

VI EDITOR ABBREVIATED

(Courtesy of Mark E. Donaldson)

Invoking vi

vi can be invoked using the following command lines:

vi (command mode)
ex (command mode)
view (read only mode)
input (input mode)

Using Modes

To switch from *command mode to input mode* type **i**. To switch from *input mode to command mode*, hit **ESC**. You must be in command mode to save or quit a file.

Once the file is opened, you are in command mode. **From command mode, you can:**

- Invoke insert mode.
- Issue editing commands.
- Move the cursor to a different position in the file.
- Invoke **ex** commands.
- Invoke a UNIX shell.
- Save or exit the current version of the file.

The following *commands invoke insert mode*:

a Append after cursor.
A Append at end of line.
c Begin change operation.
C Change to end of line.
i Insert before cursor.
I Insert at beginning of line.
o Open a line before current line.
O Open a line below current line.
R Begin overwriting text.
s Substitute a character.
S Substitute entire line.

Using Commands

To enter a command in **vi**, you must preface it with a semicolon **:**.

Save and Quit

To save a file without quitting, type **:w filename**.

To save a file and quit, type **:wq filename**

To quit a file without saving, type **:q!**.

Common Cursor and Scrolling Commands

0 (zero) Moves cursor to beginning of current line.
b Moves cursor to beginning of current word.
) Moves cursor to end of document.
\$ Moves cursor to end of current line.
e Moves cursor to end of current word.
w Moves cursor to beginning of next word
nG Moves cursor to beginning of line n.

H Moves cursor to beginning of files.
G Moves cursor to beginning of last line in file.

Edit Commands

Recall that **c**, **d**, and **y** are the basic editing operators.

a Append after cursor.
A Append to end of line.
i Insert before cursor.
I Insert at beginning of line.
o Open a line below cursor.
O Open a line above cursor.
ESC Terminate insert mode.
CTRL-J Move down one line.
RETURN Move down one line.
CTRL-I Insert a tab.
CTRL-T Move to next tab setting.
BACKSPACE Move back one character.
CTRL-H Move back one character.
CTRL-U Delete current line.
CTRL-V Quote next character.
CTRL-W Move back one word.

The last four control characters are set by **stty**. Your terminal settings may differ.

Changing and Deleting Text

cw Change word.
cc Change line.
c Change text from current position to end of line.
dd Delete current line.
ndd Delete n lines.
D Delete remainder of line.
dw Delete a word.
d} Delete up to next paragraph.
D^ Delete back to beginning of line.
d/pat Delete up to first occurrence of pattern.
dn Delete up to next occurrence of pattern.
dfa Delete up to and including a on current line.
dta Delete up to (not including) a on current line.
dL Delete up to last line on screen.
dG Delete to end of file.
p insert last deleted text after cursor.
P Insert last deleted text before cursor.
rx Replace character with x.
R text Replace text beginning at cursor.
s Substitute character.
4s Substitute four characters.
S Substitute entire line.
u Undo last change.
U Restore current line.
x Delete current cursor position.

VI EDITOR ABBREVIATED

(Courtesy of Mark E. Donaldson)

X Delete back one character.
5X Delete previous five characters.
. Repeat last change.
~ Reverse case.

Copying and Moving

y Copy current line to new buffer.
yy Copy current line.
“xyy Yank current line to buffer x.
“xd Delete into buffer x.
“Xp Delete and append into buffer x.
“xp Put contents of buffer x.
y|] Copy up to next section heading.
ye Copy to end of word.

Saving and Exiting

Writing a file means saving the edits and updating the file's modification time.

ZZ Quit vi, writing file only if changes.
:x Same as **ZZ**.
:wq Write and quit file.
:w Write file.
:w file Save copy to file.
:n1, n2w file Write lines n1 to n2 to new file.
: n1, n2w >> file Append lines n1-n2 to existing file.
:w! Write file (overriding protection).
:W! file Overwrite file with current buffer.
:w %.new Write buffer named file as file. new.
:q Quit file.
:q! Quit file (discarding edits).
Q Quit vi and invoke ex.
:vi Return to vi after **Q** command.
:e file2 Edit file2 without leaving vi.
:n Edit next file.
:el Rtn to ver current file at last write.
:e# Edit alternate file.
% Current filename.
Alternate filename.

Accessing Multiple Files

:e file Edit another file, becomes alternate.
:e! Restore last saved version of current file.
:e + file Begin editing at end of file.
:e +n file Open file at line n.
:e # Open to previous position in alternate file.
:ta tag Edit file at location tag.
:n Edit next file.
:n! Forces next file.
:n files Specify new list of files.
CTRL-G Show current file and line number.
args Display multiple files to be edited.
rew Rewind list of multiple files to top.

Interacting With UNIX

:r file Read in contents of file after cursor.
:r ! command Read in output from command after current line.
:nr ! command Like above, but place after line n.
:! command Run command, then return.
! object command Send buffer object to UNIX command; replace with output.
:n1, n2! command Send lines n1 - n2 to command; replace with output.
n!! command Send n lines to UNIX command; replace with output.
!! Repeat last system command.
:sh Create subshell; rtn to file with EOF.
CTRL-Z Suspend editor, resume with fg.
:so file Read & execute commands from file.

Macros

:ab in out Use in as abbreviation for out.
:unab in Remove abbreviation for in.
:ab List abbreviations.
:map c sequence Map character c as sequence of commands.
unmap c Disable map for character c.
:map List characters that are mapped.
:map! c sequence Map character c to input mode sequence.
:unmap! c Disable input mode map (you may need to quote the character with CTRL-V).
:map! List characters that are mapped to input mode.

The following characters are unused in command mode and can be mapped as user defined commands.

Letters: **g K q V v**
 Control keys: **^A ^K ^O ^T ^W ^X**
 Symbols: **_ * \ =**

Note: The = is used by vi if Lisp mode is set.