TEXShop's key bindings vs. macros vs. command completion

Herbert Schulz

Abstract

A workshop was held at TUG'15 about some features of TEXShop that are sometimes confused. The following article is based on the topics discussed, but we'll start with some general information about TEXShop.

1 Introduction

TEXShop is a "front end" for TEX on Mac OS X. As such it allows the user to create and edit TEX source files, interact with an installed TEX distribution (e.g., typeset the source file) and finally preview the final PDF file. It also allows the user to go back and forth between preview and source.

Over the years TEXShop has added many features. Some of them are obvious and are meant to help a novice get started. Others are a bit more subtle in their use and the underlying power of these features needs to be coaxed out.

2 Three features

There are three features of TEXShop which often get mixed up: Key Bindings (at one time called Auto-Completion), Macros, and Command Completion. Although they share similar features it is possible to discern a difference between them and decide when each is most useful.

Key Bindings assign a sequence of keystrokes to the press of a single key; e.g., typing '_' produces '`_{...|}`' (where `...|` is any selected text followed by the insertion point — the place where newly-typed text is inserted) or typing '≤' (Opt-, with the English keyboard layout) produces '`\leq`'.

Macros can also insert simple text and be given a keyboard shortcut (that always uses Cmd plus other keys) but are most useful when attached to AppleScript programs so they can do special processing of source text, etc.

Command Completion (*NOT* to be confused with Auto-Completion) allows you to type a partial command or short abbreviation and, when a trigger key is pressed (Esc by default but it can be set to Tab), have the text so far expanded into a full command or even a full environment structure.

Details follow.

3 Key Bindings

Key Bindings are enabled in general by checking TeXShop → Preferences → Source → Key Bindings. Of course they can be turned off by un-checking that preference setting. They can be turned on/off for a given editing session by using the Source → Key Bindings → Toggle On/Off menu item (a check mark means it's on).

You can see view, edit, add to, and remove from the list of Key Bindings by starting up the Key Bindings Editor via the menu item Source → Key Bindings → Edit Key Bindings File....

3.1 Notes on Key Bindings

- Inserting a \ before typing the key negates the expansion; e.g., pressing \ and then _ produces _, as it should.
- Key Bindings cannot be created for characters produced by multi-key sequences; e.g., Opt-e *followed by* e produces é on the English keyboard layout which *cannot* be set to produce `\'e` as a Key Binding. The initial Opt-e is usually called a dead key since it doesn't produce an on-screen character by itself and *must* be followed by another character.
- When creating a Key Binding using `#SEL#` or `#INS#` in the replacement text will place any selected text or the insertion point at that location respectively.
- Let's emphasize that Key Bindings *always* expand to the assigned text (unless escaped with \). There may be times when you don't wish an expansion of that keystroke. You should create a Macro to do that for you.

4 Macros

The Macros menu contains a fairly large number of pre-defined macros. You can create another macro with the Macro Editor, via Macros → Open Macro Editor. Macros can be added either by copying text into a newly created macro or by adding a macro file (with extension plist) using Macros → Add macros from file (only visible and available when the Macro Editor is active).

The order of appearance of the macros in the Macros menu can also be changed by simply moving them around on the left panel of the Macro Editor.

4.1 Notes on Macros

- Text to which you wish to assign a Cmd-based keyboard shortcut is best created using a Macro rather than a Key Binding; e.g., there is already a Macro that takes selected text and sets it in boldface (using `\textbf`) and you can assign the Cmd-B keyboard shortcut to that Macro in the Macro Editor.

- When creating a text macro using `#SEL#` or `#INS#` in the replacement text will place any selected text or the insertion point at that location respectively.

5 Command Completion

For Command Completion you enter a partial command name or a short abbreviation, press a trigger key (Esc or Tab, mentioned above, if set in TeXShop → Preferences → Source → Command Completion Triggered By:) and it gets expanded. E.g., enter

```
\sec
```

on a new line and press the trigger (Esc or . . .) and you get

```
\section{•}
```

(where • is a selected bullet — called a Mark in Command Completion parlance — so simply typing will replace that Mark with your text. There can be more than one match for a given input; if you press the trigger again (without entering text) you get

```
\section*{•}
```

and another press of the trigger gives

```
\section[•]{•}
```

for separate section titles in the toc and the document. In the last case there is a second Mark (•) for the second argument. After entering the toc section title you jump to and select the next Mark by using Source → Command Completion → Marks → Next Mark (Ctl-Cmd-F) so you can immediately start typing the section title for that document.

Better yet are abbreviations. E.g., type

```
\benu
```

(abbreviations for environments always start with a 'b') and press the trigger key to get

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

. . . ready to enter the first item.

Then to get a new `\item` simply type

```
\it
```

on a new line and the trigger to get

```
\item
•
```

To get to the very end of the enumerate environment use Ctl-Cmd-F to select the Mark at the end of the environment where simply typing Return will remove that mark and move to the next line.

5.1 Notes on Command Completion

- Command Completion *replaces* any selected text by the expansion. This is unlike Key Bindings and Macros, which can be written to include the selected text in the final result of their actions.
- The easiest way to learn how to create your own command completions or abbreviations is to examine the Command Completion File using Source → Command Completion → Edit Command Completion File. . . .
- When creating a completion or abbreviation for Command Completion you cannot use `#SEL#` since any selection will be replaced by the completion. Using `#INS#` in the replacement text will place the insertion point at that location respectively. You can also use two copies of `#INS#` and any text between them will be selected; e.g., this is useful for first arguments which should be a selected Mark, •, created with `#INS#•#INS#`. Use Cmd-8 to produce the Mark because Key Bindings, if active, will replace a typed • with `\textbullet`.

6 In closing

The current version of this document (labeled as TeXShopFeatureConfusion), and many other TEX- and TEXShop-related items, are available at the DropBox url below. The TeXShopTips document there may be of particular interest.

◇ Herbert Schulz
herbs2 (at) mac dot com
https://dl.dropboxusercontent.com/u/10932738/index.html