

# Quick Start Guide to using Command Completion with T<sub>E</sub>XShop

Herbert Schulz  
herbs2@mac.com

2016/12/26

## Introduction

L<sup>A</sup>T<sub>E</sub>X markup is rather wordy which is nice because it describes what it's supposed to do but a bit painful to write. Command Completion allows you to insert complete environments and commands with a few keystrokes and the press of a “trigger” key (this is Esc by default but can be changed to Tab in TeXShop → Preferences → Source → Command Completion Triggered By:).

Commands that have arguments usually have a Mark (•) inserted for each argument. You move to the next argument by using the Source → Command Completion → Marks → Next Mark command (Ctl-Cmd-F [or Opt-trigger]). This also selects the Mark so typing automatically removes the Mark and substitutes the typed information. See the complete documentation, with lists of commands/abbreviations supplied with T<sub>E</sub>XShop out of the box, in the ~/Library/TeXShop/CommandCompletion/ folder for much more information.

## Usage

Many commands and environments can be easily completed either by using the start of the command or a set of abbreviations to the commands and environments. The following are examples of just a few to get you started.

While the list of abbreviations is fairly extensive just learn a few that you need all the time and slowly pick up others as you need them. Also just press the trigger key multiple times to get the same commands and environments with differing numbers of options, etc.

## Command Completion

You can complete many commands by starting to type them and pressing the trigger key. Variations on the commands with differing numbers of optional arguments are generated by additional presses of the trigger. One example: typing \sec on a new line and then the trigger key produces

```
\section{•}
```

while a second press of the trigger gives

```
\section*{•}
```

the \*-variant of the command and a final press of the trigger gives

```
\section[•]{•}
```

with the optional argument.

## Abbreviations

Besides completions for partial command insertions there are also many abbreviations. These are short mnemonics for complete substitutions.

All abbreviations for environments start with a 'b'. To generate a complete `itemize` environment place `\bitem` on a line by itself and press the trigger key to get

```
\begin{itemize}
\item
•
\end{itemize}•
```

with an extra Mark at the end so you can easily jump to the end of the environment. Additional items can be generated by typing `\it` and the trigger to get

```
\item
•
```

ready for entry of text.

In addition to the `\section` command lower level sectioning commands have abbreviations. Sub-sections can be generated by typing `\ssec` and the trigger to get

```
\subsection{•}
```

with subsequent presses of the trigger key giving the \*-variant and finally the variant with the optional argument.

As a final example `\tt` and the trigger gives the `\texttt{•}` command and a second press of the trigger gives the declaration `\ttfamily` with similar results for other font changing commands.

## Comments

It is easy to remember the arguments for commands that are used fairly often but forget them for those rarely used; these are the perfect candidates for comments. An example is the order of the arguments for the `\rule` command; type `\rul` and the trigger key to get

```
\rule{•<width>}{•<height>}
```

and an additional press of the trigger key gives

```
\rule[•<lift>]{•<width>}{•<height>}
```

the version with the optional argument. Another example is the `wrapfigure` environment, from the `wrapfig` package, which has multiple versions with differing numbers and positions of optional arguments. To see the variations with the comments type `\bwr` on an empty line and press the trigger key to get:

```
\begin{wrapfigure}{•<placement: r,R,l,L,i,I,o,O>}{•<width>}
•
\end{wrapfigure}•
```

with the versions with optional arguments on succeeding presses of the trigger key.

## But Typing \ is Difficult

Some keyboard localizations make it difficult to type \ directly; e.g., it takes multiple keystrokes to do so using the French keyboard localization. Hope isn't lost! In most cases an abbreviation or start of command doesn't have to start with a \ but rather any 'white space character' (i.e., the start of a fresh line, a space or tab). So instead of

```
\sec
```

and the trigger key to produce

```
\section{•}
```

you can use

```
sec
```

at the start of a line and the trigger key will produce the same completed command.

Similarly, writing tt and the trigger will give \texttt{•} since it is preceded by a space character. However 'tt will *not* work since the tt *isn't* preceded by a 'white space character'; in that case you will have to use '\tt. The simplest way to make that easier is to create a macro that does nothing by insert a \ and assign it to a simple Cmd based keystroke.

## Hey, the examples didn't work!

If these examples don't work you probably need to let T<sub>E</sub>XShop update the ~/Library/TeXShop/CommandCompletion/ folder; simply delete that folder from ~/Library/TeXShop/ and restart T<sub>E</sub>XShop.