

# The chordbars Package

S. Kramm

December 30, 2018 - release 1.1

## Abstract

This Tikz-based musical notation related package is targeted at guitar / bass / piano / whatever players that are playing "popular music" accompaniment. They usually need only the chords and the song structure. This package produces rectangular song patterns with "1 square per bar", with the chord shown inside the square. It also handles the song structure by showing the bar count, and the repetitions of the patterns.

Package home page: <https://github.com/skramm/chordbars>

## 1 Motivation

This type of accompaniment notation is used when you don't need the melody, but you do need the exact chord and bar count. In that case, the full musical sheet is useless, although it can be used to print the chords. So some people like to write down the requested song/chord structure in a graphical view (see below). To produce these, some people use GUI software such as word processors, but this has a lot of drawbacks. The aim of this package is to have a L<sup>A</sup>T<sub>E</sub>Xway of producing these, with minimal effort.

## 2 Usage

### 2.1 Basic usage

This package provides a single environment, named `chordbar`, that has 2 mandatory arguments. The first argument is the number of bars, and the second is the pattern name. The latter one can be empty. Inside this environment, you can use several commands, the most basic one is `\chordf` (for "Chord Full bar") that has as unique argument the chord name. This argument will just be printed out "as is", usually something like Cm7, Db9, FMaj7, etc.

For example, the classical 12-bar blues progression will be written as:

```
\begin{chordbar}{12}{Classical 12 bar E blues}
\chordf{E} \chordf{A}
\chordf{E} \repeatBar
\chordf{A} \repeatBar
\chordf{E} \repeatBar
\chordf{B} \chordf{A}
\chordf{E} \chordf{B}
\end{chordbar}
```

This will be rendered as:

### Classical 12 bar E blues

E	A	E	•/•
A	•/•	E	•/•
B	A	E	B

Each square stands for a 4-beat bar, and this is read left to right. The special symbol you can see in bars 4, 6 and 8 means "play the same content on this bar as the previous bar".

As usual with L<sup>A</sup>T<sub>E</sub>X source files, spaces and line-feeds are ignored.

The default behavior is to have 4-bars long lines, but you can change this anytime by giving a different value to \NumberOfBarsPerLine. For example, issuing the command

```
\def\NumberOfBarsPerLine{3}
```

just before the above source code will print out the same 12-bars blues as above, but in a rather awkward view with 6 bars per line<sup>1</sup>.

### Classical 12 bar E blues

E	A	E	•/•	A	•/•
E	•/•	B	A	E	B

## 2.2 Handling song structure and bar count

This package can also be used to print the song structure by showing the number of repetitions of each part and by counting the bars. The number of repetitions of the part can be given as an optional argument to the environment. To enable this, you need to activate this option, with this inserted after the \begin{document} :

```
\countbarsYes
```

Then, this:

```
\begin{chordbar}[3]{4}{A}
\end{chordbar}
\begin{chordbar}[4]{B}
\end{chordbar}
```

---

<sup>1</sup>Don't do this, of course!

will give the following output, correctly printing out '13' as the initial bar number of the second part (part 1 is 4 bars long and is repeated 3 times).

### Part 1: A



### Part 2: B



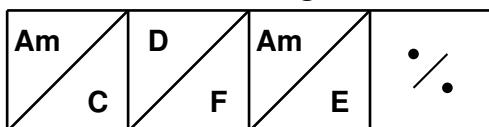
## 2.3 Two chords per bar.

The command `\chordh` enables printing two chords per bar, as this happens quite often. Its two arguments are the two chords of the bar. For example, the well-known tune "House of the Rising Sun"<sup>2</sup> chord structure can be printed as this:

```
\begin{chordbar}{4}{House of the Rising sun}
\chordh{Am}{C}
\chordh{D}{F}
\chordh{Am}{E}
\repeatBar
\end{chordbar}
```

This will be rendered as:

### House of the Rising sun



## 2.4 Sharps and flats.

If you know L<sup>A</sup>T<sub>E</sub>X, you may know that the # character is a "reserved character" and as such you shouldn't be able to use it inside your source file. However a special trick has been used here so you can directly type `\chordf{C#}`.

Warning: this **may** lead to some problems in "some" situations, that are not yet clearly identified<sup>3</sup>. If you encounter an issue, please report it on the home page of this package and post an MCVE<sup>4</sup>.

The alternate solution is either to escape the # or to use the `\sharp` and `\flat` symbols, that have been "textified" so you don't need to enter math mode. This has the advantage of being also a bit "prettier", although maybe less readable (?). You can see below the compared renderings of the two alternatives:

---

<sup>2</sup>The Animals

<sup>3</sup>See <https://tex.stackexchange.com/a/467566/11083> for details.

<sup>4</sup>Minimal, Complete, and Verifiable Example

```
\begin{chordbar}{4}{Sharps and flats}
\chordf{E\#}
\chordf{E\sharp}
\chordf{E\b}
\chordf{E\flat}
\end{chordbar}
```

### Sharps and flats

E#	E\#	E\b	E\b
----	-----	-----	-----

## 2.5 Long parts on same chord

In some songs, you often have several bars on the same chord, say as an introduction or a bridge between two parts. Instead of filling a whole chordbars environment, a command is provided that will simply render the desired number of bars with the initial chord noted in first bar.

For example, if you need a 8 bar intro on Gm7 chord, you can do this:

```
\chordline{C\#m7}{8}{Intro}
```

that will be rendered as this:

### Intro

C\#m7	•/•	•/•	•/•	•/•	•/•	•/•	•/•
-------	-----	-----	-----	-----	-----	-----	-----

Of course, the part and bar count will take this into account. Please note that this will always be a single line, so it is limited to standard situations (4 or 8 bars), it should not be used for longer parts.

## 2.6 Half bars

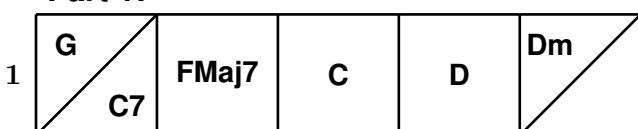
In some tunes, there can be a "special" bar inserted at one point that only has 2 beats instead of the regular 4-beat bars . This is handled with the special command \addHalfBar that needs to be inserted in last position in a chordbar environment.

See for example this chord progression and its rendering:

```
\begin{chordbar}{4}{}
\chordh{G}{C7} \chordf{FMaj7}
\chordf{C} \chordf{D}
\addHalfBar{Dm}
\end{chordbar}

\begin{chordbar}{4}{}
\chordh{G}{C7} \chordf{FMaj7}
\chordf{C} \chordf{D}
\end{chordbar}
```

### Part 1:



## Part 2:

6	G C7	FMaj7	C	D
---	---------	-------	---	---

"Part 1" has an extra 2-beats bar added at the end, and the bar count is adjusted to take this into account. So "Part 2" starts at bar 6.

### 2.7 Song title

The package provides a `\songtitle` command that will format nicely the data given in the preamble with the classical `\title` and `\author` commands.

## 3 Configuration

Several commands allow to customize the way the grids are printed out.

- The size of the grids can be adjusted with the internal variable `\barsize`. The default value is 1.6, but you can adjust it anytime inside your document, for example with:  
`\def\barsize{1.8}`

The font used for the chords can be adjusted by redefining the variable `\chordFontSize` with classical L<sup>A</sup>T<sub>E</sub>X font specifications. The default value is `\normalsize \bfseries`. If you need better readability when extending grid size, you can add for example this, in the preamble of the document:

```
\def\chordFontSize{\large\bfseries}
```

- The command `\countbarsYes` enables counting the bars of the song: each grid will have printed on the left side the number of the first bar of the grid. It also enables printing the number of repetitions of this part on the right side of the grid.

This command is useful in the sense that this package can be used in two ways: it can provide the whole structure of a song. In that case, it is useful to have for each part the number of repetitions and the bar count, so that when the band leader says "lets start again at bar 75", everybody can find it easily.

On the other side, this package can be used also to provide a quick way to show the harmony of the different parts, without any structure or bar count. This, printing the bar number becomes useless.

To stop this behaviour (anytime), the command is `\countbarsNo`.

- Additionally, if the bar counting is enabled, then the package can compute the total number of bars and the duration of the song. This is done by issuing the command `\printNbBars` at the end of the file. The duration of the song depends on both the *tempo* of the song, expressed in BPM, and the number of beats per bar. The latter can be given with the command `\bpm`, for example: `\bpm{120}`.

The number of beats per bar is limited at present at two values, 3 or 4, with the two commands `\bpbfour` or `\pbpthree`. The default value is 4 beats per bar.

- As you have seen above, the command `\repeatBar` will be rendered as "./" and means "repeat last bar". In some situations, it can be useful to show a repetition spanning two bars. For this you can use `\repeatBarPair`. For example, see the source below showing a classical II-V-I pattern played twice and its rendering:

```
\begin{chordbar}{4}{}
\chordh{G}{C7}
\chordf{FMaj7}
\repeatBarPair
\end{chordbar}
```



- `\resetchordbars`: this will reset the bar and part counters, useful if you want to print two songs in the same document.

## 4 Reference

This section is useful for those who want to contribute or expand this package.

Internal counters:

- `NumMeasure`
- `NumPart`
- `CurrentBarInLine`
- `CurrentBar`
- `CurrentLine`

## 5 Troubleshooting

The relative sizes of the grid and the chord printing are not automatically adjusted. Thus, in some case, the chord name will be too large for the grid. In that case, you just need to enlarge the grid size and/or adjust the font size (see section 3).

## 6 Release history

- 2018-12-28: First release (v1.0) and upload to CTAN
- 2018-12-29: Release 1.1, added commands `\chordline` and `\addHalfBar`, added font commands, fixed manual, bugfixes