is "Abm", F sharp major is "F#", B flat is "Bb" (watch out, you Germans!).
- Chord options are simply appended (and not affected by transposing), e.g. "C7b9" or "Dsus4"
- LivePrompter also accepts "slash chords", i.e. chords with a bass note that differs from its root, e.g. "C7/E" is a C^7 chord above an E bass note.
- Optional or passing chords can be entered in parentheses
- Unlike the standard ChordPro format, LivePrompter also recognizes multiple chords in one set of square brackets, e.g. also "[Am D7 Gmaj]" instead of "[Am] [D7] [Gmaj]".
- When a line contains only chords in one set of square brackets and no lyrics, LivePrompter does NOT create an empty line of text below it. This is convenient for instrumental parts or solos without any lyrics.

5.2 Tags – Commands for LivePrompter

To give LivePrompter instructions or meta-data, we use so-called "tags" that we enter into our song file – each in its own line! Tags will not be displayed on-screen but are used to give LivePrompter information on the song or instructions for displaying and scrolling the song.

Tags always have the same format:

`{command:data}` or `{command}`

So they are always enclosed in curly brackets and contain the actual command as well as the data needed after a colon. When the command alone is sufficient, there is no colon and no data part.

A tag always needs to stand alone in its own line, and it needs to be at the beginning of this line, so no leading spaces!

LivePrompter's tags are based on those of the ChordPro format. Today, there are a number of customized extensions of this format around, all created by various software producers using the ChordPro format. LivePrompter uses some of these extensions as well as some of its own special creations, but I have tried to stay as close to the ChordPro standard as possible, so that the song files you create stay usable in as many programs as possible.

Additionally, I have created a (admittedly rudimentary) tool called "LyricsConverter" that allows you to convert LivePrompter text files to a format optimized for other programs. Currently, this is mainly SongBook by LinkeSoft, a program that allows you to use your files on Android and iOS.

5.2.1 Song Information

These tags contain song meta-data like title, artist, key, duration, etc. This information is used to create the song header, to control scrolling speed or to transpose the song. You should enter this information at the start of the song, before entering lyrics or chords.

{title:text}	Song title
{subtitle:text}	Subtitle, often also used for artist information
{artist:text}	Artist
{book:text}	The book (the collection) this song belongs to. NB: a song can belong to multiple books – simply add multiple book-tags (each in its own line). This lets you organize your song collection by creating books for different bands or projects.
{key:text}	Song key – same format as chords (see above). The key tag is used for achieving a musically meaningful interpretation of the transposed chords (G# or Ab?). NB: multiple key tags are possible within a song (important for transposing when the root key modulates within a song. The key tag then is valid starting at its position within the text.
{capo:number}	For guitarists: capo position as a positive number (1-12). Chords will then be transposed accordingly, i.e. an A chord will be displayed as G with {capo:2}

{transpose:number}	Transposes all chords by the respective number of semitones. Positive and negative numbers are possible. NB: multiple transpose tages are possible within one song; transposition is then valid starting with the position of the tag within the song. This is pretty useful e.g. when transposing the final chorus up a semitone. Simply copy the previous chorus and precede it with the line {transpose:1}
{customcapo0:number} or **{ccp0:number},** **{customcapo1:number}** or **{ccp1:number},** **…** **{customcapo9:number}** or **{ccp9:number}**	Capo settings for individual band members (where multiple band members use the same files). In addition to the general capo settings with {capo:…}, each band member can select in LivePrompter.ini which custom transpose and capo settings will be displayed (UseCustomTranspose). Simply distinguish by number, e.g. {cc0:…} for the electric guitarist, {cc1:…} for the acoustic guitarist, etc.
{customtranspose0:number} or **{ct0:number},** **{customtranspose1:number}** or **{ct1:number},** **…** **{customtranspose9:number}** or **{ct9:number}**	Transpose settings for individual band members (where multiple band members use the same files). In addition to the general transposition settings with {transpose:…}, each band member can select in LivePrompter.ini which custom transpose and capo settings will be used (UseCustomTranspose). Simply distinguish by number, e.g. {cc0:…} for the electric guitarist, {cc1:…} for the acoustic guitarist, etc.
{tempo:number}	The song tempo in beats per minute (BPM) – this is used by the metronome NB: for slow songs, it can be useful to double the BPM, i.e. to count eighth notes.
{duration:time}	Duration of the song in the format minutes:secons, e.g. 02:03. The duration is used to calculate scrolling speed when not using time markers (see below)

5.2.2 Text Structure

These tags mark special segments of the song

{start_of_chorus} oder **{soc}**	Marks the begin of a chorus section. Various programs use this tags in different ways to highlight the chorus – LivePrompter can optionally display the chorus in bold type. The ChorusBold setting in LivePrompter.ini (see below) allows you to make a chorus bold or not. NB: LivePrompter's companion program "LyricsPrinter" uses these tags for more extensive highlighting of chorus section (indent, colored background). If you intend to occasionally print your songs, it can be useful to incorporate soc and eoc tags in your songs, even when not using ChorusBold.
{end_of_chorus} oder **{eoc}**	Marks the end of a chorus section. {soc} and {eoc} should never occur alone, always in pairs – otherwise you'll create chaos…

{start_of_tab} oder **{sot}**	Marks the beginning of a section of guitar tabulature or drum notation in text form. These section are displayed in fixed width font, so that notation is displayed correctly.
	
	Example:
	`e |-------------0-0-0-0-0-0-----0-------0-0-0-0-0---|`
	`B |-------------1-1-1-1-1-1-1h3p1p0h1-----1-1-1-1-1-|`
	`G |-----0h2-----2-2-2-2-2-2-----2-------2-2-2-2-2---|`
	`D |-0h2-------2-2-2-2-2-2-2-----2-----2-2-2-2-2-2---|`
	`A |---------0---0-0-0-0-0-----------0---0-0-0-0-0---|`
	`E |-------------------------------------------------|`

{end_of_tab} oder **{eot}**	Marks the end of a tabulature section. {sot} and {eot} should also always occur in pairs.
{hide_chords} oder **{hc}**	Following this tag, chords will be hidden, only lyrics will be displayed. This tag can be deactivated in LivePrompter.ini – then chords will always be displayed. NB: hide_chords can be pretty useful, when a song is well-known and its chord structure is simple. Simply insert an hc-tag after the first verse, then only lyrics will be displayed, and no need to delete all the chords from the text.
{show_chords} oder **{sc}**	Following this tag, chords will be displayed again after having been hidden with an hc-tag

5.2.3 Sidebar

LivePrompter allows you to split the song display into two areas: a left one that scrolls through the song and a (smaller) right one that stays fixed, independent of the left area's scrolling. This area can be used to e.g. have the chords fixed on the right side of the screen while scrolling the lyrics on the left.

When there are no sidebar-related tags in the song, the full screen will automatically be used; when there are some in the text, the screen will automatically be split for this song.

The sidebar will not be scrolled – when content is too long for the sidebar, it will be cut off at the bottom.

{start_of_sidebar} oder **{sos}**	Everything following this tag up to the next end_of_sidebar tag will not be displayed in the song text area, but in the sidebar on the right. Multiple sos-eos-pairs are possible within a song; content will be aggregated together
{end_of_sidebar} oder **{eos}**	End of sidebar content

5.2.4 Comments

These tags insert comments or instructions into the text that are neither lyrics nor chords. There are general comments (always displayed) and custom comments that are selectable in LivePrompter.ini. This allows you to have instructions for multiple musicians in the same song file, with each individual musician only viewing the common ones and the ones relevant for him/her.

{comment:text} oder **{c:text}**	General comments / instructions (all band members). Comments are formatted different from normal song text and are easier to distinguish from lyrics.
{customcomment0:text} or **{cc0:text},** **{customcomment1:text}** or **{cc1:text},** **…** **{customcomment9:text}** or **{cc9:text},**	Comments / instructions for individual band members. In addition to the general comments, each band member can select in LivePrompter.ini which custom comments will be displayed. Simply distinguish by number, e.g. {cc0:…} for the drummer, {cc1:…} for the bassist, etc. Custom comments can have individual display colors assigned to them.

5.2.5 Scroll Control

These tags control scrolling far more precisely than is possible with the duration tag. With the duration tag, an average scrolling speed is calculated for the song that ensures that the song position will reach the last line at the end of the song. But since not all parts of the song necessarily will have the same time per text line (think of a 2 minute guitar solo that consists simply of one line in the song text), the actual scrolling position will be somewhat inaccurate. Therefore, at some positions in your song, your song text will be ahead or behind.

With d_time-tags, you can set a precise time when a certain position in the song should be in the "focus area" of the screen (which is by default one third of the screen down from the top). LivePrompter will calculate scrolling speeds accordingly, speeding up scrolling for segments that contain more lines than others for the same time.

Yes, calculating or measuring these times is something of a chore, but the scrolling precision you can achieve with this method gives you a great comfort level on stage. Now you only have to correct for your drummer being a bit faster or slower on the day, which you can easily do via the jump-up/jump-down buttons when needed.

{d_time:number}	The number gives the time position of the following text line within the song in seconds. When there is at least one d_time tag within a song, LivePrompter does not use song duration for calculation scrolling speed but the time between the song start and the d_time tag, or respectively the time between two consecutive d_time tags. Song duration is then only used to calculate scrolling speed from the last d_time tage to the end of the song.
{pause:number}	Pauses scrolling for the indicated number of seconds and puts LivePrompter in Countdown mode.

	The time will be deducted from the total duration, so that scroll speed stays correct for the rest of the song.
{pause}	Pauses scrolling and puts LivePrompter into Paused mode. Also, timekeeping for d_time and duration gets frozen, so that scrolling speed stays the same.

5.2.6 Text Size

These tags allow you to change text size for the current song. Normally, text size will be defined in LivePrompter.ini, but occasionally it can be useful to display a song in larger or bigger type (e.g. when a song has extra-long text lines that would otherwise be cut off).

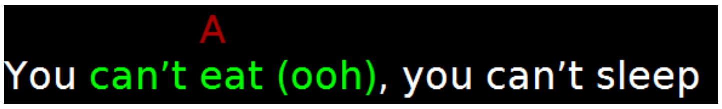
{textsize:number}	Defines main text size as a percentage of the size set in LivePrompter.ini with 100 being 100%. i.e. a setting of 80 scales all text to 80%
{sidebarsize:number}	Defines text size for the side bar (also percentage)

5.3 Text Highlights

To highlight individual words or phrases within a text line, simply enclose them in “>” and “<”. Chords within this segment will still be formatted normally. This way:

```
You >can’t [A]eat (ooh)<, you can’t sleep
```

will be displayed as:



This highlighting this way will only work within one line – don’t try this across multiple lines!

For multiple lines, you can use the tags `{start_of_highlight}` and `{end_of_highlight}` (or `{soh}` and `{eoh}`) – as usual, these tags need a line for themselves – they will highlight all text between them.

5.4 Commenting Your Song Files

Text lines that start with “#” will not be displayed on screen. You can use this to comment your song file code

```
#This line is simply a comment
```

You can also use this to temporarily deactivate tag lines without completely deleting them

```
#{transpose:2}
```

6. Set Lists

Set lists are used to define a certain song sequence, for a live performance or a rehearsal, so that the respective song files are displayed in this sequence. Using the next-song button or the universal key, you can then easily step through your set list.

Set lists are also text files (extension ".txt") that are created in the folder "Setlists" within your song folder. The name of a set list file is also the title of the set list and will be displayed in the list selector of the main window.

Within a set list file, every line stands for either a song file or a text to display (e.g. "End of first set", "Tell a dirty joke here", …).

- When the content of a line is the same as the file name of a song file (without ".txt"), the song file will be displayed
- Otherwise, instead of a song file, the content of the line will be displayed

6.1 Set List Extensions

With extensions, you can customize a song's setting without changing the actual song file. To do this, you give additional commands after the song file name in the let list file:

- **Capo:** adapts all chords for the given capo position: Cx (where x stands for the capo position). Any capo tags in the song file will then be ignored.
- **Transpose:** all chords in the song will be transposed by the given amount (positive or negative): Tx (x stands for the number of semitones). The given amount will be added to any transpose tag within the song
- **Chords:** changes the DisplayChords-Setting for the song and thus how LivePrompter deals with show_chords/hide_chords tags
  - CH+ all chords will be displayed
  - CH- no chords will be displayed
  - CH0 show_chords/hide_chords determine chord display

You can also use multiple extensions – simply separate them by commas within the curly brackets!

This function can be pretty useful, especially when multiple musicians share the same song files

- A song is played in a different key for a guest singer
- One of the musicians plays the song on a guitar with a capo. For this, he only needs his own set list file, but can still use a shared song file
- A singer plays guitar in some songs, but not in all. In her playlist, she can select for which songs to display chords

This is what a set list file can look like:

```
----- Set 1 -----
Working On It {c4,t2}
Jumping Jack Flash {ch-}
Hey Joe {ch-}
I Come In Peace {c2}
```

```
Let It Rain
```

7. The Configuration File: LivePrompter.ini

LivePrompter's settings are stored in traditional .ini files. This has proven pretty robust and useful; not only has it saved me from having to code a settings GUI, but it also allows you to use multiple different ini-files, selected at program start. Maybe you want to use LivePrompter with different song folders (one for every band) or with different display configurations (laptop screen vs. projector) and accordingly text sizes?

Just create a copy of LivePrompter.ini, make your changes to this new file and pass it to LivePrompter as a start parameter (C:\MusicProgs\LivePrompter\LivePrompter.exe CustomSettings.ini).

The format of LivePrompter.ini is simple:

```
Setting=value
```

At the beginning of the line, you type the name for the specific settings (upper/lower case IS important!), then, WITHOUT A SPACE INBETWEEN, an "=", and after that, ALSO WITHOUT A SPACE, the value for this setting up to the end of the line.

Colors for text are given in hexadecimal HTML notation, e.g. #FFFF00 for yellow. See:

```
http://www.w3schools.com/html/html_colorvalues.asp
```

Now the individual settings:

7.1 Basic Settings

HomeDir	Folder for your song files, without an ending "\", e.g. `C:\MusicData\LivePrompter`
StartWithoutButtons	When `true` or `yes`, the bottom buttons will be hidden on start (can be un-hidden using the mini menu); when `false` or `no` ,they will be shown
HeaderSize	Values from 1 to 3. You can increase the size of the title bar of the lyrics window (including the metronome) by setting HeaderSize to 2 or 3. Normal size is 1
FooterSize	Values from 1 to 3. Set the size of the footer bar (next song). 1 is normal size, 2 and 3 increase the size

7.2 Text Display

TextSize	Text size in pixels
TextColor	Text color for lyrics in HTML hex format
SectionHeaderColor	Text color for section headers in HTML hex format
CommentColor	Text color for general comments in HTML hex format
CustomCommentColor0 **…** **CustomCommentColor9**	Text color for custom comments in HTML hex format. One line for every custom comment number

HighlightColor	Text color for highlighted text (>Text<) in HTML hex format t
HideComments	When `true` or `yes`, general comments will be hidden, when `false` or `no`, they will be shown
ShowCustomComments	Selects which individual comments will be displayed – enter a number sequence that contains all custom comments to be shown e.g. `ShowCustomComments=127` → CustomComments 1, 2, and 7 will be shown
ChorusBold	Should the chorus (contained in {start_of_chorus} and {end_of_chorus}) be displayed in bold type? If yes, then `yes`, otherwise `no`
CleanEmptyLines	When `true` or `yes`, multiple consecutive empty lines will be reduced to one. This comes in handy when using HideComments and/or Hide Chords, which will sometimes create a number of empty lines. This setting is `true` by default.

7.3 Chord Display

ChordColor	Text color for chords headers in HTML hex format
ChordsBold	When `true` or `yes`, chords will be displayed in bold type, when `false` or `no`, in normal type
GermanChords	When `true` or `yes`, “B” chords will be displayed in German style (“H”), when `false` or `no`, in normal English format (“B”)
ChordsAboveText	When `true` or `yes`, chords will be displayed in their own line above the lyrics (default setting), when `false` or `no`, then within the song lyrics
HideChords	When `true` or `yes`, all chords will be hidden, when `false` or `no`, chords will always be displayed. When set to `auto`, LivePrompter will initially display chords, but obey hide_chords and show_chords tags.
UseCustomTranspose	Selects which individual transpose settings (CustomCapo or CustomTranspose) will be used. Similar to ShowCustomComments, enter a number sequence that contains all custom transpose instructions to be used. Normally, you’d probably only use one number.

7.4 Scroll Controll

AutoScroll	When `true` or `yes`, songs will be scrolled according to their duration or to their d_time tags. When `false` or `no`, songs will not be scrolled automatically, but can be manually scrolled by key presses in fixed amounts.
ScrollToMarker	When `true` or `yes`, manual scrolling via the jump-up or jump-down buttons will scroll to the next d_time tag, when `false` or `no`, always by a fixed amount
HalfPageAmount	This defines by what percentage of the screen height the screen will be scrolled by manual jumping with the jump-up / jump-down button. The amount is given by an integer number, with 50 meaning 50%.
FullPageAmount	This defines by what percentage of the screen height the screen will be scrolled by manual jumping with the universal key when AutoScroll is false or for songs without a duration or d_time tag. The amount is given by an integer number, with 80 meaning 80%.

ScreenFocus	This defines the position where the current line in the song should be on the screen (it shouldn't be all the way at the top, otherwise it would be gone too soon…). Typically you would choose a point in the top third of the screen. The position is given as an integer number, with 25 meaning 25%, i.e. the current line is always at a vertical position one quarter of the screen down from the top.

7.5 Keyboard Control

This defines the keys for song control and the universal key. At present, LivePrompter accepts the following keys (the curly brackets are obligatory): {SPACE} – die Leertaste

- {RETURN} – the space bar
- {LEFT} – the left-arrow cursor key
- {RIGHT} – the right-arrow cursor key
- {UP} – the up-arrow cursor key
- {DOWN} – the down-arrow cursor key
- {PGUP} – the page-up key (page↑)
- {PGDN} – the page-down key (page↓)

KeyUp	The key for the jump-up button
KeyDown	The key for the jump-down button
KeyPrevSong	The key for the previous-song button
KeyNextSong	The key for the next-song button
KeyReset	The key for the re-start button
KeyPausePlay	The key for the pause/play button
KeyUniversal	The universal key

8. LivePrompter MIDI control

OK, so why would I want MIDI on a teleprompter?

There are a couple of scenarios:

- Controlling LivePrompter's buttons (Play/Pause, Next Song, Up, Down, …) via MIDI keys or pedals (e.g. from your master keyboard) without the need to have direct access to a keyboard or a touch screen – pretty useful for live musicians that have their hands full
- Selecting LivePrompter songs from another program, e.g. a sequencer that you use for your playbacks
- Selecting programs / patches etc. on a MIDI keyboard, an effects device or a light controller when selecting a new song in LivePrompter
- Controlling a MIDI device at the same time as pressing LivePrompter's keys, e.g. to synchronize song switches (next song)

Personally, I built MIDI support to synchronize LivePrompter with my favorite VST host, Cantabile (www.cantabilesoftware.com) – any time I select a song in LivePrompter, Cantabile is automatically switched to the same song, so all my sounds are ready to play immediately. Alternatively, if Cantabile is your main program, LivePrompter can also follow Cantabile by sending a program change to LivePrompter on loading a song.

Essentially, there are two MIDI control mechanisms in LivePrompter:

- Sending and receiving program changes: on loading a song, LivePrompter can send a program change associated with the song. Reversely, when LivePrompter receives a program change, it will automatically load the corresponding song. The mappings between program changes and songs are (again) controlled by simple text files.
- Sending and receiving note and controller commands: every time a LivePrompter key like Pause/Play, Next Song, Up, … is pressed, LivePrompter can send out a MIDI command, either a MIDI note or a specific controller. Reversely, LivePrompter can react to MIDI notes or controllers to remotely press any of the LivePrompter keys, similar to keyboard control, just over MIDI. You could also use this feature to remote-control multiple instances of LivePrompter (for multiple musicians) from one device via MIDI.

Using rtpMIDI (http://www.tobias-erichsen.de/software/rtpmidi.html), you can even do this wirelessly – I use an 8’’ Windows tablet on my master keyboard to wirelessly remote-control Cantabile on my laptop at the back of the stage!

8.1 MIDI configuration

MIDI settings are made in LivePrompter.ini:

MidiInPort	The name of your selected Input MIDI port. If you are unsure about what the exact name is, use the small utility “MidiEnum”, included in the LivePrompter package, which lists the exact names of all MIDI input and output ports in your system
MidiInChannel	The MIDI channel that Liveprompter should react to. Set it to a number from 1-16 for a specific channel or simply set it to `omni` to react to all channels
MidiOutPort	The name of your selected Output MIDI port
MidiOutChannel	The MIDI channel that Liveprompter will send on. Set it to a number from 1-16.
MidiOutKeys	Set this to `true` or `yes` to activate sending out commands on any key press in LivePrompter. If set to `false` or `no`, LivePrompter will not send out any keys
MidiOutPC	Set this to `true` or `yes` to activate sending out program changes on loading a song in LivePrompter. If set to `false` or `no`, LivePrompter will not send out any program changes.
MidiInKeys	Set this to `true` or `yes` to activate receiving commands (note on or control change) for key presses in LivePrompter. If set to `false` or `no`, LivePrompter will not react to any commands
MidiInPC	Set this to `true` or `yes` to activate receiving program changes to load a song in LivePrompter. If set to `false` or `no`, LivePrompter will not react to program changes.
MidiInKeyUp	The MIDI command that will activate the jump-up button. Commands are defined in the following form: CC x (with x being a number from 1 to 127)

	N x (with x being either a number from 0 to 127 or a note descriptor like C#1 or A-1 with C4 for "middle C") So typical valid commands are "CC 7", "N C3" or "N 64" To trigger a key, you need to send a note on with a velocity of > 0 or a controller value of >0
MidiInKeyDown	The MIDI command that will activate the jump-down button
MidiInKeyPrevSong	The MIDI command that will activate the previous-song button
MidiInKeyNextSong	The MIDI command that will activate the next-song button
MidiInKeyReset	The MIDI command that will activate the re-start button
MidiInKeyPausePlay	The MIDI command that will activate the the pause/play button
MidiInKeyUniversal	The MIDI command that will activate the universal key
MidiOutKeyUp	The MIDI command that will be sent by the jump-up button. The format is the same as for incoming commands.
MidiOutKeyDown	The MIDI command that will be sent by the jump-down button
MidiOutKeyPrevSong	The MIDI command that will be sent by the previous-song button
MidiOutKeyNextSong	The MIDI command that will be sent by the next-song button
MidiOutKeyReset	The MIDI command that will be sent by the re-start button
MidiOutKeyPausePlay	The MIDI command that will be sent by the the pause/play button
MidiOutKeyUniversal	The MIDI command that will be sent by the universal key
MidiAlwaysSendBankSelect	If "yes", LivePrompter will always send Bank Select messages for all Program changes´. If "no", Bank Select messages will only be sent when a different bank than the current one needs to be selected.
MidiChangeToAllOnPC	If yes, then on receiving a program change for a song that isn't in the currently selected setlist or book, LivePrompter will try to find it in the "All Songs" list and, if found, switch to the "All Songs" setlist.

8.2 Mapping songs to program changes

LivePrompter needs to know which program change will activate which song or which program change to send when a song is loaded. I considered building this into the song files themselves, but found it easier to manage with a separate file that keeps all the information organized in one place.

So this mapping is defined in two files that reside in a subdirectory of your song folder called MIDI which contains two text files, called "MidiReceive.txt" and "MidiSend.txt". To avoid issues with international characters, be sure that they are coded as UTF-8 (same as your song files).

Each of them has the same layout: one line for each mapping that looks like this:

```
{Bank}.{Program},{SongName}
```

or simply

```
{Program},{SongName}
```

Since MIDI program changes are restricted to 128 different programs, LivePrompter also allows multiple banks of programs to break this limit. MIDI-wise, this means that a program change is preceded by two bank select commands, namely “bank select MSB” (CC 0) and “bank select LSB” (CC 32), which together tell the recipient of the messages that it should call up the respective program in the addressed bank.

To keep things simple, I’ve avoided the mess with MSB and LSB and simply put in a combined “bank” number.

So this is what a MidiSend.txt file could look like:

```
0.1,Addicted To Love
0.3,Angels
0.4,Another Brick In The Wall
0.5,Behind Blue Eyes
0.7,Brown Sugar
```

Or alternatively:

```
1,Addicted To Love
3,Angels
4,Another Brick In The Wall
5,Behind Blue Eyes
7,Brown Sugar
…
```

if you only have 128 or less songs in your list

8.2.1 Sending Program Changes

When loading a song, LivePrompter will scan through the list of program changes defined in “MidiSend.txt” and try to find a song name that is identical to the file name of the song being loaded. If it finds one, it will send the respective program change (and bank select commands when necessary).

This will even work for playlist entries that don’t reference a “real” song file – entries like “--- End of Set 1 ---”. If you enter this as a song name in “MidiSend.txt”, Live Prompter will also send the respective program change when you get to this point in your setlist. Nifty, huh?

By default, LivePrompter will only send bank select messages when changing banks, i.e. when the song being loaded is in a different bank. But in case you want to be extra sure (e.g. to avoid conflicts with program changes from other sources that will change the current bank), you can also force LivePrompter to always send bank select commands with every program change. Simply put this line in your LivePrompter.ini:

```
MidiAlwaysSendBankSelect=yes
```

8.2.2 Receiving Program Changes

When LivePrompter receives a program change, it will scan through the list of program changes define in “MidiReceive.txt” and try to find an entry with the correct program change. In this, it will also respect any bank select commands received previously and only look with program changes with this bank.

Once it finds a correct program change / bank combination, it will look if its current setlist contains the name referenced. If yes, it will load the corresponding song. It will even automatically open the song window if it isn't open yet.

Note: LivePrompter will by default only try to find the program in the currently loaded setlist or book. If it doesn't find the song there, it will NOT change the song. But you can also tell LivePrompter to behave differently: put the following line in your LivePrompter.ini:

```
MidiChangeToAllOnPC=yes
```

Now, if LivePrompter doesn't find a song in the currently loaded setlist or book, it will search the full list of files in the song directory. If it finds a song there, it will automatically set the list to "all songs" and switch to the song selected by the program change.

OK, that's all – have fun networking LivePrompter via MIDI!

9. Windowed Mode and Multi-Screen setup

9.1 Windowed Mode vs. Full-Screen Mode

By default, LivePrompter displays its lyrics and chords in full-screen mode; no window borders, no resizing… This is by design: a prompter should be just a prompter, no distractions, no multitasking.

But since some users want to run LivePrompter alongside other applications, I have implemented a "windowed" mode. If you add the following line to LivePrompter.ini, song display will open in a resizable window:

```
FullScreen=no
```

To avoid nasty side-effects, the main window will be hidden as soon as a song is displayed.

Just be aware of one thing: if the song window loses the focus (when you activate a different window), the key commands will not work anymore, so don't be surprised when this happens. Simply click on the song window and everything will be back to normal.

9.2 Saving Window Positions

On exiting, LivePrompter saves the positions of the main window and (if run in windowed mode) of the song display window. When starting again, both windows should remember their previous positions.

For those users working with varying screen setups, you can have different screen layouts (positions of both windows) by using multiple .ini files to start LivePrompters with and changing the setting of "ScreenLayout".

```
ScreenLayout=1
```

You can choose any number to select a different "slot" to save your window positions to (except 0, which is the default layout). So, maybe you have one configuration for a double-screen setup in your studio and one for your laptop on the road. Now you create two .ini files: studio.ini and mobile.ini, which are identical, except for different values for "ScreenLayout". Simply start Cantabile with these .ini file. The easiest way to do this: create a shortcut with the target:

```
"C:\path to prompter\LivePrompter.exe" studio.ini
```

Now, clicking this shortcut will launch LivePrompter with the multi-monitor layout defined in your studio (you define a layout by simply positioning your windows to your likings and closing LivePrompter).

This also works for those of you who want to run multiple instances of LivePrompter in parallel (maybe displaying your lyrics with chords and comments for yourself, while only showing lyrics on your second screen for a sing-along event). Simply launch each instance with a different .ini file with a different value of ScreenLayout – now both will have their window locations saved individually.

9.3 Forcing Window Positions

If you want to be absolutely sure that a LivePrompter window always appears at a certain position and size, you can override any saved settings and specify the window positions and sizes in LivePrompter.ini using ScreenX, ScreenY, ScreenW, ScreenH for the main window and SongX, SongY, SongW, SongH for the song display window.

9.4 Multi-Monitor Setups

Multi-monitor setups are pretty common nowadays, especially with musicians. So, LivePrompter has ways to customize its display for these setups:

- In its default full-screen mode, LivePrompter will display its song window on the same monitor that contains its main window. So, if you move your main window to a different monitor, the song window will then display there.
- You can change this behavior by defining a specific monitor for the full screen view to show on, independent of the main window (maybe you'll want to select songs on your main screen and display the song on your secondary window. In this case, put "`FullScreenDisplay=x`" in your .ini file, with x giving the display number of the selected monitor. You can find the correct number in the Windows display properties ("Identify").
- In windowed mode, simply push your main and song window to whatever monitor you want them to display on. LivePrompter will remember the positions and restore them on startup
- When you change your monitor configuration (e.g. unplug your second screen on your laptop) and one of the windows suddenly finds itself outside the viewable range, LivePrompter will (on its next start) simply relocate the window to your monitor #1.

Here are all the configuration settings for monitor magic in summary:

FullScreen	If `yes`, the song window will show on the full screen, if `no,` the song window will be a normal, moveable and sizable window.
ScreenLayout	This allows you to have different saved window layouts. Simply put a different number (not 0) in any configuration (custom .ini file) and the size and position of windows will be saved and recalled separately
FullScreenDisplay	In multi-display configurations, specify the number of the display that you want the full-screen song window to show on.
ScreenX	Force horizontal position of the selection window by specifying the x coordinate in pixels here
ScreenY	Selection window y coordinate (top to bottom; top = 0)
ScreenW	Selection window width
ScreenH	Selection window height

SongX	Song window x coordinate
SongY	Song window y coordinate
SongW	Song window width
SongH	Song window height