

DRAFT

Fast_ETeX Documentation

Filip G. Machi
Jerrold E. Marsden
Wendy G. McKay

May 25, 1996

This documentation is set up for pages to be printed in landscape orientation

FASTEX PROJECT

Control & Dynamical Systems 104-44
California Institute of Technology
Pasadena, CA 91106, USA

`fastex@cds.caltech.edu`
`http://cds.caltech.edu/~fastex`
`ftp://cds.caltech.edu/pub/fastex`

FAST_EX: A SYSTEM OF KEYBOARD SHORCUTS FOR THE FAST KEYING OF T_EX

©1995 Filip G. Machi, Jerrold E. Marsden, and Wendy G. McKay

Version 1: March 27, 1996—update(May 25, 1996)

FASTEX: A SYSTEM TO SPEED UP THE INPUT OF TEX ON THE MACINTOSH

©1990 Jerrold and Barbara Marsden

Version: October 1, 1994

PART I

F A S T E X
Preface & Introduction

DRAFT VERSION MAY 25, 1996

CONTENTS—Part I: Introduction & Preface

1	General Features of FasTeX	2
1.1	What Types of Shortcuts Come with FasTeX?	3
1.2	Examples of Simple Expansions	3
1.3	Examples of Generic (Universal) Simple Expansions	7
1.4	Operator Names in Mathematics	7
1.5	Complex Expansions—Templates	8
1.6	The Importance of Producing Default Files	9
1.7	FasTeX is Universal, Fast, and Accurate	9
1.8	You can get FasTeX off the WEB	9
2	A Sample Input	10
3	The Macintosh Version of FasTeX	11
3.1	Installation	11
3.2	A little about <i>TypeIt4Me</i>	11
3.3	A Note about \$	12
3.4	A Sample Creation of a Shortcut	12
4	The UNIX version of FasTeX	13

4.1	System Requirements	14
4.2	Installation	14
4.3	Customization	16
4.4	Some specific differences between the <i>scedit</i> and <i>TypeIt4Me</i> versions of FasTeX	16

Fast_TE_X:

A SYSTEM OF KEYBOARD SHORTCUTS FOR THE FAST KEYING OF T_EX¹

Preface

Fastex is a project that has been in development since 1986, when personal computers and the popularity of TeX made it obvious that individuals were going to be doing much more typesetting than in the past. The need for a system enabling fast input became clear. A few systems allow one to call up abbreviations or shortcuts that are called things like “glossaries” or “libraries”. However, they are system specific and editor specific and also often require special keystroke sequences to activate. Fastex is designed with the user in mind to facilitate fast and accurate inputting and to help one remember Tex commands.

Some early versions of this system were prepared in collaboration with June Meyermann, Esther Zack and Barbara Marsden. Their work laid an important foundation for the current product and we are grateful to them for their work. The project proceeded over the years by improving the quality of the shortcut naming conventions and by locating good software with which to run the system. We feel that we have now achieved our goals. This system can be run on both Macintosh and Unix platforms and the shortcut names are universal, independent of the platform and the particular editor being used. This universality is ultimately of considerable importance.

We hope that you will find this product useful. If you do, we do accept shareware donations (made out to Caltech) for whatever amount you find appropriate, to keep the project running and the files updated.

Fil Machi
Jerrold Marsden
Wendy McKay

¹This introduction is based on the article in TUGboat Volume 16, Number 4, December 1995—VERSION MARCH 1, 1996

Acknowledgements

As we said above, we especially thank June Meyerman, Esther Zack and Barbara Marsden for their development on early versions. In particular, Barbara's work on an early version of the Mac documentation was very helpful. We also thank Jonathan Dorfman for his early help with software issues and for proving that this concept could work. We also thank Ricardo Ettore, whose software *TypeIt4Me* was selected for the Mac version.

We would like to collectively thank all the prior users of Fastex who told us of bugs and who made valuable suggestions.

We also thank the California Institute of Technology for their general support that was crucial for this project to come to fruition. We also thank Richard Murray for doing some testing and for his valuable advice.

We would also like to add a special word of thanks to Calvin Jackson and David Saloman, who helped us with advice on $\text{T}_{\text{E}}\text{X}$ outputting so that it was possible to integrate plain $\text{T}_{\text{E}}\text{X}$ commands with \LaTeX commands to produce *automatic documentation* from the shortcut text files for any kind of $\text{T}_{\text{E}}\text{X}$ via some Unix incantations to create the $\text{T}_{\text{E}}\text{X}$ code.

1 General Features of Fas $\text{T}_{\text{E}}\text{X}$

Fas $\text{T}_{\text{E}}\text{X}$ is a system of keyboard shortcuts for speeding up the typing of $\text{T}_{\text{E}}\text{X}$ from the keyboard. Fas $\text{T}_{\text{E}}\text{X}$ is currently available for the Macintosh and UNIX. It replaces any keyboard shortcut by the equivalent $\text{T}_{\text{E}}\text{X}$ command or group of commands in Plain $\text{T}_{\text{E}}\text{X}$, $\mathcal{A}\mathcal{M}\mathcal{S}\text{-T}_{\text{E}}\text{X}$, $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$, or \LaTeX . At the start of the session, the typist specifies which flavor of $\text{T}_{\text{E}}\text{X}$ is to be used from the set of files, one for each flavor, containing the mapping information that expands the keyboard shortcut into the corresponding $\text{T}_{\text{E}}\text{X}$ command(s). These files we shall refer to as the "Fas $\text{T}_{\text{E}}\text{X}$ shortcut files".

While keying in the text of the document, the typist enters keyboard shortcuts. Expansion of a shortcut is activated by typing one of several predefined activation keys. After pressing the activation key the shortcut name is overwritten with the expansion text. It is common to use the spacebar as one of the activation keys.

For example, FasTeX replaces the keystroke sequence “xa” by `\alpha`, and the sequence “dxa” by `$$\alpha$` etc. (The motivation for “xa” is that ‘x’ introduces any Greek letter and “a” is the Latin form of the Greek letter alpha. Similarly, the letter “d” preceding the “xa” in this example indicates that dollar signs be put around the expansion of “xa”.) FasTeX can deal with long or short abbreviations with equal ease.

Using FasTeX, the typist, whether the author or another keyboard entry person, is able to keep his/her hands on the standard portion of the keyboard; and need only very rarely hit keys far from the home row of the keyboard, such as the backslash or dollar sign key.

1.1 What Types of Shortcuts Come with FasTeX?

The distributed FasTeX shortcuts are all designed to keep the typist’s fingers near the home row keys. Some FasTeX shortcuts just rename standard TeX commands, while others expand into multiple lines of TeX incantations, referred to here as templates, which can be used to simplify entering large or complex structures. Although FasTeX comes with a comprehensive, well thought out, and thoroughly tested list of shortcuts, personal shortcuts are easily customized.

Here are the kinds of shortcuts contained in the distributed FasTeX files.

1.2 Examples of Simple Expansions

These are complete expansions of some commonly used TeX commands.

- *Greek letters*

<i>To get this</i>	<i>Type</i>
<code>\alpha</code>	xa
<code>\beta</code>	xb
...	

The general shortcut naming rule for producing Greek letters is “x” followed by the Latin equivalent of the Greek letter, to produce the TeX command for that Greek symbol.

- *Math environment*

<i>To get this</i>	<i>Type</i>
d	<code>d</code>
$\text{d}l\text{r}$	<code>dlr</code>
$\text{d}\alpha$	<code>dxa</code>
$\text{d}\beta$	<code>dxb</code>
$\text{d}\pi$	<code>dxp</code>
$\text{d}\varphi$	<code>dxvph</code>
...	

The letter “d” at the start of a shortcut name generally means surround the expansion text with dollar signs.

- *Capital letters*

<i>To get this</i>	<i>Type</i>
$\text{d}A$	<code>dca</code>
$\text{d}\Gamma$	<code>xcg</code>
$\text{d}\Omega$	<code>dxco</code>
...	

In compound shortcut names like ‘dca’ and ‘xcg’, capital letters are entered by typing a “c”, to indicate “Capital”, before the lowercase version of the letter.

- *Calligraphic letters*

<i>To get this</i>	<i>Type</i>
$\text{c}al A$	<code>cca</code>
$\text{c}al B$	<code>ccb</code>
...	

A “c” preceding the shortcut name for a capital letter will produce the calligraphic form of that letter.

- *Superscripts*

<i>To get this</i>	<i>Type</i>
α	hxa
β	hxb
4	h4
i	hi
$\{ij\}$	hij
$^{-1}$	hmo
\ast	hst
...	

- *Subscripts*

<i>To get this</i>	<i>Type</i>
α	lxa
β	lxb
4	l4
...	
i	li
$\{ij\}$	lij
0	l0

Remember this as “h” for higher, “l” for lower.

- *Fractions*

<i>To get this</i>	<i>Type</i>
$\frac{1}{2}$	f12
$\frac{\quad}{\quad}$	fu
$\{ \}$	fof
$\frac{\quad}{\quad}$	fu fof eb
$\frac{\partial}{\partial x}$	fpx
...	

- *Other Symbols (German or Fraktur, Open or Blackboard Bold letters)*

<i>To get this</i>	<i>Type</i>
$\frac{G}{C}$	<code>\frac G C</code>
\mathbb{C}	<code>{\Bbb C}</code>
\mathbb{R}^2	<code>{\Bbb R}^2</code>
...	

Remember this as “bb” for blackboard bold, “gm” for german.

- *Word or Word-phrase abbreviations*

A “w” at the start of a shortcut name means it is a word or word-phrase abbreviation.

<i>To get this</i>	<i>Type</i>
Department of Mathematics	<code>wcdm</code>
Department of Physics	<code>wcdp</code>
Euler-Poincaré	<code>wep</code>
...	

- *Formatting features*

<i>To get this</i>	<i>Type</i>
\backslash	<code>\</code>
\newline	<code>\newline</code>
$\hspace{0.2in}$	<code>\hspace{0.2in}</code>
...	

1.3 Examples of Generic (Universal) Simple Expansions

These expansion create partial TeX commands that can be extended or completed using other shortcuts or additional user input. In general, these shortcut names have a trailing “u”, meaning Universal (or Unfinished).

<i>To get this</i>	<i>Type</i>
<code>\frac{</code>	fu
<code>\int</code>	intu
<code>^{</code>	hu
<code>_ {</code>	lu
<code>{\it</code>	hitu
<code>\sqrt{</code>	squ
<code>\vec{</code>	ovu
<code>\overline{</code>	olu
<code>\bar{</code>	obu
<code>\check{</code>	ocu
<code>\dot{</code>	odu
<code>\ddot{</code>	oddu
<code>ch</code>	ohu
<code>\tilde{</code>	ot <u>u</u>
<code>\index{</code>	idu
<code>\mbox{\rm</code>	opnu
...	

1.4 Operator Names in Mathematics

<i>To get this</i>	<i>Type</i>	
<code>\mbox{\rm</code>	opnu	(\LaTeX)
<code>\operatorname{</code>	opnu	($\AMS\TeX$)

The shortcut ‘opnu’ stands for ‘operator name universal’ which is used whenever an abbreviated name is to appear in roman letters within math in-line text or displays. In $\AMS\TeX$ and $\AMS\LaTeX$ this will produce the command `\operatorname{`, and in \LaTeX the equivalent code is `\mbox{\rm`. Several of the standard operator-names are predefined as shortcuts with the prefix ‘op’. For example, ‘oprank’, ‘opcrank’, ‘opsl’, ‘opcscl’ would

produce “rank”, “Rank”, “sl”, and “SL”, respectively.

1.5 Complex Expansions—Templates

This category contains expansions for such things as matrices, commutative diagrams, equation environments, figures, tables, command definitions for the preamble section of a TeX file, etc. Even larger templates are available to produce skeleton versions of complete documents, such as letters and articles. Some examples are given here:

<i>To get this</i>	<i>Type</i>
<code>\begin{equation}</code> <code>\end{equation}</code>	beq eeq
<code>\begin{eqnarray}</code> <code>\lefteqn{ } \nonumber \\</code> <code>& &</code> <code>\end{eqnarray}</code>	lequ
<code>\title{Title of paper}</code> <code>\author{</code> Author1 <thanks{research ...}<="" by="" partially="" supported="" th=""> \\Department of Mathematics \\University of ... \\ \and Author2 <thanks{research ...}<="" by="" partially="" supported="" th=""> \\Department of Physics \\State University of ... } <code>\date{put in custom date; omit for today's date}</code> <code>\maketitle</code></thanks{research></thanks{research>	teaut

1.6 The Importance of Producing Default Files

When dealing with coauthors, or anytime one is communicating documents, it is important to be able to deliver files that are as “plain vanilla” as possible so that the recipient avoids typesetting problems or troubles understanding the \TeX source file for editing purposes. For example, custom definitions (or macros) can seriously interfere with this need and even make otherwise beautifully composed documents difficult for exchange between coauthors. \FasTeX can help with these problems because all the shortcuts are *local* modifications to the keyboard input and do not remain in the actual text of the file created.

1.7 \FasTeX is Universal, Fast, and Accurate

The \FasTeX system is editor, application, and computer platform independent, and as we already indicated, can be used with plain \TeX , $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$, $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\text{\TeX}$, or $\mathcal{L}\text{\TeX}$. The use of \FasTeX will *speed up the typical users typing input by a factor of about 3*.

This speed up is achieved by not only faster inputting, but through the fact that it *avoids simple typing errors* (if \FasTeX is activated, the document will typeset). How many times has a document failed to typeset for you because your finger slipped when typing a backslash alpha or similar \TeX command? This won't happen with \FasTeX !

In addition, \FasTeX is a convenient way of *remembering, storing and learning \TeX commands*. \FasTeX does not conflict with normal English useage; however, versions of \FasTeX suitable for foreign languages require the replacement of certain shortcut names, depending on the language (some shortcut names may be words in certain foreign languages, such as “la” which is the code for “lower (subscript) a”; however, this is also a word in French but not in English).

1.8 You can get \FasTeX off the WEB

\FasTeX is currently available free on the World Wide Web for the Macintosh and UNIX environments. The URL locator is: <http://cds.caltech.edu/~fastex>. The UNIX version differs slightly from the Mac version but the basic operation and the names of the shortcuts are the same for both systems. System independence is another attractive feature when dealing with coauthors who may be using different systems but want to be able to efficiently share information about \FasTeX .

2 A Sample Input

Here is an example of the input sequence one would use for the following equation using \LaTeX .

$$\phi_{\alpha}(x) = \int_0^x \frac{f(t)}{\alpha^2 + x^2} dt \quad (1)$$

The sequence “`\begin{equation} \phi_{\alpha}(x) = \int_0^x \frac{f(t)}{\alpha^2 + x^2} dt`” produces the text in \LaTeX . A brief description of these shortcuts will help the reader to understand how the shortcuts names are a key to the functionality of the \LaTeX system. As a typist becomes familiar with the pseudo-generic system of naming shortcuts, typing \LaTeX documents becomes easier and faster to do.

<i>To get this</i>	<i>Type</i>	<i>Description</i>
<code>\begin{equation}</code>	beq	begin equation
<code>\phi</code>	xph	greek letter phi
<code>_{\alpha}</code>	lxa	subscript (or lower) α
<code>(x)</code>	ox	‘of x ’ (x in parentheses)
<code>=</code>	eq	equal
<code>\int</code>	intu	set up the integral
<code>_0</code>	l0	subscript 0
<code>^x</code>	hx	superscript x
<code>\frac{</code>	fu	prepare for a fraction
<code>f</code>	f	f in numerator
<code>(t)</code>	ot	‘of t ’ (t in parentheses)
<code>}{</code>	fof	forward fraction (go to the denominator)
<code>\alpha</code>	xa	greek letter alpha
<code>^2</code>	sq	square it (or use “h2”)
<code>}</code>	eb	end brace (of fraction)
<code>\,dt</code>	spdt	thin space dt
<code>\end{equation}</code>	eeq	end equation

3 The Macintosh Version of FasT_EX

The Macintosh version of FasT_EX uses the shareware control panel *TypeIt4Me*, and the FasT_EX shortcut files. You will need some version of system 7 to use FasT_EX on the Mac.

You can easily locate what you need from the FasT_EX WEB site (URL: <http://cds.caltech.edu/~fastex>). Besides the documentation, you need to collect *TypeIt4Me* (unpacks using *Stuffit*), and the FasT_EX files (you decide which flavor of T_EX you want).

3.1 Installation

After collecting the software off the WEB or using the FasT_EX disk (Macintosh users can purchase a disk containing everything needed from TUG or the authors):

1. Copy the *TypeIt4Me* control panel to the control panels folder.
2. Copy the shortcut files to a convenient folder or on the desktop available for *TypeIt4Me* to open.
3. Restart the machine.
4. Select, for example, `latex.scm` from the *TypeIt4Me* menu. Open your favorite editor (Textures, Alpha, etc.) and type a shortcut such as “xa-spacebar” to test the system.

3.2 A little about *TypeIt4Me*

TypeIt4Me is a shareware product of Ricardo Ettore and if you really use it, you are expected to support the shareware idea and send him \$30US. It is worth every penny! It is great for many purposes besides T_EX: email addresses, commonly typed phrases, etc.

After restarting your machine, you will see the *TypeIt4Me* icon in the top left corner of your screen. Here are a few of the menu items you will find under the icon:

1. A help manual under *About TypeIt4Me* explains the workings of *TypeIt4Me* in detail.

2. To add an entry of your own, type it somewhere (such as your $\text{T}_{\text{E}}\text{X}$ editor), copy it to the clipboard, and call up the *Add an entry* item (for us, this is equivalent to the hot key “Shift-Option-C”) and insert your shortcut name in the window. Now that shortcut is available!
3. To edit entries, call up this menu item (for us, it is the hot key “Shift-Option-E”). The dialog box that results is self-explanatory; you can, as the name suggests, edit any existing entry, change shortcut names, etc.
4. The preferences choice allows you to set hot keys, select what keys trigger the shortcut besides the spacebar, let you decide if you want the trigger included in the expansion, etc.
5. Below this you find date and time items.
6. The final menu items tell you what file of shortcuts you want open. On the Macintosh there is a limitation of having one file open at a time and each file is limited to 2500 shortcuts (this is due to resource editor limitations and is normally not a problem). You probably want to have your addresses, email addresses, etc., in a different file from the $\text{T}_{\text{E}}\text{X}$ shortcuts. *FasT_EX* provides different files for \LaTeX , $\text{A}_{\text{M}}\text{S-}\text{T}_{\text{E}}\text{X}$, and $\text{A}_{\text{M}}\text{S-}\text{\LaTeX}$; again, you choose the one you are working with for the particular document.

If for some reason you do not want *TypeIt4Me* to activate, you simply turn it off with the appropriate hot key (which we have set at Shift-Option-O)—that is an “oh”. When off, the *TypeIt4Me* icon appears dimmed. For example, if you are typing a passage in French, you probably should turn off *TypeIt4Me*.

3.3 A Note about \$

The dollar sign is of course the most common key needed for typesetting mathematical expressions in $\text{T}_{\text{E}}\text{X}$. Because of its frequent use, we have made the single shortcut “d” for it. Surprisingly, this causes very few problems. Of course, as with all shortcuts, this does not cause a problem with a “d” occurring in a word. If you want to type a literal “d” in the middle of a formula and do not want it to come out with a dollar sign, you can use the shortcut “sd” which produces a literal “d”, or you can type a “d” and (at least on the Macintosh) hold the shift key down when typing the spacebar, which keeps *TypeIt4Me* temporarily from activating.

3.4 A Sample Creation of a Shortcut

Let us suppose that you want to create a shortcut for a combination that occurs frequently in your work. For example, assume it is a $dx dy$ occurring in a double integral. First of all, decide on what spacing you want in

the mathematics. For instance, let's say that the literal keystrokes you want are `\;dx\,dy` and, consistent with the FasTeX scheme of naming, you chose the shortcut name "spdxdy". Then you would create this shortcut as follows:

1. Type `\;dx\,dy` in any text editor.
2. Select and copy it to the clipboard.
3. Select "add an entry" from the *TypeIt4Me* menu (or hit "Shift-Option-C").
4. Type in the shortcut name spdxdy.

That's it! Now that shortcut is available.

4 The UNIX version of FasTeX

There is a version of the Macintosh FasTeX engine (*TypeIt4Me*) that runs on most flavors of UNIX. Called *scedit*, the UNIX version is based on the Expect program of Don Libes. Though not identical to the Mac version of *TypeIt4Me*, *scedit* attempts to include all of the essential functionality of the Mac version, with a few concessions to differences in the operating systems.

The shortcut files that make up FasTeX are necessarily different between the Mac and UNIX, but both are derived from a common base file, so portability between Mac and UNIX introduces no incompatibilities.

The *scedit* program works by isolating an editor program of the user's choice from the keyboard and screen (window). All interaction between the user and the editor is intercepted by the *scedit* program, which performs the shortcut expansions.

An advantage of this approach is that shortcut expansion is independent of the windowing environment under which *scedit* runs, so for example, *scedit* will work over a telnet or modem connection. A disadvantage is that the editor program run by *scedit* can not respond to mouse events.

One benefit to using *scedit* over expansion facilities that may be native to a particular editor program is that the same shortcut files will work across different editor programs, regardless of what kind of expansion facility the editor may have, if any. For example, both vi and emacs have native expansion (abbreviation) capabilities that

are incompatible with each other; but *scedit* and FasT_EX works with either editor by changing a few configuration parameters.

The *scedit* program has some additional capabilities over the Mac *TypeIt4Me* program. One significant feature is that *scedit* can have many shortcut files active at once, unlike *TypeIt4Me*, which allows only one active shortcut file.

4.1 System Requirements

The *scedit* program will run on most versions of UNIX. The main requirement is that the UNIX platform have a recent version (5.7.0 or later) of the Expect program installed.

To run FasT_EX on UNIX you need:

<i>Program</i>	<i>Available from</i>
Expect/Tcl	ftp://ftp.cme.nist.gov/pub/expect
<i>scedit</i>	http://cds.caltech.edu/~fastex ftp://cds.caltech.edu/pub/fastex
FasT _E X files	http://cds.caltech.edu/~fastex ftp://cds.caltech.edu/pub/fastex

4.2 Installation

The UNIX version of FasT_EX can be installed in your personal file space or in a system area. Either approach will work, and the choice depends mainly on whether you want to maintain the FasT_EX system yourself or leave that up your computer's system administrator.

There are four components to the UNIX FasT_EX system which need to be installed in four distinct areas. These are the *scedit* program, the FasT_EX shortcut files, the documentation and the public, default configuration file. Each user may also have a personal configuration file which overrides the public one—see Customization below.

Installation consists of editing the Makefile that came with the FasT_EX distribution to reflect the installation

directories you chose, then running the command `make install`.

Within the Makefile, the variable `INSTALLDIR` should be edited to contain the name of a directory containing the sub-directories `bin`, `lib/scedit` and `man`. If any of these directories or sub-directories does not exist, you should create them first and set their permissions to allow the kind of access you think appropriate. Also, edit the variable `EXPECT` to contain the name of the directory containing the Expect executable. When you complete these two edits you may run `make install`.

The following example shows a typical installation of UNIX FasTeX into a personal area. To install into a system area just change the names of the destination directories (and be sure you have write permission in the affected system directories).

```
mkdir /accts/fil/fastex
chmod 755 /accts/fil/fastex
mkdir /accts/fil/fastex/bin
chmod 755 /accts/fil/fastex/bin
mkdir /accts/fil/fastex/lib
chmod 755 /accts/fil/fastex/lib
mkdir /accts/fil/fastex/lib/scedit
chmod 755 /accts/fil/fastex/lib/scedit
mkdir /accts/fil/fastex/man
chmod 755 /accts/fil/fastex/man
mkdir /accts/fil/fastex/man/man1
chmod 755 /accts/fil/fastex/man/man1

chmod 644 Makefile
vi Makefile
...
INSTALLDIR = /accts/fil/fastex
EXPECT = /usr/local/bin/expect
...
make install
```

At this point FasTeX has been installed and is ready to use. You may wish to create an alias for the *scedit* executable so that you do not have to type its full path name when you invoke it. Alternatively, you can place a symbolic link to the *scedit* executable into one of the `bin` directories listed in your shell's `PATH` variable. With this latter approach, you could make *scedit* available to the general user community by placing the symbolic link into a system `bin` directory, such as `/usr/local/bin` if that is where locally added programs are kept on your

system:

```
ln -s /accts/fil/fastex/bin/scedit usr/local/bin
```

Another approach is to place the symbolic link into your own personal `bin` directory if you wanted access to *scedit* to be semi-private.

4.3 Customization

The file *sceditrc*, which was installed into the `lib` directory during the installation procedure, contains customization parameters for *scedit*. These parameters control several aspects of the way *scedit* operates, and take effect for every user of *scedit*.

An optional file, *.sceditrc* in each user's home directory, can also contain *scedit* parameter values. These will override the values in *sceditrc* located in the `lib` directory. In this way each user can specify personal customization of *scedit*.

The parameters include: `FASTEX_FILES` which contains a list of shortcut file names to use by default; `FASTEX_PATH` which lists the directories to be searched for shortcut files listed on the command line; `NAME_SUFFIX` which contains the list of characters that will serve to activate shortcut expansion; `REDRAW` which holds the sequence of characters that will make your editor redraw the screen; etc. The *scedit* documentation discusses these and the other parameters in more detail.

Standard (Bourne) shell syntax can be used in the customization files. This allows a single customization file to include parameters for a variety of different editors. The default *sceditrc* file that comes with the *scedit* distribution shows how to represent separate customizations for `vi` and `emacs`, selected by examining the UNIX standard environment variable `EDITOR`.

4.4 Some specific differences between the *scedit* and *TypeIt4Me* versions of FasTeX

Because the UNIX version of FasTeX does not assume a windowing environment, some of the user interaction with *scedit* is necessarily different than the user interaction with *TypeIt4Me*. For example, *scedit* uses special escape keystrokes to perform file management operations that would be done with pull-down menus on the Mac. Most of these meta-level operations are introduced with the tilde character (`~`).

For example, the escape command `~e1`. would start a new editing session on the main shortcuts file. The display and keyboard will now be attached to this new editor session. The old editor session is waiting undisturbed in the background. When you exit this new editor session, the old one is resumed. Viewing the shortcuts in this way is helpful when you forget the name of a shortcut and want to look it up; you can then use the full search capabilities of your editor to help you locate shortcut definitions within a potentially large shortcut file.

Any changes you make to a shortcut file when viewed in this way will be incorporated back into the active *scedit* session when you exit the viewing editor. So, for example, to add a new shortcut while editing a document, use (`~ef`) to view/edit your “first” shortcut file (the “first” file is the one that *scedit* reads before the main shortcut file), add the new shortcut to the file using the editor, then write and quit the editor. The new shortcut will now be available to you as you resume editing your document.

A list of all the escape commands is available through the escape command (`~h`). To enter a actual tilde character into the text of your document, double the tilde (`~~`).

The usual UNIX job control mechanism can also be used to suspend and resume the entire *scedit* session.