

Vi Commands Cheat Sheets

Before doing anything to a document, type the following command followed by a carriage return:

```
:set showmode
```

Good practice note:

Especially for beginners, when using Vi, hit [esc] twice before every new command. This makes sure you aren't in the wrong mode. ESC exits all modes except the initial Command Mode.

When you open Vi, you can't just start typing a thesis because Vi has different modes such as APPEND MODE, INSERT MODE, REPLACE MODE, and COMMAND MODE. So to do certain actions, you must first enter the appropriate mode. See "MODES" for more in-depth information.

The final two pages of this document is a list of Vi core commands. These are the bare essentials of Vi compressed onto two pages.

Table of Contents

STARTING VI (VI IS CASE SENSITIVE!!! SO MAKE SURE CAPS LOCK IS OFF.)	2
ENDING VI (SAVING, EXITING, ETC.)	2
STATUS (LINE NUMBERS, ETC.)	3
MOVING (THESE WILL TELL YOU HOW TO GET THE CURSOR WHERE YOU WANT IT-FAST.)	3
MOVING THE CURSOR.....	3
MOVING THE SCREEN.....	4
MODES (ADDING AND/OR REPLACING TEXT).....	5
SEARCHING (FIND AND/OR REPLACE WHAT YOU NEED)	6
EDITING (DELETING, COPYING, PLACING TEXT, JOINING LINES, CHANGING TEXT, REPEAT COMMAND, UNDO COMMAND)	7
SHELL WORKS (COMMANDS THAT EXECUTE OUTSIDE OF THE VI PROGRAM.)	9
MACROS, SUBS, ABBREVIATIONS (MACRO FUNCTIONS, SUBSTITUTION STRINGS, AND ABBREVIATIONS)	10
Setting Options	10
VI CORE COMMANDS (THE MINIMUM FUNCTIONS)	12
TECHREPUBLIC	14
Additional resources	14
Version history.....	14
Tell us what you think.....	14

STARTING Vi (Vi is CaSe SENSITIVE!!! So make sure Caps Lock is OFF.)

<u>Command</u>	<u>Result</u>
<i>Vi filename</i>	Edits <i>filename</i> .
<i>Vi -r filename</i>	Edits last saved version of <i>filename</i> after a crash .
<i>Vi + n filename</i>	Edits <i>filename</i> and places the cursor at line n .
<i>Vi + filename</i>	Edits <i>filename</i> and places cursor on the last line .
<i>Vi filename file2 ...</i>	Edits <i>filename</i> and then edits file2 and so on. After saving <i>file1</i> enter :n for the next file.
<i>Vi +/string file</i>	Edits <i>file</i> and places cursor at the first line containing string .

ENDING Vi (Saving, exiting, etc.)

<u>Command</u>	<u>Result</u>
<i>ZZ</i> or <i>:wq</i> or <i>:x</i>	Saves and exits Vi .
<i>:w</i>	Saves current file but does not exit .
<i>:w file</i>	Saves current as file but does not exit .
<i>:w! file</i>	Saves file overriding normal checking .
<i>:n,mw file</i>	Saves lines n through m to a file named file .
<i>:n,mw>>file</i>	Appends lines n through m to the end of a file named file .
<i>:q</i>	Quits Vi, saving changes before leaving. (You may be prompted to save.)
<i>:q!</i>	Quits Vi without saving changes .
<i>Q</i>	Escapes Vi into ex editor with the same file; :Vi returns.
<i>:e!</i>	Re-edits current file disregarding changes since last save.
<i>:we!</i>	Re-edits current file saving changes .

STATUS (Line numbers, etc.)

<u>Command</u>	<u>Result</u>
:=	Shows current line number .
:=	Shows number of lines in file.
CTRL-g	Shows filename , current line number , total lines in file, and % of file location .
: (letter "l")	Displays tab (^I) backslash (\) backspace (^H) newline (\$) bell (^G) formfeed (^L) of current line.

MOVING (These will tell you how to get the cursor where you want it—fast.)

MOVING THE CURSOR

<u>Command</u>	<u>Result</u>
arrow keys	These <i>do</i> work, but they may be too slow on big files.
h j k l	Left, down, up, and right , respectively.
CTRL-d	Moves forward ½ screenful .
CTRL-f	Moves forward one entire screenful .
CTRL-u	Moves backward ½ screenful .
CTRL-b	Moves backward one entire screenful .
nG	Moves to line n in the file .
G	Moves to the end of file .
H	Moves to the top of the screen.
nH	Moves to line n from the top of the screen.
M	Moves to the middle of the screen .
L	Moves to the bottom of the screen .
nL	Moves to line n from the bottom of the screen.
w or W	Moves to the start of the next word in the line; W ignores punctuation.
b or B	Moves to the start of the previous word in the line; B ignores punctuation.

<u>Command</u>	<u>Result</u>
e or E	Moves to the end of the next word in the line; E ignores punctuation.
0 (zero) or	Moves to the first column in the current line.
n	Moves to the column n in the current line.
^	Moves to the first non-blank character in the current line.
\$	Moves to the last character in the current line.
+ or <CR>	Moves to the first character in the next line .
-	Moves to the first non-blank character in the previous line .
(Moves back to the beginning of sentence .
)	Moves forward to the beginning of the next sentence .
{	Moves back to the beginning of the paragraph .
}	Moves forward to the beginning of the next paragraph .

MOVING THE SCREEN

CTRL-e	Moves screen up one line .
CTRL-y	Moves screen down one line .
CTRL-u	Moves screen up ½ page .
CTRL-d	Moves screen down ½ page .
CTRL-b	Moves screen up one page .
CTRL-f	Moves screen down one page .
CTRL-l	Redraws screen .
z <CR>	z-carriage return makes the current line the top line on the page.
nz <CR>	Makes the line n the top line on the page.
z.	Makes the current line the middle line on the page.
nz.	Makes the line n the middle line on the page.
z-	Makes the current line the bottom line on the page.
nz-	Makes the line n the bottom line on the page.

MODES (Adding and/or replacing text)

<u>Command</u>	<u>Result</u>
<i>itext</i> ESC	insert mode —You can start typing and it will insert <i>text</i> <u>before</u> the letter your cursor currently highlights until ESC is used.
<i>ltext</i> ESC	insert mode —Same as (i) except it will insert <i>text</i> before the first nonblank character on the line until ESC is used.
<i>atext</i> ESC	append mode —Allows you to insert <i>text</i> <u>after</u> the letter your cursor currently highlights until ESC is used.
<i>Atext</i> ESC	append mode —Same as (a) except it will insert <i>text</i> at the end of the line until ESC is used.
<i>rchar</i> ESC	replace mode —Replaces the currently highlighted character with <i>char</i> until ESC is used.
<i>Rtext</i> ESC	replace mode —Same as (r) except it will overwrite until ESC is used.
<i>o</i>	Opens new line <i>below</i> the current line and insert until ESC is used.
<i>O</i> (letter “O”)	Opens new line <i>above</i> the current line and insert until ESC is used.
CTRL- <i>v char</i>	While inserting , ignores special meaning of <i>char</i> (e.g., for inserting characters like ESC and CTRL) until ESC is used.
<i>:r file</i>	Reads <i>file</i> and inserts it after current line.
<i>:nr file</i>	Reads <i>file</i> and inserts it after line number <i>n</i>.

SEARCHING (Find and/or replace what you need)

Characters	What they match...
^ (caret)	Matches beginning of line.
\$	Matches end of line.
.	Matches any single character.
\<	Matches beginning of word.
\>	Matches end of word.
[<i>str</i>]	Matches any single character in <i>str</i> .
[^ <i>str</i>]	Matches any character <u>not</u> in <i>str</i> .
[<i>a-n</i>]	Matches any character between <i>a</i> and <i>n</i> .
*	Matches zero or more occurrences of previous character in expression.
\	Escapes the meaning of the next character (e.g., \\$ allows you to search for \$).
\\	Escapes the \ character.
%	Searches to beginning of balancing () [] or { }.
f <i>char</i>	Searches forward in current line to <i>char</i> .
F <i>char</i>	Searches backward in current line to <i>char</i> .
t <i>char</i>	Searches forward in current line to character before <i>char</i> .
T <i>char</i>	Searches backward in current line to character before <i>char</i> .
/ <i>str</i> <CR>	Finds forward to <i>str</i> .
? <i>str</i> <CR>	Finds in reverse for <i>str</i> .
:set ic	Ignores case when searching .
:set noic	Pays attention to case when searching .
: <i>n</i> , <i>ms</i> / <i>str1</i> / <i>str2</i> / <i>opt</i>	Searches from <i>n</i> to <i>m</i> for <i>str1</i> ; replaces <i>str1</i> to <i>str2</i> ; using <i>opt-opt</i> can be g for global change, c to confirm change (y to acknowledge, <CR> to suppress), and p to print changed lines.
&	Repeats last :s command.
:g/ <i>str</i> / <i>cmd</i>	Runs cmd on all lines that contain str .
:g/ <i>str1</i> / <i>str2</i> / <i>str3</i>	Finds the line containing <i>str1</i> , replaces <i>str2</i> with <i>str3</i> .
:v/ <i>str</i> / <i>cmd</i>	Executes <i>cmd</i> on all lines that do not match <i>str</i> .

EDITING (Deleting, copying, placing text, joining lines, changing text, repeat command, undo command)

Command	Result
CTRL-h <u>or</u> Backspace	While inserting, deletes previous character.
CTRL-w	While inserting, deletes previous word
CTRL-x	While inserting, deletes to start of inserted text.
nX	Deletes n characters starting with current; omitting n deletes current character only.
nX	Deletes previous n characters; omitting n deletes previous character only.
xp	Switches character at cursor with following character.
ndw	Deletes the next n words starting with current; omitting n deletes the current word only.
ndb	Deletes the previous n words starting with current; omitting n deletes the previous word only.
ndd	Deletes n lines beginning with the current line; omitting n deletes the current line only.
:n,m	Deletes lines n through m.
D or d\$	Deletes from the cursor to the end of the line.
dcursor_cmd	Deletes everything included in the cursor command (e.g., dG would delete from current position to the end of the file, and d4 would delete to the end of the fourth sentence).
nyy or nY	Places n lines in the buffer-copies; omitting n copies only the current line to the buffer.
ycursor_cmd	Copies from cursor to cursor_cmd (e.g., yG copies current line to the last line in the file to the buffer).
“(a-z)nyy <u>or</u> “(a-z)ndd	Copies or cuts (deletes) n lines into a named buffer a through z; omitting n works on current line.
p	Pastes copied text after cursor; also prints last deleted text.
P	Pastes copied text before the cursor; also prints last deleted text.
“(a-z)p or “(a-z)P	Pastes text from a named buffer a through z after or before the current line.
nJ	Joins the next n lines together; omitting n joins the beginning of the next line to the end of the current line.
s <i>text</i> ESC	Substitutes <i>text</i> for the current character until ESC is used.

<u>Command</u>	<u>Result</u>
S or cc <i>text</i> ESC	Substitutes <i>text</i> for the entire line until ESC is used.
cw <i>text</i> ESC	Changes current word to <i>text</i> until ESC is used.
C <i>text</i> ESC	Changes rest of the current line to <i>text</i> until ESC is used.
o <i>cursor_cmd</i> <i>text</i> ESC	Changes to <i>text</i> from current position to <i>cursor_cmd</i> until ESC is used.
u	Undoes last command.
U	Restores current line to its original state.
"np	Retrieves the last nth delete (last 9 deletes are kept in a buffer).
"1pu.u.	Scrolls through the delete buffer until the desired delete is retrieved (repeat u.).
n	Repeats last / or ? search command.
N	Repeats, in reverse order, last / or ? search command.
; (semi-colon)	Repeats last f F t or T search command.
, (comma)	Repeats, in reverse direction, last / or ? search command.
. (period)	Repeats last text change command.
CTRL-i or TAB	While inserting, inserts one shift width.
n<< or n>>	Shifts n lines left or right (respectively) by one shift width ; omitting n shifts one line.
< or >	Use with cursor command to shift multiple lines left or right.

SHELL WORKS (Commands that execute outside of the Vi program.)

<u>Command</u>	<u>Result</u>
:! cmd	Executes shell command cmd ; you can add these special characters to indicate: % name of current file # name of last file edited
!! cmd	Executes shell command cmd , places output in file starting at current line.
:!!	Executes last shell command.
:r! cmd	Reads and inserts output from cmd .
:f <i>file</i>	Renames current file to file .
:w !cmd	Sends currently edited file to cmd as standard input and execute cmd .
:cd dir	Changes current working directory to dir .
:sh	Starts a sub-shell (CTRL-d returns to editor).
:so <i>file</i>	Reads and executes commands in <i>file</i> (<i>file</i> is a shell script).
!cursor_cmd cmd	Sends text from current position to cursor_cmd to shell command cmd .
!)sort <CR>	Sorts from current position to end of paragraph and replaces text with sorted text.

MACROS, SUBS, ABBREVIATIONS (Macro functions, substitution strings, and abbreviations)

<u>Command</u>	<u>Result</u>
<code>:[address]s/search</code> <code>-string/replace-</code> <code>string[/g]</code>	This is the format of a substitute command—description below.
address	One line number or two line numbers separated by a comma. A. represents the current line, \$ represents the last line, and % represents the entire file.
<i>search-string</i>	A regular expression that can be a simple string of characters.
<i>replace-string</i>	The replacement string.
g	Indicates a global replacement (more than one replacement per line).

Setting Options

<u>Command</u>	<u>Result</u>	<u>Command</u>	<u>Result</u>
NOTE: Map allows you to define strings of Vi commands. If you create a file called ".exrc" in your home directory, any map or set command you place inside this file will be executed every time you run Vi. To imbed control characters like ESC in the macro, you need to precede them with CTRL-v. If you need to include quotes ("), precede them with a \ (backslash). Unused keys in Vi are: K V g q v * = and the function keys.			
EX: <code>:map v /I CTRL-v ESC dwiYou CTRL-v ESC ESC</code> when v is pressed, search for "I" (/I ESC), delete word (dw), and insert "You" (iYou ESC). CTRL-v allows ESC to be inserted.			
<code>:map key</code> cmd_seq	Defines <i>key</i> to run cmd_seq when pressed.	<code>:map</code>	Displays all created macros on status line.
<code>:unmap key</code>	Removes macro definition for key.	<code>:ab str string</code>	When <i>str</i> is input, replaces it with string .
<code>:ab</code>	Displays all abbreviations.	<code>:una str</code>	Unabbreviates <i>str</i> .
Options given are default. To change them, enter type <code>:set option</code> to turn them on or <code>:set nooption</code> to turn them off.			
To make them execute every time you open Vi, create a file in your HOME directory called .exrc and type the options without the colon (:) preceding the option.			
<code>:set all</code>	Prints all options to the screen.	<code>:set nooption</code>	Turns off <i>option</i> .
<code>:set ai</code>	Turns on auto indentation.	<code>:set ap</code>	Prints line after <i>d c J m :s t u</i> commands.
<code>:set bf</code>	Discards control characters from input.	<code>:set eb</code>	Precedes error messages with a bell.
<code>:set ic</code>	Ignores case when searching.	<code>:set dir=tmp</code>	Sets directory or buffer file.
<code>:set lisp</code>	Modifies brackets for Lisp compatibility.	<code>:set magic</code>	Allows pattern matching with special characters.

Command	Result	Command	Result
<code>:set mesg</code>	Allows others to send messages.	<code>:set list</code>	Shows tabs (^I) and end of line (\$).
<code>:set nu</code>	Shows line numbers.	<code>:set opt</code>	Speeds output; eliminates automatic RETURN.
<code>:set prompt</code>	Prompts for command input with <code>..</code>	<code>:set re</code>	Simulates smart terminal on dumb terminal.
<code>:set report</code>	Indicates largest size of changes reported on status line.	<code>:set ro</code>	Changes file type to "read only."
<code>:set scroll=<i>n</i></code>	set <i>n</i> lines for CTRL-d and z	<code>:set</code>	set shell escape (default is /bin/sh)
		<code>sh=<i>shell_path</i></code>	
<code>:set showmode</code>	Indicates input or replace mode at bottom.	<code>:set sw=<i>n</i></code>	Sets shift width to <i>n</i> characters.
<code>:set term</code>	Prints terminal type.	<code>:set terse</code>	Shorten messages with terse.
<code>:set timeout</code>	Eliminates one-second time limit for macros.	<code>:set tl=<i>n</i></code>	Sets significance of tags beyond <i>n</i> characters (0 means all).
<code>:set ts=<i>n</i></code>	Sets tab stops to <i>n</i> for text input.	<code>:set wa</code>	Inhibits normal checks before write commands.
<code>:set warn</code>	Warns "no write since last change."	<code>:set window=<i>n</i></code>	Sets number of lines in a text window to <i>n</i> .
<code>:set wm=<i>n</i></code>	Sets automatic wraparound <i>n</i> spaces from right margin.		

Vi Core Commands (The minimum functions)

<u>Command</u>	<u>Result</u>
STARTING Vi	(Vi is CaSe SEnSItIve!!! So make sure Caps Lock is OFF .)
<code>Vi file</code>	<code>edit file</code>
ENDING Vi	(Saving, exiting, etc.)
<code>ZZ</code> or <code>:wq</code> or <code>:x</code>	Saves and exits Vi.
<code>:q</code>	Quits Vi, saving changes before leaving. (You may be prompted to save.)
<code>:q!</code>	Quits Vi without saving changes.
MOVING	(These will tell you how to get the cursor where you want it—fast.)
<u>MOVING THE CURSOR</u>	
arrow keys	These <i>do</i> work, but they may be too slow on big files.
<code>h j k l</code>	Left, down, up, and right , respectively.
<code>CTRL-d</code>	Moves forward ½ screenful .
<code>CTRL-f</code>	Moves forward 1 whole screenful .
<code>CTRL-u</code>	Moves backward ½ screenful .
<code>CTRL-b</code>	Moves backward 1 entire screenful .
<code>nG</code>	Moves to line n in the file .
<code>G</code>	Moves to the end of file .
<code>H</code>	Moves to the top of the screen.
<code>M</code>	Moves to the middle of the screen .
<code>L</code>	Moves to the bottom of the screen .
<code>w</code> or <code>W</code>	Moves to the start of the next word in the line; <code>W</code> ignores punctuation.
<code>b</code> or <code>B</code>	Moves to the start of the previous word in the line; <code>B</code> ignores punctuation.
<code>e</code> or <code>E</code>	Moves to the end of the next word in the line; <code>E</code> ignores punctuation.
<code>^</code>	Moves to the first non-blank character in the current line.
<code>\$</code>	Moves to the last character in the current line.
<code>+</code> or <code><CR></code>	Moves to the first character in the next line .
MODES	(Adding and/or replacing text)
<code>itext ESC</code>	insert mode —You can start typing, and it will insert <i>text</i> <u>before</u> the letter your cursor currently highlights until <code>ESC</code> is used.
<code>atext ESC</code>	append mode —This allows you to insert <i>text</i> <u>after</u> the letter your cursor currently highlights until <code>ESC</code> is used.
<code>Atext ESC</code>	append mode —Same as (a) except it will insert <i>text</i> at the end of the line until <code>ESC</code> is used.
<code>rchar ESC</code>	replace mode —Replaces the currently one highlighted character with <i>char</i> until <code>ESC</code> is used.
<code>Rtext ESC</code>	replace mode —Same as (r) except it will overwrite until <code>ESC</code> is used.
<code>o</code>	Opens new line below the current line and insert until <code>ESC</code> is used.
<code>O</code> (letter "O")	Opens new line above the current line and insert until <code>ESC</code> is used.

SEARCHING (Find and/or replace what you need)

<u>Command</u>	<u>Result</u>
<i>fchar</i> <i>Fchar</i>	Searches forward in current line to <i>char</i> . Searches backward in current line to <i>char</i> .
<i>/str <CR></i> <i>?str <CR></i>	Finds forward to <i>str</i> . Finds in reverse for <i>str</i> .
EDITING (Deleting, copying, placing text, joining lines, changing text, repeat command, undo command)	
CTRL-h or Backspace	While inserting , deletes previous character .
CTRL-w	While inserting , deletes previous word
<i>nx</i>	Deletes n characters starting with current ; omitting <i>n</i> deletes current character only.
<i>ndw</i>	Deletes the next n words starting with current ; omitting <i>n</i> deletes the current word only.
<i>ndd</i>	Deletes n lines beginning with the current line ; omitting <i>n</i> deletes the current line only.
D or d\$	Deletes from the cursor to the end of the line.
<i>nyy</i> or <i>nY</i>	Places n lines in the buffer—copies ; omitting <i>n</i> copies only the current line to the buffer.
<i>p</i>	Pastes copied text after cursor ; also prints last deleted text.
<i>nJ</i>	Joins the next n lines together ; omitting <i>n</i> joins the beginning of the next line to the end of the current line.
<i>u</i>	Undoes last command.
<i>U</i>	Restores current line to its original state.
<i>n</i> <i>N</i>	Repeats last / or ? search command . Repeats, in reverse order, last / or ? search command .
<i>.</i> (period)	Repeats last text change command .
CTRL-i or TAB	While inserting, inserts one shift width .

OPTIONS

<u>Command</u>	<u>Result</u>	<u>Command</u>	<u>Result</u>
Options given are default. To change them, enter type <code>:set option</code> to turn them on or <code>:set nooption</code> to turn them off. To make them execute every time you open Vi, create a file in your home directory called <code>.exrc</code> and type the options without the colon (:) preceding the option.			
<code>:set all</code>	Prints all options to the screen.	<code>:set nooption</code>	Turns off <i>option</i> .
<code>:set ic</code>	Ignores case when searching.	<code>:set magic</code>	Allows pattern matching with special characters.
<code>:set nu</code>	Shows line numbers.	<code>:set ro</code>	Changes file type to "read only."
<code>:set scroll=n</code>	Sets <i>n</i> lines for CTRL-d and z.	<code>:set sw=n</code>	Sets shift width to <i>n</i> characters.
<code>:set showmode</code>	Indicates input or replace mode at bottom.	<code>:set window=n</code>	Sets number of lines in a text window to <i>n</i> .
<code>:set ts=n</code>	Sets tab stops to <i>n</i> for text input.	<code>:set wm=n</code>	Sets automatic wraparound <i>n</i> spaces from right margin.

TechRepublic

Additional resources

- [Use vi to edit files in a UNIX environment](#) (Article)
- [15 tips for a more pleasant Vi experience](#) (Download)
- [Vi for programmers, part 1](#) (Article)
- [Vi for programmers, part 2](#) (Article)

Version history

Version: 2.0

Reformatted for easier reading and printing

Published: March 22, 2005

Tell us what you think

TechRepublic downloads are designed to help you get your job done as painlessly and effectively as possible. Because we're continually looking for ways to improve the usefulness of these tools, we need your feedback. Please take a minute to [drop us a line](#) and tell us how well this download worked for you and offer your suggestions for improvement.

Thanks!

—The TechRepublic Downloads Team