

Using Command Completion with T_EXShop

Herbert Schulz
herbs2@mac.com

2010/01/30

Introduction

As of v1.34 T_EXShop offered a Command Completion facility that was reasonably powerful, if under-utilized. Command Completion in T_EXShop allows continuations (completions) or substitutions (abbreviations) for a set of characters bounded on the left by a Word Boundary Character¹ and triggered by the Escape (⌘) key.

With the help of the good folks on the Mac OS X TeX e-mail list², an updated `CommandCompletion.txt` file was created along with associated Applescript macros to take advantage of that facility. The completions and abbreviations supplied often contain bullet characters, '•', called Marks³, as placeholders for command arguments or to easily get to the end of an environment. Skipping forward and backward to these Marks was accomplished using macros⁴; the macros jumped to the Marks and selected, or deleted, them. Most of the abbreviations were inspired by those used in the FastT_EX⁵ set used with Typelt4Me⁶.

Version 2.30 and later of T_EXShop offers a built-in and enhanced version of Command Completion inspired by Hugh Neary and Will Robertson. There is no longer a need for the Applescript macros and the arguments of completions can have short comments.

Installation

Simply place this version of T_EXShop into `/Applications/TeX/`. If you ever started up an older version of T_EXShop (earlier than 2.30) also follow the directions in the following sub-section.

¹The Word Boundary Characters are space, tab, linefeed(newline), period, comma, semicolon, colon, {, }, (,) or \ (actually the TeX Command Character which can vary in different implementations). The { and \ also become part of the expansion.

²Subscribe by sending an e-mail to <mailto:MacOSX-TeX-on@email.esm.psu.edu>.

³Previously called Tabs.

⁴The original `CommandCompletion.txt` files, macros and documentation are still available as `CommandCompletion.zip` from <<http://homepage.mac.com/herbs2/>>.

⁵FastT_EX was developed by Filip G. Machi, Jerrold E. Marsden and Wendy G. McKay. For more information see the FastT_EX web page, <<http://www.cds.caltech.edu/~fastex/>>.

⁶Typelt4Me, by Riccardo Ettore, presently a preference pane, allows abbreviation replacement in most OS X programs with "dictionaries" that can be application dependent. See the Typelt4Me web page, <<http://www.typeit4me.com/>>, for more information.

CommandCompletion.txt

A new default version of `CommandCompletion.txt` is embedded within this new version of `TEXShop`. It will get installed, along with other files, the first time `TEXShop` runs *unless* you've used `TEXShop` before. In that case you need to move the directory `~/Library/TeXShop/CommandCompletion/` (~ is your HOME directory) to your Desktop and start the new version of `TEXShop`; a replacement folder with the new version of `CommandCompletion.txt` will be created.

If you've already added items to the original file you will be able to merge them into the new file after it's created. You can open the new version of `CommandCompletion.txt` by clicking on the `Source→Completion→Open Completion File...` menu item. The older file *must* be opened with Unicode (UTF-8) encoding: use the `File→Open...` (⌘-O) menu command and make sure to select the `Unicode (UTF-8)` encoding before opening your old version in `TEXShop`.

Some of What's New in T_EXShop

Most important are four interconnected changes to `TEXShop`: an addition to the way `TEXShop` handles completions from `CommandCompletion.txt`; a new menu that has commands for searching and selecting Marks within completions; the ability to have comments attached to Marks; a new `CommandCompletion.txt` file that takes some advantage of the previous three changes. The following four sub-sections cover each of these changes in more detail. Some minor menu changes are discussed separately.

Changes to Completion Handling

Completions (in the `CommandCompletion.txt` file) in previous versions of `TEXShop` could contain a single `#INS#` command for the positioning of the insertion point within the completion.

This version of `TEXShop` allows completions to have *two* copies of `#INS#` and the text between them is selected. A single `#INS#` behaves the same as before; there is complete backward compatibility with the previous versions of `TEXShop`.

The New Source→Completion→Marks Menu

The new `Source→Completion→Marks` menu contains commands to search for, move to and select Marks and Comments. The commands are shown in Table (1) on page 3. The (De1) versions of the search commands only show in the menu when the Option (⌥) key is pressed and the `Insert Comment` command only appears when you hold down the Control (⌘) key. The `Insert Mark` command is added since using `TEXShop`'s Auto-Complete (key-binding) facility will insert `\bullet` in the document when the keystroke that normally inserts a '•' (⌥-8 with a US keyboard mapping) is pressed. The default version of the Marks menu is shown in Figure (1) on page 3.

Comments

The changes mentioned in the previous sub-sections allow completions to contain Comments—short “memory joggers” that have some information about the contents of a given argument. The comments are contained within arguments and are surrounded

Menu Item	Shortcut	Internal Connection
Next Mark	\wedge -⌘-F	Jump to and select the next Mark and/or Comment.
Next Mark (Del)	\wedge -⌘-F	Jump to and select the next Mark and/or Comment and delete the Mark. This is most useful when you have nested environments to automatically delete a Mark at the end of an inner environment.
Previous Mark	\wedge -⌘-G	Like Next Mark but search backwards.
Previous Mark (Del)	\wedge -⌘-G	Like Next Mark (Del) but search backwards.
Insert Mark	⌘-8	Places a Mark at the insertion point. Used to mark arguments, etc., when creating new completions in <code>CommandCompletion.txt</code> .
Insert Comment	\wedge -⌘-8	Places a Comment Skeleton, “•<>” with the insertion point before the “>”, at the insertion point. Handy for creating comments in <code>CommandCompletion.txt</code> .

Table 1: Commands in the Source–Completion–Marks Menu.

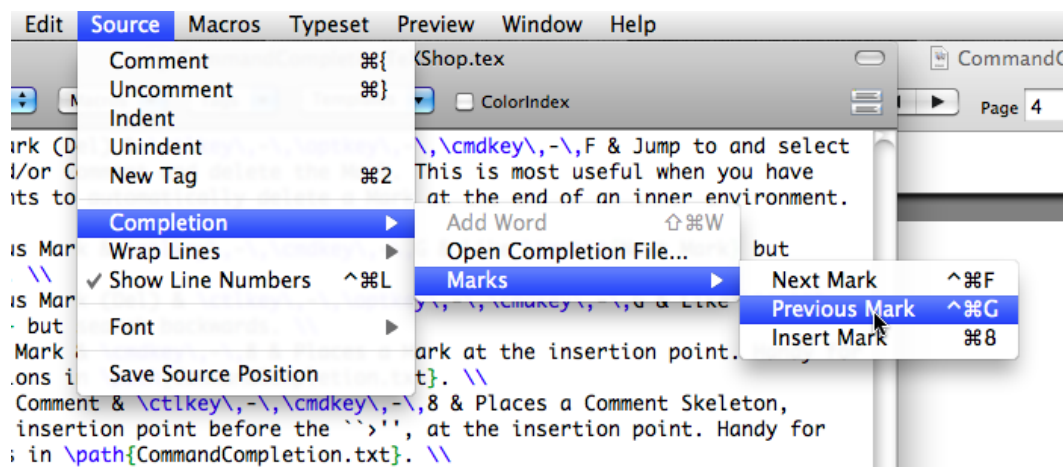



Figure 1: The Default Source → Completion → Marks Menu.

by “•<” and “>”⁷ within the arguments; if the first argument contains a comment it should be surrounded by two #INS# so it is the initial selection.

The New CommandCompletion.txt File

There are a few changes to the new CommandCompletion.txt file that comes with this version of TeXShop when compared to the one supplied with the original add-on version:

- there are some minor bug fixes and a few additional environments and commands;
- all single #INS# commands are replaced by #INS#•#INS# so that the initial selection is a selected Mark, . This is for consistency with the appearance and behavior when using the Next/Previous Mark commands;
- there are a few (too few?) examples of comments in commands and environments.

Minor Menu System Revisions

There are three minor changes to the Menu system:

1. The Source→Show Line Numbers menu item is now assigned the shortcut ^-⌘-L so that line numbering may be quickly toggled on and off. This doesn't change the default behavior set in TeXShop→Preferences.
2. The keyboard shortcut ^-2 is assigned to Window→Split Window. This works in both the Source and Preview windows; whichever is active.
3. There is an additional “hidden” menu item in Source→Wrap Lines “under” the Hard Wrap command called Soft Wrap that is assigned to ⌘-⇧-⌘-H. Press ⌘ when the Source→Wrap Lines sub-menu is open. Use this cautiously since it will combine multiple paragraphs into a single large paragraph if they are all selected and using the command with no selection will remove *all* of the newlines in the document.

Usage

Command Completion

A typical use of command completion is to set up environments. To do this type \b and ⌘; this should get you \begin{. Then start to type the environment name; e.g., eq and ⌘ will give


```
\begin{equation}
```



```
\end{equation}•
```

while the next ⌘ gives eqnarray followed by it's *-variant. After entering your equation text at the cursor run the Source→Completion→Marks→Next Mark command and the cursor will select (and delete if Next Mark (Del) is used) the next ‘•’ so you can start to type following text.

The macros are also handy for commands that take multiple arguments. For example, to create a new command with an optional argument type \new or \newc and then ⌘ three times to get

```
\newcommand{}[•][•]{•}
```

with the first mark selected. After entering the new command's name use the Next Mark command to jump to the next argument, etc.

⁷Note that ‘<’ and ‘>’ are single “guillemot” glyphs, not ‘<’ and ‘>’.

Abbreviations

In addition to command completion there also exist many abbreviations for commands. The principal difference is that the abbreviations are not just the start of a command name. For example typing `benu` and then \mathfrak{O} *at the beginning of a line*⁸ will produce the complete enumerated list environment:

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

as you might expect. Abbreviations like this exist for many environments as well as sectioning commands. Alternate command versions with one or more options or *-variants have names that end with 'o' (one or more) or 's' respectively: e.g., `sec` and two presses of \mathfrak{O} or `secs` and a single \mathfrak{O} at the start of a new line give `\section*{•}`. By the way, After typing the text for the first item, typing `it` and \mathfrak{O} on a new line will generate another `\item` with a selected Mark on the line below it; continued presses of \mathfrak{O} will give `\item{•}` with a Mark on the following line, `\textit{it{•}}` and finally `\itshape` before returning to the original `it`.

You must remember to have one of the Word Boundary Characters before use or the substitution won't operate properly. This is not a problem with environments and sectioning commands, since you usually start them on a new line, but can be for other abbreviations. Therefore many abbreviation also have a '\ ' version; e.g., `\tt` and \mathfrak{O} will not expand properly since the '\ ' isn't a Word Boundary Character while `\tt` and \mathfrak{O} will expand to `\texttt{•}` while a second \mathfrak{O} will give the declaration `\ttfamily`⁹.

Many of the Greek characters and in-line math versions of the Greek characters have abbreviations with the following rules:

1. The abbreviations for Greek characters all start with an 'x' and a notation for the character: e.g., `xa` or `\xa`¹⁰ and \mathfrak{O} give `\alpha`.
2. The var version of several Greek characters start with 'xv' and the notation for the character: e.g., `xth` gives `\theta` while `xvth` and \mathfrak{O} gives `\vartheta`.
3. To get capitals for some letters use an 'xc': e.g., `xg` gives `\gamma` while `xcg` gives `\Gamma`.
4. Finally, preceding by a 'd' gives the following Greek character as an in-line math equation: e.g., `dxcd` gives `\(\Delta\)`.

Abbreviations will be completed and cycle through matches just like the command completions: e.g., both the abbreviation `newcoo` (note the 'oo' at the end of the abbreviation) and \mathfrak{O} or `newc` followed by three \mathfrak{O} key presses on a new line give `\newcommand{•}[•][•]{•}`, the `\newcommand` with two optional arguments. There are alternate abbreviations for some commands: e.g., `ncm` gives the same result as `newc`.

I suggest that you read through the `CommandCompletion.txt` file to see what abbreviations are available; all lines with ':' are abbreviations. Naturally, you can change them to suit your needs and add and delete others.

Comments

It is easy to remember the arguments for commands that are used fairly often but to forget those rarely used; these are the perfect candidates for comments. An example

⁸Or after any other Word Boundary Character.

⁹Similar abbreviations exist for `bf`, `sf`, `sc`, etc. Math versions have a preceding `m`; e.g., `mbf` and \mathfrak{O} will give `\mathbf{•}`.

¹⁰All of the Greek character abbreviations have \ versions.

is the order of the arguments for the `\rule` command; type `\rul` and `␣` twice to get `\rule[•<lift>]{•<width>}{•<height>}`, the version with the optional argument¹¹. 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 `␣` 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 `␣`.

Other Environments

Environments that aren't built into the `CommandCompletion.txt` file can always be added if you use them a lot but there is an alternative for occasional use. Built into the completion algorithm is a way to complete environments. First press `\b` and `␣` to get `\begin{`, enter the environment name and the closing `}` and then `␣` again; the closing `\end{}` with the corresponding environment name will be generated on a separate line.

Abbreviations in CommandCompletion.txt

This section contains a, hopefully, complete list of the abbreviations supplied in `CommandCompletion.txt`. The list has been broken up into Environments, Commands & Declarations and Greek Letters. If you supply a certain beginning abbreviation the search will start at the first match to what you supply and succeeding presses of `␣` will go down the list until there is no longer a match; e.g. if you type `be` succeeding presses of `␣` will match `benu`, `benuo`, `bequ`, `bequs`, `beqn` and `beqns` before returning to the original `be`. Adding more letters to the abbreviation may get you to the desired completion with fewer presses of `␣`. In reality some of the Commands & Declarations are scattered between the Environments in the `CommandCompletion.txt` file so there might be additional items at times. The tables don't include standard completions or the `'\'` versions of the abbreviations.

NOTE: The list may be a bit intimidating. There is no need to “memorize” all of these abbreviations; learn the minimum number as you need them and use the `␣` to get the variations.

Environment Abbreviations

Table (2) on page 8 contains a list of abbreviations for different environments supplied in `CommandCompletion.txt`. Multiple vertically adjacent Environments with the same name correspond to variations in number and distribution of possible optional arguments or `*`-variants. There can also be more than one abbreviation for the same environment.

Commands & Declarations

As with Environments there are lots of variations with options and `*`-variants as well as multiple abbreviations corresponding to the same command. See Table (3) on page 9.

¹¹This is `\rule[#INS#•<lift>#INS#]{•<width>}{•<height>}` in the `CommandCompletion.txt` file.

Greek Letters

The Greek Letter abbreviations appear in Table (4) on page 10. The in-line equation, i.e., 'd', versions of the letters are not shown.

Making Additions to `CommandCompletion.txt`

If you are adding items to the `CommandCompletion.txt` there are a few things you should know about its structure:

- Each environment has three entries: a completion that removes the leading `\begin`, i.e., it starts with a leading '{' and the environment name; two abbreviations that have an abbreviation name without a backslash (\) and the same abbreviation with the backslash. Commands may have more forms; the full command as well as abbreviation(s) all with and without a leading \.
- You should add all the variations with slightly different endings for the abbreviations. I use an 'o' at the end of an abbreviation if that variation has an optional argument, 'oo' for two optional arguments, 's' for starred forms of commands, etc.
- The order of similar items in the file *does* make a dramatic difference in the order in which items are found; items placed *later* will be found *earlier* (the file is searched backwards). E.g., the order of items obtained when you press `\b` and then `␣` depends purely on the order of matches in the `CommandCompletion.txt` file.
- For maximum convenience place a Mark¹² within each argument of commands. Surround the very first argument with two `#INS#` commands so it comes out selected. If you want to have a comment in any arguments insert a Comment Skeleton¹³ and fill it in.

I'd suggest that you take a look in the `CommandCompletion.txt` file for examples.

Bugs

The `CommandCompletion.txt` file is usually searched backward from the last item but, on rare occasions, the search direction seems to switch so you don't get matches in the order you expect. You can usually force the search to go back to the "correct" direction by pressing `---` and then `␣` three times and then remove the `---`. If that doesn't correct the direction you can use `⇧-␣` to search in the "other" direction.

What's Missing

I'd love to be able to have the completions preserve indentation but that is not in the books for now.

Any other suggestions are welcome and will be considered for inclusion in later iterations of the Command Completion code.

Try it... I hope you like it.

¹²Using Insert Mark (`⌘-8`) from the Source→Completion→Marks menu.

¹³Using Insert Comment (`⌘-8`) from the Source→Completion→Marks menu.

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	btabs	tabular
bcenum	compactenum	btabs	tabular*
bcenumo	compactenum	btabsx	tabularx
bcitem	compactitem	btabsl	table
bcitemo	compactitem	btabslo	table
bdes	description	btabslso	table*
benu	enumerate	btabslso	table*
benuo	enumerate	btabslso	table
bequ	equation	btabslso	table
bequs	equation*	btabslso	table*
beqn	eqnarray	btabslso	table*
beqns	eqnarray*	btabslso	table*
bfig	figure	btabslso	tabbing
bfigo	figure	btabslso	thebibliography
bframe	frame	btabslso	theindex
bframeo	frame	btabslso	theorem
bflalign	flalign	btabslso	titlepage
bflaligns	flalign*	btabslso	trivlist
bfl	flushleft	btabslso	varwidth
bflr	flushright	btabslso	verbatim
bgath	gather	btabslso	verse
bgaths	gather*	btabslso	wrapfigure
bgathed	gathered	btabslso	wrapfigure
bgathedo	gathered	btabslso	wrapfigure
bite	itemize	btabslso	wrapfigure
biteo	itemize	btabslso	

Table 2: Environment abbreviations supplied in `CommandCompletion.txt`.

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	newcolumnntype	sparo	subparagraph
geometry	geometry	newct	newcolumnntype	setl	setlength
hw	headwidth	newpg	newpage	stcount	stepcounter
hw2tw	headw=tw	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	sqr	sqr
itd	itshape	nenvo	newenvironment	sqrto	sqr
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: Commands and Declarations in `CommandCompletion.txt`.

Abbreviation	Command	Abbreviation	Command
xa	alpha	xph	phi
xb	beta	xcph	Phi
xch	chi	xvph	varphi
xd	delta	xps	psi
xcd	Delta	xcps	Psi
xe	epsilon	xs	sigma
xve	varepsilon	xcs	Sigma
xet	eta	xvs	varsigma
xg	gamma	xz	zeta
xcg	Gamma	xr	rho
xio	iota	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
xvp	varpi		

Table 4: Greek Letters in `CommandCompletion.txt`. The ‘d’ versions are not shown.