Visual Editor (vi)

Quick Reference Guide
UCLA Math Series V–1
Version 1.3 of October 4, 1994

The Standard UNIX Full Screen Editor

How to Use

The Unix Visual Editor **vi** is a full-screen display oriented interactive editor with a rich set of commands. The editor has two modes, a command mode and an insert mode. You can always hit the Escape key to exit the insert mode and enter the command mode if you are not sure what mode you are in. An excellent interactive tutorial is available on the system and can be accessed by typing the command `learn vi`.

Notation

^x	Hold down `[CTRL]` key while typing (**x**)
[n]	Optional number of repetitions or line numbers preceding a command
$\boxed{\text{BS}}$	Backspace key
$\boxed{\text{Esc}}$	Escape key
$\boxed{\text{Ret}}$	Carriage Return key

Special Keys

:	In command mode, **:** is a prefix for file and option commands, which appear at the bottom of the screen. Finish the command with $\boxed{\text{Ret}}$ or abort with ^**c**.
$\boxed{\text{Esc}}$	Leave input mode, back to command mode

The arrow keys work on ANSI and X terminals, but not on Wyse or TVI.

Entering the Editor

Note: Follow entry with $\boxed{\text{Ret}}$

vi *file*	Edit *file*
vi `+n` *file*	Edit at line `n` in *file*
vi `+` *file*	Edit at last line in *file*
vi *file1 file2* . . .	Edit *file1*; *file2*; (After *file1* use `:n` to edit each remaining file.)
vi `+/pat` *file*	Search for and edit at `pattern` in *file*
vi `-r`	List any saved files after editor crash
vi `-r` *file*	Recover and edit saved *file*. (Must be executed on the same *host* computer where the file was saved.)

Leaving the Editor

:q$\boxed{\text{Ret}}$	Quit vi when no changes have occurred since last write
:q!$\boxed{\text{Ret}}$	Quit vi, discard changes since last write
:wq$\boxed{\text{Ret}}$	Write, quit (exit vi, saving changes)
ZZ	Write, quit (exit vi, saving changes)

Creating Text

a*text*	Append *text* after cursor
i*text*	Insert *text* before cursor
A*text*	Append *text* at end of line
I*text*	Insert *text* before first non-blank
o*text*	Open new line below cursor
O*text*	Open new line above cursor

Note that *text* can be many lines long. You should press $\boxed{\text{Esc}}$ after you finish inputing *text* to return to the command mode.

Editing in Insert Mode

^**W**	Erase last word during an insert
^**V***x*	Insert non-printing character (*x*)
$\boxed{\text{Esc}}$	End insertion, enter command mode
$\boxed{\text{BS}}$	Erase last character

Editing in Command Mode

J	Join lines
s*text*$\boxed{\text{Esc}}$	Replace one character with *text*
S*text*$\boxed{\text{Esc}}$	Substitute the entire line with *text*
cc*text*$\boxed{\text{Esc}}$	Replace a whole line with *text*
C*text*$\boxed{\text{Esc}}$	Change from cursor to end of line
cw*text*$\boxed{\text{Esc}}$	Change next word to given *text*
r*x*	Replace one character with *x*
R*text*$\boxed{\text{Esc}}$	Overwrite multiple characters with *text*
i$\boxed{\text{Ret}}\boxed{\text{Esc}}$	Split a line before the cursor
a$\boxed{\text{Ret}}\boxed{\text{Esc}}$	Split a line after the cursor

Cursor Positioning in a File

[n]^**F**	Scroll down screenful
[n]^**B**	Scroll back screenful
[n]^**D**	Scroll down (default is half screen)
[n]^**U**	Scroll up (default is half screen)
*[n]***G**	To line `n` (default is last line of file)
[n]`/pat`$\boxed{\text{Ret}}$	Put cursor on next line matching `pattern`
[n]`?pat`$\boxed{\text{Ret}}$	Place cursor on previous line matching `pattern`
*[n]***n**	Repeat last / or ? search command
*[n]***N**	Reverse last / or ? search command

Cursor Positioning on Screen

*[n]***H**	Top of display, or nth line
*[n]***L**	Last line, or nth from bottom

Line Positioning

0	Go to beginning of line
*[n]***$**	Go to end of line
[n]$\boxed{\text{Ret}}$	Place cursor *[n]* lines down
*[n]***j** or ↓	Go to next line
*[n]***k** or ↑	Go to previous line

Character Positioning

*[n]***l** or →	Next character, same line
*[n]***h** or ←	Previous character, same line
[n] |	Move to column number *[n]*

Word Positioning

*[n]***w**	Move forward to beginning of word
*[n]***b**	Move back to beginning of word
*[n]***e**	Move forward to end of word
*[n]***W**	Move forward to beginning of word. Punctuation ignored
*[n]***B**	Move back to beginning of word. Punctuation ignored
*[n]***E**	Move forward to end of word. Punctuation ignored

Deleting Text

D	Delete from cursor to end of line
*[n]***x**	Delete a character
*[n]***X**	Delete character to the left of cursor
*[n]***dw**	Delete the next word
*[n]***dd**	Delete current line

Moving, Copying, Yanking

*[n]***Y**	Yanks a copy of the current line into memory area
*[n]***yy**	Same as **Y**
y *[n]***w**	Yanks a copy of words into memory area
p	Puts the last deleted or yanked text after or below cursor
P	Puts the last deleted or yanked text before or above cursor

Undoing, Redoing

u	Undo last change
U	Undo all changes to current line
” *[n]***p**	Put back *n*th most recent delete (where *n* is between 1 and 9, inclusive).
•	Repeat last append, insert, open, delete, change or put command

Global Searches and Changes

Note: Follow entry with $\boxed{\text{Ret}}$

:g/*text***/p**	Print all lines containing *text*
:g/*text***/nu**	Print all lines and line numbers containing *text*
:*[m]*,*[n]*s/*text*/*newtext*/g	Find all *text* in lines *m* through *n* and replace with *newtext*
:*[m]*,*[n]*s/*text*/*newtext*/gc	Same, ask before each replacement

Manipulating Files

Note: Follow entry with $\boxed{\text{Ret}}$

:r *file*	Insert *file* contents after current line
:w	Write changes to current file
:w *file*	Write everything to *file*
:w! *file*	Overwrite existing *file*
:w>>*file*	Append at end of *file*
:*[m]*,*[n]***w** *file*	Write lines *m* through *n* to *file*
:*[m]*,*[n]***w!** *file*	Same, overwriting existing *file*
:*[m]*,*[n]***w>>***file*	Add lines *m* through *n* to end of *file*

Editing Different Files

Note: Follow entry with $\boxed{\text{Ret}}$

:e!	Re-edit file discarding changes since last write
:e *file*	Edit a different *file*
:e! *file*	Discard changs since last write and edit a different *file*
:n	Edit next file in list when vi is called with more than one file
:e#	Edit alternate file when two files are being edited

Miscellaneous Operations

~	Switch character from lowercase to uppercase and vice versa
xp	Switch two characters' position in line
^G	Show current file name and line number (same as **:f**)
^L	Redraw the screen
:set nu	Show line numbers
:set nonu	Do not show line numbers
:set wm=*x*	Define margin for automatic line wrap
:set all	Show all pre-set options
:stop	Stop the editor and return to shell. (Type `fg` to return to **vi**)
:sh	Invoke a shell; type `exit` to go back to vi
:!*cmd*	Run Unix command *cmd*
!!*cmd*	Run Unix command *cmd*; Output replaces current line