

August 5, 1996

This is a summary of the most common **vi** commands. Consult the Sun Microsystems documentation for complete reference on these and other **vi/ex** commands and command options.

Parameters are shown in *italics* and must be replaced as shown in the command description. Text in **Courier** are commands and must be entered as shown.

Entering and Leaving vi

vi *filename* edit file *filename*
vi -r *filename* recover file *filename* after crash
ZZ exit from **vi**, saving changes
:w write changes to disk
:wq write changes to disk and quit
:q quit
:q! quit, discard changes
:sh shell out to Unix, return with **exit**

VI Modes

Command

Normal and initial mode. Commands will not be printed to the screen as they are executed. **Esc** returns to this mode from any other mode. Partial command-mode commands can be aborted by pressing **Esc**.

Insert

Entered by using a **I A I o O c C s S** or **R**. Subsequent characters are inserted into the file. A status message such as "INSERT MODE" will show in the bottom right of the screen. Press **Esc** to return to command mode.

Lastline

Reading input for **: / ?** or **!**. Press **Return** or **Esc** to execute the command.

Repeated Command Execution

* Commands marked with an asterisk can be repeated by entering the number of repetitions before the command. See "Repeated Command Execution" for details.

Many commands in **vi** may be repeated by typing the number of repetitions before the command. Commands that affect a line will affect multiple lines if repeated and commands that affect a character will affect multiple characters if repeated. For example, to delete 5 characters in a row, type **5x**. Repeatable commands are marked with an asterisk (*) after the command explanation.

Also, the last command can be repeated after its execution with a period (.).

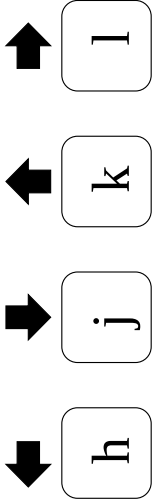
File Manipulation

:r *filename* read in file *filename*
:w *filename* write to file *filename*
:w! *filename* overwrite file *filename*
:! *command* run *command*, then return to **vi**
:e! discard changes, re-edit file
:n edit next file if multiple files are specified when starting **vi**
:f show *filename* and line number information
Ctrl-G show *filename* and line number information

Moving

h left one character *
j down one line *
k up one line *
l right one character *
Return first non-blank character of the next line *
Ctrl-F forward one screen *
Ctrl-B backward one screen *
Ctrl-D forward half screen *
Ctrl-U backward half screen *
H top line of screen
M middle line of screen
L last line of screen
:n go to line number *n*
nG go to line number *n*
G go to last line
^ first non-blank character on line
o first character on line
\$ last non-blank character on line
x | move to column *x*

Cursor Movement Keys



Moving Through Text

w next word forward *
b previous word *
e end of word *
) next sentence *
(previous sentence *
} next paragraph *
{ previous paragraph *
W forward to next blank-delimited word *
B previous blank-delimited word *
E end of blank-delimited word *

Inserting

Each command in this section causes **vi** to enter insert mode. To leave insert mode, press **Esc**.

I insert text before cursor
a append text after cursor
I insert before first non-blank on line
A append text at end of line
o open and insert text on line below
O open and insert text on line above

Editing during Insert Mode

Esc exit insert mode, enter command mode
Ctrl-H erase last character
Ctrl-W erase last word
Ctrl-C interrupt, terminates insert
Ctrl-D backtab over auto-indent

Text Alteration

End alterations with **Esc** when using **R s S cw cc c**) or **c**(

r *c* replace single character with *c*
(does not enter insert mode) *****
R replace characters
S substitute text under character *****
S substitute entire line *****
CW change word starting at cursor
CC change line *****
C change from cursor to end of line *****
c change sentence beginning at cursor
c change from sentence beginning to cursor
>> right shift line a tab stop *****
<< left shift line a tab stop *****
J join lines *****
~ toggle case of letter, move forward one character *****

Special Alteration

Each of the following commands affect only one line. To affect multiple lines, add line numbers (*targets*) to the beginning of the command as follows:

n, *m* affects line numbers *n* through *m*
leaving either *m* or *n* blank (") means substitute current line number

\$ represents the last line of the file
% affects all lines, same as **1, \$**

: *target* **s** / *oldstring* / *newstring* / *options*
substitute *oldstring* with *newstring*. *Options* are: **g** to replace all occurrences within one line, **c** to confirm changes, and **p** to print changed lines

Example of global replacement:
:%s/old/new/g

& repeat last substitute command
: *target* **co** *line-number*
copy line(s) and put duplicate after *line-number*
: *target* **m** *line-number*
move line(s) after *line-number*

***** Commands marked with an asterisk can be repeated by entering the number of repetitions before the command. See "Repeated Command Execution" for details.

Deleting, Copying, and Moving Text

After each yank operation, the affected text is placed into the last-change buffer, replacing the buffer's previous contents. Alternatively, yanked text may be placed into a named buffer *x* with the "**x****y**" or "**x****yy**" command. Yanked text is retrieved with the "**x****p**" and "**x****P**" commands. Yank buffers may be named **a** through **z**.

Deletions are placed into a deletion buffer. Deletion buffers are numbered **1** (last deletion) through **9** (oldest deletion) and may be retrieved with the "**x****p**" or "**x****P**" command. See the "Correction" section for more undelete commands.

x delete character on cursor *****
X delete character before cursor *****
dw delete word *****
de delete word, leaving punctuation *****
dd delete line *****
D delete from cursor to end of line
y yank (copy) a character or characters into buffer *****
yy yank (copy) a line or lines into buffer *****
Y yank (copy) a line or lines into buffer *****
p put buffer after current character (after **y**)
or line (after **yy**)
P put buffer before current character (after **y**)
or line (after **yy**)
"**x****y** yank (copy) into named buffer *x* *****
"**x****Y** yank (copy) into named buffer *x* *****
"**x****p** put buffer after current character or line from buffer *x* (*x* is a through *z* or 1 through 9)
put buffer before current character or line from buffer *x* (*x* is a through *z* or 1 through 9)

Correction

u undo last command
U undo all changes on current line
"**n****p** retrieve the *n*'th previous deletion (*n* is 1 through 9)
repeat last non-movement command
"**1pu.u.u.u.** walk through deletion buffers 1 through 5, and restore buffer 5 to current line
restore deletion buffers 1 through 5, restoring each to the previous line
"**1P.....**

Saving Your Place

m *t* mark current cursor position with character tag *t*
' *t* move to first non-blank character in line with tag *t*
'' move to first non-blank character in line of previous change
` *t* move to position marked tag *t*
`` return to place after cursor motion caused by last command

Text Searching

/ *pattern* search for next occurrence of *pattern*
/ repeat last search in the forward direction
? *pattern* search for previous occurrence of *pattern*
? repeat last search in the reverse direction
n repeat last / or ?
N repeat last / or ? in the opposite direction
f *x* find next occurrence of character *x* *****
F *x* find previous occurrence of character *x* *****
; repeat last **f** or **F** *****

Set Commands

Options without parameters act as toggles. For example, type **:set number** to show line numbers and **:set nonumber** to remove them.

:set all show settings of all options
:set number display line numbers
:set showmode place insert mode indicator in bottom-right hand corner
:set ignorecase ignore case while searching with **** and **?**
:set wrapscan wrap text searches across the end of the file
:set tabstop=*k* set tabs every *k* characters
:set wm=*k* wraps characters to next line at column *k*
:set showmatch show matching sets of {, (,), and } as typed.
:set autoindent indent next line if previous line is indented.