Quick reference for the *vi* editor

Nathan Paul Simons

2011-01-30 16:40

Abstract

Describes common operations in the *vi* text editor.

This publication is available in Web form[1] and also as a PDF document[2]. Please forward any comments to **tcc-doc@nmt.edu**.

Table of Contents

1. What is *vi*?

vi's power is in the fact that it was designed for powerful text manipulation and quick movement using only a keyboard. It is also pretty standard (although I won't guarantee that all commands listed here will work on all variants) and very small.

These may not be all the commands or combinations of commands for vi. I tried to include here only what I consider the most standard and most useful *vi* commands. I can't guarantee that all these commands will work in all *vi* versions.

Most of these commands are for command mode (also known as escape mode; just hit the *esc* key a couple of times and you'll be in command mode; usually once will do it, unless you have given it certain commands).

The carat (**^**) symbol indicates a control character. For example, to get **^F** ("control-F"), hold down the *control* key and then press **F**.

When a command says *can be enumerated*, that means that you can repeat that command a certain number of times if you precede it by the number of times you want it to be repeated. This has different effects with different commands: "**3dd**" will delete three lines and put them all into the buffer (clipboard, to you Windows people). "**3p**" will put three instances of whatever's in the buffer after the cursor. So if you use "**3dd**" followed by "**3p**", you will end up with nine lines of text where there were three.

---

[1] http://www.nmt.edu/tcc/help/pubs/vi/
[2] http://www.nmt.edu/tcc/help/pubs/vi/vi.pdf

Almost all commands can be enumerated. Unless otherwise stated, "current word" or "current character" usually refers to the word under the cursor (the word that contains the letter under the cursor) and the character under the cursor.

A lot of special characters are used for commands in *vi*, but there is one that will get mentioned only here: the colon (":"). Use it to go to the *special* or *file command mode* where you can do things like saving files. All commands that are used in this mode are preceded by a colon.

2. Starting *vi*

To edit one or more files:

```
vi f₁ f₂…
```

where f_1 is the name of the first or only file, f_2 is the name of the second file, and so on.

> **Note**
>
> Most *vi* versions will not recognize a change in permissions, so if you start editing a file to which you didn't have write permission, you won't be able to save changes (except to another file) even if you are root, or change the permissions while still editing it in *vi*. Also, it's best to give *vi* a file name upon invocation, because if you don't you will have to tell it the file name every time you want to save it.

3. Moving around

You can use a digit (1-9) followed by an "arrow key" to jump that many lines or letters in that direction—in other words, arrow key commands can be enumerated.

h or ←	Cursor left
l or →	Cursor right
j or ↓	Cursor down
k or ↑	Cursor up
^F or *PageDown*	Move down one page.
^B or *PageUp*	Move up one page.
^U	Move up half a page.
^D	Move down half a page.
^E	Move the screen down one line.
^Y	Move the screen up one line.
+	Go to the beginning of the next line.
-	Go to the beginning of the previous line.
0 (zero)	Go to the beginning of the current line.
$	Go to the end of the current line.
}	Go to the next blank line.

Quick reference for the vi editor

New Mexico Tech Computer Center

{	Go to the previous blank line.
[[	Go to the beginning of the file.
]]	Go to the end of the file.
''	Switch between beginning of file and current line.
b	Go back one word or punctuation mark.
B	Go back one word, including punctuation marks.
e	Go to the end of the next word or punctuation mark.
E	Go to the end of the next word, including punctuation marks.
gg	Go to the top of the file.
G	Go to the end of the file.
^H	Go back one character, including newlines.
H	Go to the top of the current page.
^P	Go up one line.
L	Go to the bottom of the current page.
w	Go to the beginning of the next word or punctuation mark.
W	Go to the beginning of the next word, including punctuation marks.
^Z	Suspend *vi*; use the **fg** command to resume.
ZZ	Save and exit.

4. Searching

)	Go to the beginning of the next English line (i.e., sentence-ending punctuation followed by white space and a capital letter).
(	Sort of like above only backwards, and doesn't get every sentence.
*	Go to the next occurrence of the current word.
#	Search backwards for the current word.
/	Do a search (i.e., "/myword") and *enter* will find the next occurrence after the cursor of "myword". Pressing **n** will find the next instance after that, and pressing **N** will find the previous instance. This method of searching does loop, so if you are planning on finding the last occurrence of a word by holding down **n**, you may wait a while.
?	Repeats the previous "**/**" search.

5. Editing files

back-space	Delete the character before the cursor.
del	Delete the character under the cursor.
i	To insert text under the cursor (insert mode).
I	To insert text under the cursor starting from the beginning of the current line (insert mode).
a	Insert starting after current character.

A	Insert starting at the end of the line after the last character.
cc	Delete the *entire* line and go into insert mode.
cw	Delete the current word and go into insert mode.
C	Delete to the end of line and go to insert mode.
Y	Yank (copy current line to buffer). If you give this a number before it, it will "yank" the specified number of lines after the cursor (including current line the cursor is on). Many commands also have this enumerated functionality.
dd	Delete (cut) the current line. Can be enumerated.
dw	Delete (cut) the current word. Can be enumerated.
D	Delete current line (but leave the newline).
J	Join the next line with the current line.
o	Insert after the current line.
O	Insert before the current line.
p	Put (paste) the lines in the buffer (from **Y**, **dd**, or **x**) after the current line. Can be enumerated.
P	Put (paste) the lines in the buffer before the current line. Can be enumerated.
r*c*	Overwrite with character **c**, replace the current character.
R	Change to replace (overwrite) mode.
s	Delete the current character and go to insert mode.
S	Delete the current line and go to insert mode.
u	Undo the last operation.
U	Switch between undoing and redoing the last edit.
v	Visual: start the selection for a "**d**" (delete) or "**Y**" (yank). Type "**v**", then move in a direction using one of the movement keys to see what you have selected.
V	Visual line: select the current line for delete or Yank.
x	Delete (cut) the character under the cursor. Can be enumerated.
X	Delete (cut) the character after the cursor. Can be enumerated.
~	Capitalize the current character.
==	Unindent the current line (type the equal sign twice).
.	Repeat the last command.
F1	Display online help.

6. Colon (":") commands

:w	Write to file; defaults to the file name you gave it on invocation. If you didn't give it a file name on the command line, it will gripe.
:w *filename*	Save the file to **filename**.
:q	Quit. In *vim*, also exits from help.
:q!	Quit without saving.
:wq	Write to the file and quit (see also "**ZZ**").

:r *file-name*	Read in file (handy for "cut and paste" between two different files).
:e	Exit (without saving).
:e!	Exit immediately, and don't ask about saving.
:d	Delete the current line.
:h	Go to online help.
:j	Delete the line below the cursor.
:n	If you specified multiple files on the command line, go on to edit the next file.
:u	Undo the last operation.
:x	Quick exit. Saves the file without being asked.
:*n*	Go to line number *n*.
:-*n*	Go up *n* lines.
:+*n*	Go down *n* lines.