TeXShop Tips & Tricks

v0.8–2018/09/02

H. Schulz
herbs2@mac.com

Contents

1 Introduction

TeXShop is a "Front End" for a TeX distribution on Mac OS X. As such it allows the user to create and edit TeX source files, interact with the 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.

Note: Starting with TeXShop 3.99 the Source tab of TeXShop → Preferences has been split into Source and Editor tabs. This document will use the notation of TeXShop 3.99 and later; if you are using earlier versions of TeXShop items in the Editor tab are found in the Source tab.

1.1 What Isn't Here

This article is, first of all, *not* about TEX or LATEX. I don't intend to teach you how to write TEX source. There are many fine books and articles that will teach you how to become a TEXpert or, at least, a TEXpätzer like me.

Although there is some introductory material it is also *not* meant as a complete manual for the use of TEXShop for the total novice. Over time it might evolve into such a document but I've got to start somewhere and this is that start.

1.2 What Is Here

In this article I hope to introduce you to some of the more subtle things you can do to make your life as a TEX source editor easier. These include adding keyboard commands and extending the editing capabilities of TEXShop; helping you make short(er) work of creating documents, etc., with the use of Macros and Command Completion; and, finally, how one can extend the processing capabilities of TEXShop using Engines.

2 Editing, Typesetting and Viewing — the Work Cycle

This is about as close to a beginner's section you will get in this document.

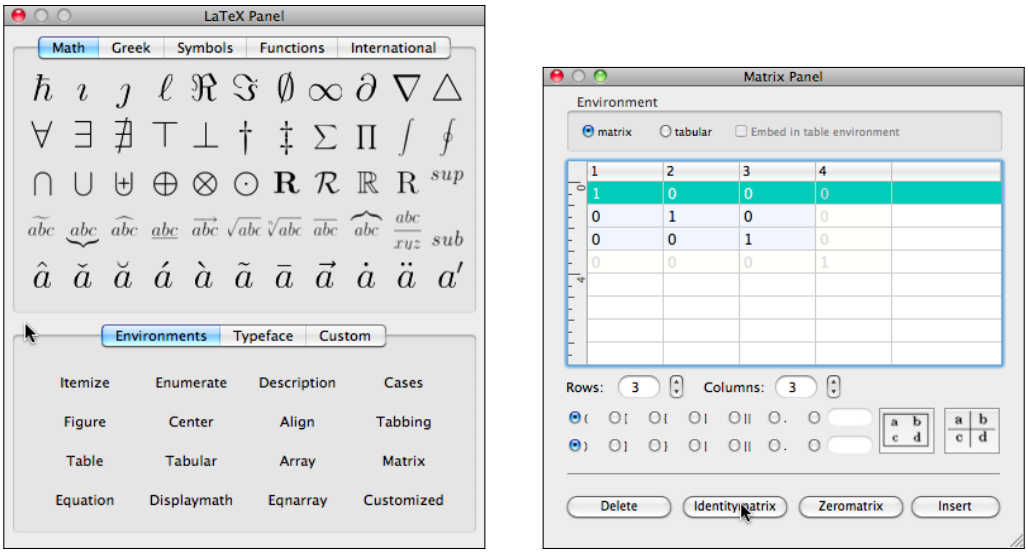
The usual cycle for producing a document with TEXShop is: first, edit the Source document, entering necessary codes for the typesetting phase; second, typeset the edited document and; third, examine the resulting pdf file. You may have to return to editing the document after attempting to typeset if errors are detected during the typeset phase. You will almost always cycle through these stages multiple times.

2.1 Editing the Source File

The first thing you've got to do to create that great work is to type it into the source document that will be typeset and viewed later. This involves both putting LATEX markup as well as your wonderful words into the document.

To get started you can open a new document using File → New (Cmd-N) and then fill in the start of a new document by choosing a template from the Templates popup menu in the Source Window or use the File → New From Stationery... command and picking appropriate Stationery from the list. Note that the templates and stationery provided are certainly not complete; if you have some that you think are of general use feel free to submit them for inclusion in TEXShop. You can add personal Templates and Stationery to ~/Library/TeXShop/Templates and ~/Library/TeXShop/Stationery respectively. **Note: In TEXShop 3.58 and later you can use the TeXShop → Open ~/Library/TeXShop Menu item to open that folder in Finder. Note: ~/Library is the Library folder in your HOME folder; *not* /Library, the Library folder at the root of your Hard Drive. Note: Under Mac OS X 10.7 and later the Library folder is "hidden" by default; in Finder hold the Opt key down and click on the Go menu and it will be available. Under OS X 10.9 and later you can *permanently* show ~/Library in your HOME folder by opening and selecting your HOME folder, choosing View → Show View Options (Cmd-J) in Finder and then checking Show Library Folder.**

Stationary is meant to be a skeleton for a complete, new document while a Template can be added at any point in a document so may just contain fragments that may be useful as additions to certain documents; e.g., specific entries for certain packages that may only be needed in a particular document. On the other hand, a Template may also contain the skeleton for a complete new document for a particular use; that's the way I tend to use them.



(a) (b)

Figure 1: *(a) The LaTeX Panel; and (b) Matrix Panel.*

2.1.1 LaTeX & Matrix Panels

While I believe that panels with a clickable interface actually hinder learning I'll mention that TeXShop has two panels: one to help with entering LaTeX code (the LaTeX Panel) and one for setting up the basic structure of a matrix or tabular (the Matrix Panel). These are toggled on/off under the Window Menu. Figure (1) shows what the panels look like.

It is possible to make a few changes and additions to the LaTeX Panel by editing the ~/Library/TeXShop/LatexPanel/completion.plist file. **Note: all plist files must be edited using UTF-8 Unicode encoding.**

2.1.2 The Tags Popup

The Tags popup menu on the Source Toolbar will automatically list sectioning commands so you can quickly jump to a relevant part of your document source. You can add your own tag to the list at a particular place in the document by placing the line

```
%:my tag name
```

at that position and it will then appear in the popup list so you can jump to that location quickly. See Figure (2). Sorry, tags are not recursively included for files you `\include` or `\input`.

2.1.3 Find/Replace

There are three Find/Replace "panels" available with TeXShop 3.xx (two with TeXShop 2.xx). Each is discussed individually below. You choose the Find/Replace Panel you wish to use on the Source tab under TeXShop → Preferences. You must restart TeXShop to enable any changes made to the Find/Replace Panel choice in Preferences.

Apple Find Panel The traditional Apple Find/Replace panel. A simple to use panel for finding and replacing text. The Standard Apple Find/Replace Panel is shown in Figure (3a) on page 4.

OgreKit Find Panel An advanced Find/Replace panel that supports Regular Expressions (regex for short) of various styles (press the More Options button to select the dialect). Regex is a

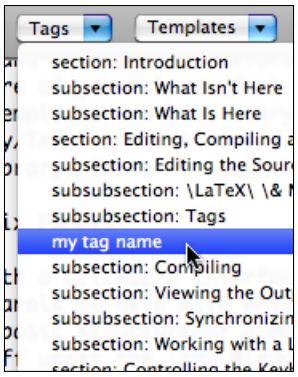
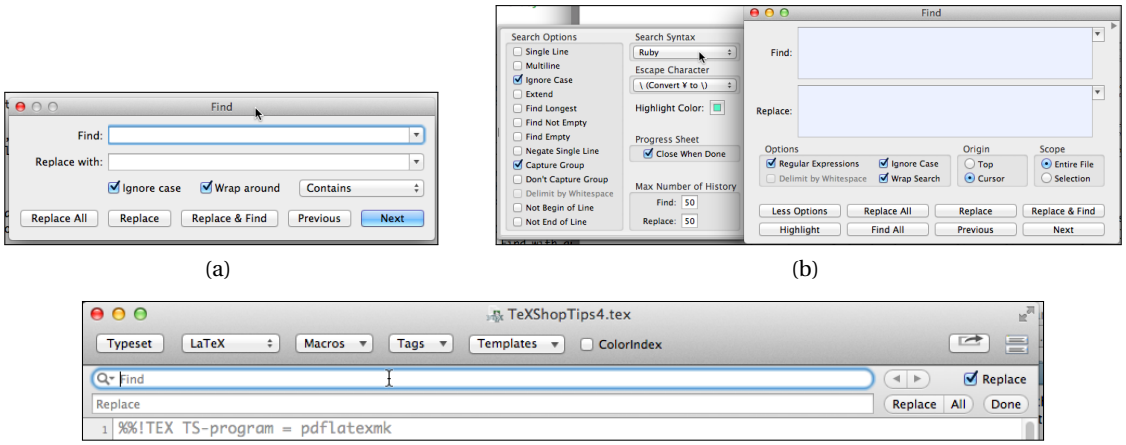


Figure 2: *The Tags Popup Menu.*



(a) (b)

(c)

Figure 3: *The Find/Replace "panels" available in TEXShop: (a) the standard Apple Find Panel; (b) the Ogrekit Find Panel with the More Options panel displayed; and, (c) the Apple Find Bar available with TEXShop 3.xx.*

very advanced way to find and replace text and is a good investment of your time to learn. The OgreKit Find Panel with the More Options panel displayed is shown in Figure (3b) on Page 4.

Apple Find Bar Only available in OS X 10.7 and later; and therefore only in TEXShop 3.xx. It provides a drop down bar for doing Find with an additional line if you select Replace. See Figure (3c) on page 4 for an example of the Apple Find Bar; the additional Replace line is displayed.

2.1.4 Spell Checking

By default TEXShop allows you to use Apple's Spell Checker as built into most applications. Unfortunately that Spell Checker doesn't know anything about LATEX commands so there is a tendency to flag those commands as misspelled words. There are several Spell Check applications that are LATEX-aware with the two most popular being Excalibur (maintained by Rick Zaccone, currently at version 4.07 and installed in /Applications/TeX/Excalibur by the MacTEX install package) and cocoAspell (by Anton Leuski and currently at version 2.5 for El Capitan and later) which installs

a Spelling Preference Pane in System Preferences. More information about these two Spell Checkers is found below.

If you use different dictionaries for different documents (e.g., English or German depending upon the document) you can have TeXShop automatically choose the proper dictionary on a document by document basis by placing a line like

```
% !TEX spellcheck = English
```

(for the English (Aspell) dictionary in this case) near the top of each document. Search for 'checking spelling' (without the quotes) in TeXShop's Help → TeXShop Help Panel... for more detailed information on the designation of a particular dictionary.

Excalibur The Excalibur Spell Checker is a stand-alone application that reads in a Source file, allows you to run a spell check which you then Save as a modified Source file; TeXShop automatically picks up the changes in that Source file. There are several versions of Macros that allow you to run Excalibur from within TeXShop. One by Michael Sharpe (with minor modifications by H. Schulz) can be downloaded from <https://herbs.github.io> as TeXShopExcaliburMacro.zip. With any of those macros TeXShop automatically picks up the spell checked and saved version of the Source file and *replaces* the old contents of the displayed Source document by the spell checked version; any changes you make to the Source file while Excalibur is still correcting the document *will be lost* so don't do that!

More dictionaries for Excalibur are available at <http://excalibur.sourceforge.net>.

cocoAspell The cocoAspell Spell Checker integrates itself into the Apple Spell Check system. After enabling it and choosing the active dictionaries from the installed Spelling Preference Pane in System Preferences you can choose one to use within TeXShop by using Edit → Show Spelling and Grammar (Cmd-:) and choosing an Aspell dictionary. You must Quit and Restart TeXShop if you wish to make that dictionary the default.

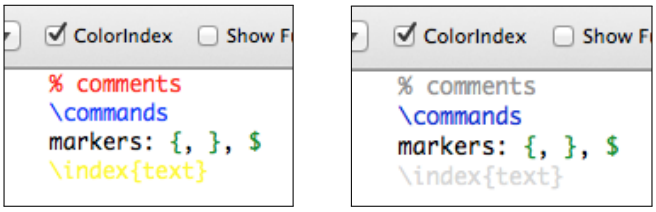
Information on obtaining and installing more dictionaries for cocoAspell is available at <http://people.ict.usc.edu/~leuski/cocoaspell/>. Version 2.5 of cocoAspell installs and works properly with macOS versions El Capitan, Sierra and later. If you have a problem installing on High Sierra or later download the document 'InstallingCocoAspell.pdf.zip' from <https://herbs.github.io>, un-zip it and follow the directions to complete the installation.

2.1.5 "Hiding" Index Commands

Indexing commands tend to duplicate information that is part of the text and therefore interfere with the process of comprehending the text itself. It is possible to have TeXShop colorize `\index` commands in a bright yellow. To do that you need to add a 'ColorIndex' checkbox to the Source window's Toolbar. With the Source window active you can either Right-Click (or Ctl-Click) on the Toolbar, choose Customize Toolbar..., or Window → Customize Toolbar..., and Drag and Drop the ColorIndex checkbox to a place on the Toolbar. Checking that box will make all `\index{text}` commands turn a bright yellow, by default, and "recede" into the background; see Figure (4a) on page 6.

2.1.6 Syntax Coloring

TeXShop provides Syntax Coloring for TeX documents as an aid to pick out text versus markup in source documents. To activate the Syntax Coloring make sure that Syntax Coloring is checked in TeXShop → Preferences → Editor. The default color scheme is a bright red for comments, a dark blue for commands and a dark green for "marker" characters ({, } and $); see Figure (4a) on page 6. In addition, as noted in section (2.1.5) above, TeXShop offers a special Syntax Coloring for



(a) (b)

Figure 4: *The Default (4a) and an alternate set (4b) of Syntax Colors in TeXShop.*

`\index` commands so that they "recede" into the background and you can more easily read the surrounding text.

For TeXShop earlier than 4.08. You may not like the default Syntax Coloring scheme. Searching for 'syntax colors' (without the quotes) in TeXShop's Help Panel gives information on how to change the colors for comments, commands and "marker" characters. It is also possible to change the color of `\index` commands from the default bright yellow to some other color. The corresponding hidden preference variables are indexred, indexgreen and indexblue. See Figure (4b) on page 6 for an example. (If you like those adjusted syntax colors you can download TeXShopSyntaxColors.zip from <https://herbs.github.io>. You can also edit those scripts to create colors you prefer.)

For TeXShop version 4.08 and later. Recent versions of TeXShop now have a Themes tab in TeXShop → Preferences. This tab allows you to change many colors of foreground, background, syntax colors, etc., and save them as a complete Color Theme to be used in the standard Light Mode or the Dark Mode that comes with macOS Mojave. See the version 4.08 section of the Changes document under the Help menu for complete instructions.

If you have changed the syntax colors for pre-4.08 versions of TeXShop using the method given above those colors will *not* be used by version 4.08 out of the box. To retrieve those colors see the method in the Changes document above.

2.1.7 Line Numbers

It is sometimes handy to show line numbers to the left of your Source document. Errors and warnings shown in the Console window during typesetting give the File Name and Line Number in the proper Source file. To turn on Line Numbering by default check the Line Numbers box in the Editor section of the Editor pane in TeXShop → Preferences. You can use Source → Show Line Numbers (Ctl-Cmd-L) to toggle Line Numbering on/off for a particular document. There is also an Edit → Line Number. . . (Cmd-L) Menu Item which will prompt for a line number and then jump to that location.

Note: Lines are defined to be bounded by typed Return characters so a soft-wrapped paragraph will count as a single line.

2.2 Typesetting

Once you are ready to take a look at how your document will appear you typeset it with the default engine, `pdflatex` out of the box, by simply using the Typeset → Typeset (Cmd-T) command.

You may wish to use a different engine as your default. You can change the default engine in TeXShop → Preferences → Typesetting.

If you use the pstricks package extensively or include many eps graphics files in your document you may wish to typeset using `latex` → `dvips` → `ps2pdf` since `pdf(la)tex` does not allow

for direct inclusion of eps files[1]. The easiest way to do this is to include the line

```
% !TEX TS-program = latex
```

at the top of your document. Then TeXShop will use the `latex+distiller` typesetting method noted above no matter what the default engine setting. Change `latex` to `pdflatex` to force use of `pdflatex` to typeset your file.

2.2.1 Removing "Aux" files

The process of typesetting produces several auxiliary files that contain information about cross references, bibliography, indexes, etc. If an error occurs during typesetting these files can be left in some unknown state and need to to be removed before attempting to typeset the document again. The File → Trash Aux Files (Ctl-Cmd-A) command removes most of the files that may create problems.

With TeXShop 3.22 and later there is an additional way to remove those files and then typeset the document with a single command. If you hold the Opt key down while clicking the Typeset menu the Typeset → Typeset command becomes Typeset → Trash Aux & Typeset (Opt-Cmd-T).

Search for 'trash aux' in Help → TeXShop Help Panel... for the list of all file extensions removed by the Trash Aux Files and Trash Aux & Typeset menu commands. The Terminal commands used to add additional extensions to the trash list and return the list to the default list of extensions is also given in that section of TeXShop's Help Panel.

2.2.2 Experimenting

Version 3.37 and later of TeXShop has an Edit → Experiment... menu item. Clicking on that item, with a Source file open, opens a new, resizable "Experiment" window which allows you to enter text. When you click the Typeset button on that window TeXShop will use the preamble from your open Source file and typeset the text in the Experiment window, opening a special Preview window to show the result. Great for experimenting with a figure to get it just right, etc.

2.2.3 Dealing with Errors

Figure (5) on page 8 shows the Console Window which appears when you typeset a file. The Console tab of TeXShop → Preferences allows you to change the background and foreground (character) colors[2] along with some other preferences.

If your typeset run comes across an error it will stop and wait for input which is done on the bottom line. If you wish to end the typeset run you can click on the Abort button. Click on the Goto Error (Ctl-Cmd-E) to move to the error location in the proper Source file. Understand that your typesetting run gives an error when it finally figures out there is an unrecoverable error but that error may be much earlier in your Source file. Also, make sure to read Section (2.4) to properly set up the files in a distributed document.

2.3 Viewing the Output pdf File

Assuming the document was successfully typeset the pdf file will automatically open in a separate Preview window.

You can control how it's displayed in the Preview Menu. You can change the default settings in TeXShop → Preferences → Preview.

[1] The `pdflatex` program in MacTeX-2010 and later will do on-the-fly conversion of eps files.

[2] I've chosen colors that correspond to my ancient Heathkit Terminal.

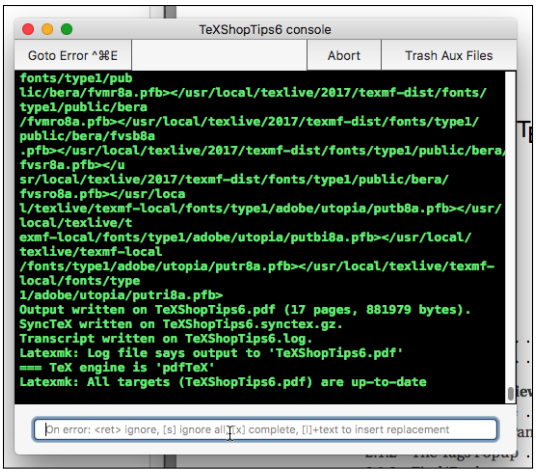
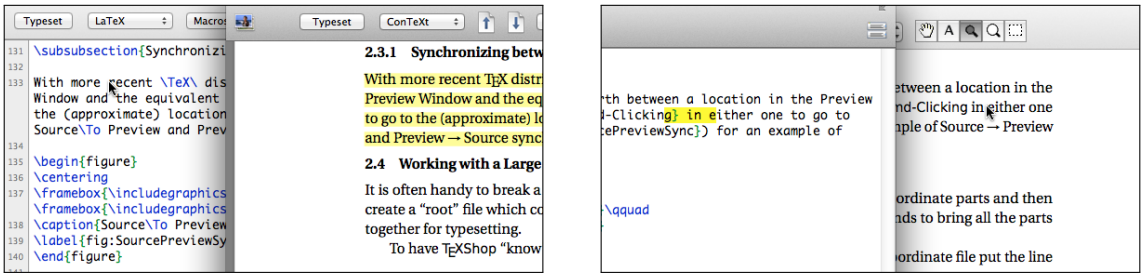


Figure 5: *Console Window*



(a) (b)

Figure 6: *(a) Source → Preview Synchronization; and (b) Preview → Source Synchronization.*

2.3.1 Synchronizing between pdf and Source

With more recent TEX distributions you can also skip back and forth between a location in the Preview Window and the equivalent location in the Source Window by Cmd-Clicking in either one to go to the (approximate) location in the other. See Figure (6) for an example of Source → Preview and Preview → Source synchronization.

2.4 Working with a Large Document

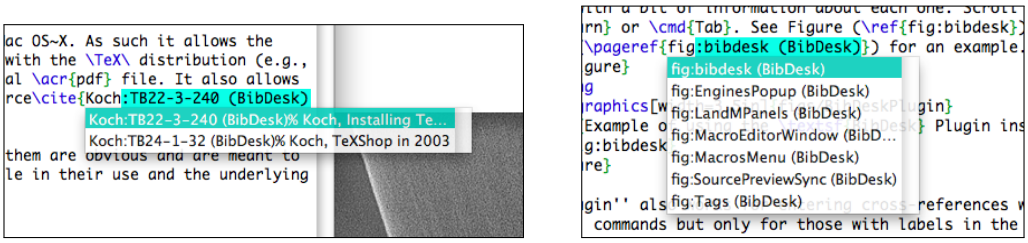
It is often handy to break a large document into more manageable subordinate parts and then create a "root" file which contains the preamble and `\include` commands to bring all the parts together for typesetting.

To have TEXShop "know" which file to typeset when working on a subordinate file put the line

```
% !TEX root = path/to/rootfile.tex
```

at the top of your subordinate file; path/to/rootfile.tex is the relative or absolute path to the root file for this document. Once this is done TEXShop will typeset the root file if you press Typeset → Typeset (Cmd-T) even though you are editing a subordinate file and properly synchronize between the Source and pdf. E.g., if the root file is called mygreatbook.tex and the chapter files, chapter1.tex, etc., are in a chapters sub-folder below the root file then place the line

```
% !TEX root = ../mygreatbook.tex
```



(a) (b)

Figure 7: *BibDesk Plugin: (a) citation insertion; and (b) cross-reference insertion.*

at the top of each of the chapter files. The ../ means go up one folder level to find the root file.

2.4.1 Switching between Source Windows

If you have multiple source files open you can switch between just those windows by using the Window → Next/Previous Source Window (Cmd-F2/Shft-Cmd-F2) menu commands.

2.5 Working with BibDesk and Citations

TeXShop has a built-in "plugin" that interacts with the BibDesk bibliography application to allow you to complete citation references in the `\cite` command. To enable the use of the "plugin" make sure that TeXShop → Preferences → Editor → Editor → BibDesk Completions is checked.

To use it you must first open the required bibliography (`bib`) file(s) in BibDesk. Enter several characters from the reference label within the `\cite` command and press F5 to get a list of matching references from the `bib` file(s) with a bit of information about each one. Scroll to the one you want and press Return or Tab. See Figure (7) on page 9 for an example.

The "plugin" also works for entering cross-references within `\ref` or `\pageref` commands but only for those with labels in the file you are editing.

2.6 Getting Help for Packages

There are many times when having help about a given package can be handy. TeXShop has an interface to `texdoc` which will bring up that documentation. Execute Help → Show Help for Package... (Opt-Cmd-I) and enter the name of the package.

You can also easily look at a package directly with the Help → Open Style File... command and enter the full package file name *including the proper extension* (e.g., `.sty` for packages or `.cls` for document classes).

3 Controlling the Keyboard

One of the best ways to speed up your entry of text in a source file is to keep your hands on the keyboard as much as possible—only one of the reasons I don't like the "clicky" interface of the LaTeX and Matrix Panels. There are many shortcuts associated with the TeXShop menu system but this section is about changing and adding others and other keyboard customizations.

3.1 Menu Shortcuts & System Preferences

Sometimes you'd like to add a shortcut to a menu item that doesn't have one or add one to a command whose shortcut you dislike. Mac OS X 10.4 (Tiger) and later have a method to add shortcuts to specific menu items both globally and in specific programs. This feature has become much more reliable in OS X 10.5 and especially in OS X 10.6 and later.

One example using Mac OS X 10.6 (Snow Leopard) or later: TeXShop 2.36 has added a File → New from Stationery... command, without a shortcut, which can be very helpful once

you set up stationery the way you want. To add Opt-Cmd-N as the shortcut to that menu item: open up the System Preferences application to Keyboard → Keyboard Shortcuts (just Shortcuts in Mavericks) and select Application Shortcuts (App Shortcuts in Mavericks); press the + button to add a shortcut; select TeXShop as the application; enter the exact menu title [New from Stationery... — note you *must* enter a real ellipsis, '...', (Opt-; with the English keyboard layout)]; and press Opt-Cmd-N as the shortcut.

Note: If you don't like a particular shortcut to a menu item you can usually change it to something that suits you better using the same technique used above.

3.2 More Editing Help

TeXShop is built using Apple's programmers interfaces (called frameworks) and therefore inherits all the properties and functionality of those interfaces. There are many things available through the Text framework that aren't tied to the keyboard by default, e.g., many 'emacs-like' keyboard commands, but Apple has made it possible to add those commands to all applications that use the Text framework; e.g., TextEdit and Mail as well as TeXShop.

This is done by creating a special file, DefaultKeyBinding.dict, and placing it in a particular location, ~/Library/KeyBindings (you may have to create the KeyBindings folder there if it doesn't already exist).

You can get more information about this, as well as a (useful) sample, by downloading the KeyBindings.zip file at <https://herbs.github.io>.

3.3 Key Bindings

Besides adding shortcuts to Menu Items you can actually bind keystrokes, within TeXShop, to expand into groups of characters. Checking the TeXShop → Preferences → Editor → Key Bindings option will enable this feature (again use the Editor tab in TeXShop 3.99 and later). You can also toggle it on/off for any particular document using the Source → Key Bindings → Toggle On/Off Menu Item. This feature was previously called Auto Completion; not to be confused with Command Completion—see section (5) below. Note: this facility only works with code generated by a single keystroke (possibly obtained by pressing multiple keys at once rather than in sequence); e.g. it won't work with é on the US keyboard since that is generated by the two keystroke sequence (Opt-e e).

E.g., pressing Opt-, with a US keyboard layout, usually enters ≤ into your document but with Key Binding enabled `\leq` will be entered. Similarly, with some text selected pressing " will surround the selected text with `‘‘` and `’’`.

You can add, remove or change the key bindings using the Key Bindings Editor (Source → Key Bindings → Edit Key Bindings File...). Figures (8) and (9) show the Key Bindings Menu and Editor.

Once in the Editor the left hand column displays the input keystroke while the right hand column shows what will be substituted for that keystroke. To see how you produce some of those keystrokes enable the Keyboard Viewer in System Preferences → Keyboard → Keyboard by checking the 'Show Keyboard & Character Viewers in menu bar' item and then clicking on the new keyboard icon in your Menu Bar.

4 Macros

Macros can be simple text substitutions or Applescript programs that can do all sorts of processing on a file. You can also assign a keyboard shortcut to any macro for direct execution. The ones that are part of TeXShop are found under the Macros Menu.

You can remove or add additional macros to the menu by using the Macro Editor (use the Macros → Open Macro Editor command). The Macro Editor window and extra menu items in the

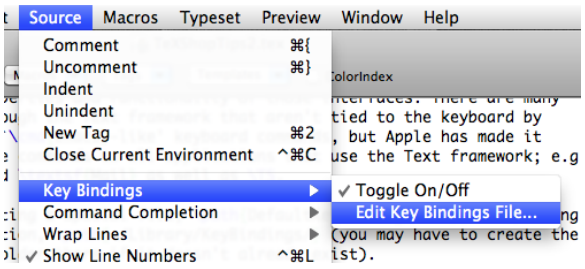


Figure 8: *The Key Bindings Menu.*

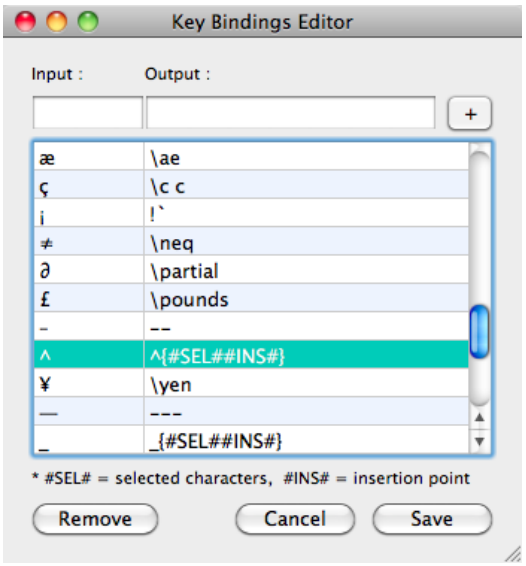


Figure 9: *The Key Bindings Editor.*

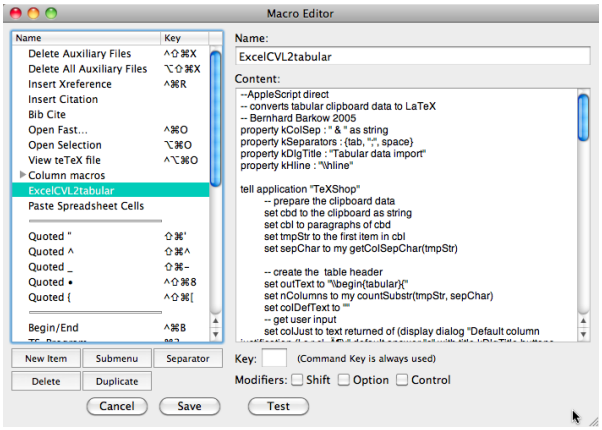


Figure 10: *The Macro Editor Window.*

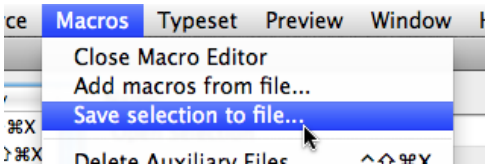


Figure 11: *The extra menu items when the Macro Editor is open.*

Macros Menu when the Editor is open are shown in Figures (10) and (11) respectively.

Besides writing your own macros you can add macros supplied by others to the Macros menu one of two ways: copy and paste the text version of the macro into a New Item in the Macro Editor; or obtain the macro as a `plist` file and use the Add macros from file... command found in the Macros Menu when the Macro Editor is open (again, see Figure (11)).

More information on macros can be found by searching for macros in Help → TeXShop Help Panel....

4.1 Text Macros

Text macros are simple text substitutions. You can also tell TeXShop to insert any selected text using `#SEL#`, place the cursor using `#INS#` and even put in multiple lines in the macro itself. Then you can assign the text macro to a keyboard shortcut.

I like to use Cmd-B and Cmd-I to insert `\textbf{...}` and `\emph{...}` into the document where `...` is any possible selected text. Macros to do that are already under the Macros → Text

Styles Menu so we need only assign keyboard shortcuts to them. To assign Cmd-I to the emphasize macro: open the Macro Editor where the form of the Macros menu appears in the left hand pane; click the emphasize macro found under Text Styles; click the Key insertion box and simply insert a lower case 'i' (the Cmd key is assumed and additional modifier keys can be checked off).

4.2 Applescript Macros

You cannot distinguish Applescript macros in the Macros Menu from text macros but they can do complicated processing and add/change the source file in TeXShop. One example in the default set is the Program macro that creates a

```
% !TEX TS-program = xxxx
```

line at the top of a file with your choice of engine substituted for xxxx. You can look at the Applescript code for this macro by clicking on its name in the Macro Editor.

Some detailed tips on creating Applescript Macros for use in TeXShop can be found in the Help → Notes on Applescript in TeXShop document by Michael Sharpe.

5 Command Completion

LaTeX 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 the Source Tab's Command Completion Triggered By: section in TeXShop → Preferences).

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 in the ~/Library/TeXShop/CommandCompletion folder for much more information. The complete list of completions and abbreviations is available in Appendix A, starting on page (15).

5.1 Completions

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 and then the trigger on a new line 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.

5.2 Substitutions or 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 \bite 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.

A set of tables for all the completions and abbreviations supplied with TeXShop can be found in Appendix A on page 15.

5.3 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 to 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 but insert a '\' (without the quotes) and assign it to a simple Cmd based keystroke.

5.4 Hey, it doesn't work!

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

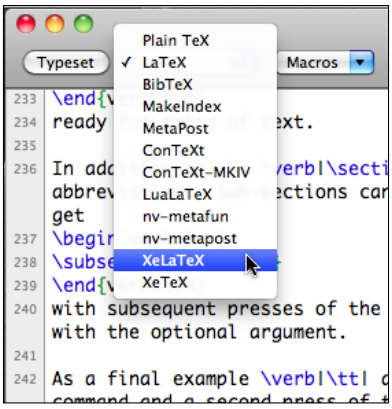


Figure 12: *The Engines Popup Menu on the Source Toolbar.*

6 Extending Processing via Engines

TEXShop offers several default "engines" (also referred to as "scripts" which is left over from earlier times) in its Typeset Menu. These include running Plain TeX or LaTeX (either using `pdftex` or TeX+DVI), BibTeX, MakeIndex, MetaPost or ConTeXt. But there are many things you may wish to do that fall outside of this limited set so TEXShop also allows you to create new engines that are stored in ~/Library/TeXShop/Engines. These additional engines do not show up in the Typeset menu but only in the popup list on the Source and Preview Toolbar (see Figure (12) on page 14).

You can use these engines by choosing from that popup list and then pressing the Typeset button or, a better choice if you use different engines for different documents, by putting a line like

```
% !TEX TS-program = xelatex
```

at the top of your source file; the example given will run the `xelatex` engine on this file independent of other choices. You can override the choice you make in the line with one of the basic engines (e.g., run BibTeX) by using the items in the Typeset Menu directly.

TEXShop is shipped with a few engines activated (i.e., directly in the ~/Library/TeXShop/Engines folder) but also includes several additional ones in ~/Library/TeXShop/Engines/Inactive. As an example let's activate and use the pdflatexmk engine found in ~/Library/TeXShop/Engines/Inactive/Latexmk.

6.1 The `pdflatexmk` engine

If your document has cross-references, bibliographies and/or indexes it takes multiple `pdflatex` runs with intermediate runs of `bibtex` and/or `makeindex` to create the bibliographies, indexes and resolve all cross-references. The `pdflatexmk` engine automates this whole process.

TEXShop 3.07 or 2.46 and later activate the pdflatexmk engine by default *in a fresh installation.* If you are using an earlier version of TEXShop, or even updated to the latest version from an earlier version, you need to activate the engine. To activate the engine simply move the `pdflatexmk.engine` file from ~/Library/TeXShop/Engines/Inactive/Latexmk two folders up, to ~/Library/TeXShop/Engines. When you restart TEXShop you can check that pdflatexmk is now in the popup menu.

Then place the line

```
% !TEX TS-program = pdflatexmk
```

at the top of your source file. From then on when you simply typeset the file (Typeset → Typeset or Cmd-T) TeXShop will use this engine and the complete process of typesetting the document to its final form will be carried out.

Appendices

A — Command Completion Tables

The following tables contain the Command Completions and Abbreviations included by default with TeXShop. Table 1 on page 16 is a list of all included environment abbreviations. Table 2 on page 17 lists included abbreviations/completions for commands and declarations. Finally, Table 3 on page 18 are the included abbreviations for Greek letters.

It is important to remember that with a given abbreviation successive presses of the trigger key go to the next match in the list. E.g., there are three sectioning commands, sec for the standard section command, secs for the "starred" version of the command and seco for the version with an optional argument; if you enter sec as your abbreviation successive presses of the trigger goes from the sec to the secs to the seco versions before returning to your original abbreviation. That means there are many abbreviations that you never need to remember.

Note: do *not* attempt to memorize these tables. Learn a few items that you use all the time and then slowly add to your knowledge as you need them.

Table 1: *Environment abbreviations.*

Abbreviation	**Environment**	**Abbreviation**	**Environment**
barr	array	blett	letter
babs	abstract	blist	list
bali	align	bminp	minipage
balis	align*	bminpo	minipage
baliat	alignat	bmult	multline
baliats	alignat*	bmults	multline*
balied	aligned	bpict	picture
baliedat	alignedat	bpmat	pmatrix
baliedato	alignedat	bquot	quotation
bapp	appendix	bquo	quote
bbmat	bmatrix	bsplit	split
bcase	cases	bsubeq	subequations
bcent	center	btab	tabular
bcenum	compactenum	btabs	tabular*
bcenumo	compactenum	btabx	tabularx
bcitem	compactitem	btabl	table
bcitemo	compactitem	btablo	table
bdes	description	btabls	table*
benu	enumerate	btablso	table*
benuo	enumerate	btbl	table
bequ	equation	btblo	table
bequs	equation*	btbls	table*
beqn	eqnarray	btblso	table*
beqns	eqnarray*	btabb	tabbing
bfig	figure	bbib	thebibliography
bfigo	figure	bindex	theindex
bframe	frame	btheo	theorem
bframeo	frame	btitpg	titlepage
bflalig	flalign	btrivl	trivlist
bflaligs	flalign*	bvarw	varwidth
bfll	flushleft	bverb	verbatim
bflr	flushright	bvers	verse
bgath	gather	bwrap	wrapfigure
bgaths	gather*	bwrapo	wrapfigure
bgathed	gathered	bwrapo2	wrapfigure
bgathedo	gathered	bwrapoo	wrapfigure
bite	itemize		
biteo	itemize		

Table 2: *Commands and Declarations.*

Abbreviation	**Command**	**Abbreviation**	**Command**	**Abbreviation**	**Command**
- -	textendash	midr	midrule	renewcomo	renewcommand
- - -	textemdash	mnorm	mathnormal	renewcomoo	renewcommand
- - -	textemdash w/sp	msf	mathsf	rncm	renewcommand
adlen	addtolength	mtt	mathtt	rnewc	renewcommand
adcount	addtocounter	mit	mathit	rncmo	renewcommand
bf	textbf	midr	midrule	rnewcoo	renewcommand
bfd	bfseries	mnorm	mathnormal	rncmoo	renewcommand
biblio	bibliography	mdd	mdseries	rmc	rmfamily
bibstyle	bibliographystyle	mbox	mbox	rbox	raisebox
botr	bottomrule	makebox	makebox	rboxo	raisebox
bibitem	bibitem	mboxo	makebox	rboxoo	raisebox
bibitemo	bibitem	makebox	makebox	sec	section
center	centering	mboxoo	makebox	secs	section*
chap	chapter	mpar	marginpar	seco	section
cmidr	cmidrule	multic	multicolumn	ssec	subsection
cmidro	cmidrule	ncol	space & space	ssecs	subsection*
em	emph	ncm	newcommand	sseco	subsection
emd	em	newc	newcommand	sssec	subsubsection
foot	footnote	ncmo	newcommand	sssecs	subsubsection*
frac	frac	newco	newcommand	ssseco	subsubsection
fbox	fbox	ncmoo	newcommand	spar	subparagraph
fboxo	framebox	newcoo	newcommand	spars	subparagraph*
fboxoo	framebox	nct	newcolumntype	sparo	subparagraph
geometry	geometry	newct	newcolumntype	setl	setlength
hw	headwidth	newpg	newpage	stcount	stepcounter
hw2tw	headw=textw	npg	newpage	sf	textsf
href	href	nline	newline	sfd	sffamily
item	item	newlin	newline	sc	textsc
ito	item	nlen	newlength	scd	scshape
incg	includegraphics	newlen	newlength	sl	textsl
incgo	includegraphics	nenv	newenvironment	sld	slshape
it	textit	newenv	newenvironment	sqrt	sqrt
itd	itshape	nenvo	newenvironment	sqrto	sqrt
latex	LaTeX	newenvo	newenvironment	tt	texttt
latexs	LaTeX w/sp	nenvoo	newenvironment	ttd	ttfamily
latexe	LaTeXe	newenvoo	newenvironment	tw	textwidth
latexes	LaTeXe w/sp	pgref	pageref	tex	TeX
label	label	par	paragraph	texs	TeX w/sp
lbl	label	pars	paragraph*	tilde	textasciitilde
lettrine	lettrine	paro	paragraph	topr	toprule
lettrineo	lettrine	pgs	pagestyle	toc	tableofcontents
listf	listoffigures	parbox	parbox	tableofcontents	tableofcontents
listt	listoftables	parboxo	parbox	tpgs	thispagestyle
rule	rule	parboxoo	parbox	thispagestyle	thispagestyle
ruleo	rule	parboxooo	parbox	up	textup
mbf	mathbf	pbox	parbox	upd	upshape
mrm	mathrm	pboxo	parbox	url	url
mcal	mathcal	pboxoo	parbox	usep	usepackage
msf	mathsf	pboxooo	parbox	usepo	usepackage
mtt	mathtt	ref	ref	verb	verb
mit	mathit	renewcom	renewcommand	verb2	verb

Table 3: *Greek Letters. The 'd' versions are not shown.*

Abbreviation	**Command**	**Abbreviation**	**Command**
xa	alpha	xvp	varpi
xb	beta	xph	phi
xch	chi	xcph	Phi
xd	delta	xvph	varphi
xcd	Delta	xps	psi
xe	epsilon	xcps	Psi
xve	varepsilon	xs	sigma
xet	eta	xcs	Sigma
xg	gamma	xvs	varsigma
xcg	Gamma	xz	zeta
xio	iota	xr	rho
xk	kappa	xvr	varrho
xl	lambda	xt	tau
xcl	Lambda	xth	theta
xm	mu	xcth	Theta
xn	nu	xvth	vartheta
xo	omega	xu	upsilon
xco	Omega	xcu	Upsilon
xp	pi	xx	xi
xcp	Pi	xcx	Xi