

Beginners guide to VI editor (command line reference)

The vi editor is a command-line, interactive editor that you can use to create and modify text files. The vi editor is also the only text editor that you can use to edit certain system files without changing the permissions of the files. The Vim editor is an enhanced version of the vi editor.

Accessing the VI Editor

To create, edit, and view files in the vi editor, use the vi command. The vi command includes the following three syntaxes:

```
$ vi
$ vi filename
$ vi options filename
```

If the system crashes while you are editing a file, you can use the **-r** option to recover the file.

```
$ vi -r filename
```

The file opens so that you can edit it. You can then save the file and exit the vi editor, by using the following command:

```
$ vi -R filename
```

The file opens in read-only mode to prevent accidental overwriting of the contents of the file.

The VI Editor Modes

The vi editor provides three modes of operation:

1. Command mode – The command mode is the default mode for the vi editor. In this mode, you can run commands to delete, change, copy, and move text. You can also position the cursor, search for text strings, and exit the vi editor.

2. Input mode – You can insert text into a file in the input mode. The vi editor interprets everything you type in the input mode as text. To invoke input mode, press one of the following lowercase keys:

- **i** – Inserts text before the cursor
- **o** – Opens a new blank line below the cursor
- **a** – Appends text after the cursor

You can also invoke the input mode to insert text into a file by pressing one of the following uppercase keys:

- **I** – Inserts text at the beginning of the line
- **O** – Opens a new blank line above the cursor
- **A** – Appends text at the end of the line

3. Last line mode – You can use advanced editing commands in the last line mode. To access the last line mode, enter a colon (:) while in the command mode. Entering the colon (:) character places the cursor at the bottom line of the screen.

Note: The last line mode is actually the ex mode. The vi editor is essentially a visual extension to the ex editor, which in turn is an extended version of the ed editor.

Switching Between Modes

- The default mode for the vi editor is the command mode.
- To switch to the input mode, press i, o, or a.
- To return to the command mode, press the Escape key.
- In the command mode, you can save the file and quit the vi editor, and return to the shell prompt.

Viewing Files in the Read-Only Mode

The view command enables you to view files in the read- only mode.

```
$ view filename
```

The view command invokes the vi editor in the read-only option, which means you cannot save changes to the file. For example, to view the data.txt file in the read-only mode, enter the following command:

```
$ view data.txt
```

The data.txt file appears. Enter the: q command to exit the file, exit the vi editor, and return to the shell prompt.

Moving the Cursor within the VI Editor

The table shows the key sequences that move the cursor.

Key Sequence	Cursor Movement
h, left arrow, or Backspace	Left one character
j or down arrow	Down one line
k or up arrow	Up one line
l, right arrow, or space bar	Right (forward) one character
w	Forward one word
b	Back one word
e	To the end of the current word
\$	To the end of the line
0 (zero)	To the beginning of the line
^	To the first non-white space character on the line
Return	Down to the beginning of the next line
G	Goes to the last line of the file

1G	Goes to the first line of the file
:n	Goes to Line n
nG	Goes to Line n
Control + F	Pages forward one screen
Control + D	Scrolls down one-half screen
Control + B	Pages back one screen
Control + U	Scrolls up one-half screen
Control + L	Refreshes the screen
Control + G	Displays current buffer information

Inserting and Appending Text

The table describes the commands to insert and append text to a new or existing file by using the vi editor.

Command	Function
a	Appends text after the cursor
A	Appends text at the end of the line
i	Inserts text before the cursor
I	Inserts text at the beginning of the line
o	Opens a new line below the cursor
O	Opens a new line above the cursor
:r filename	Inserts text from another file into the current file

Note: The vi editor is case-sensitive. Use the appropriate case for the input commands. Also, most of the input commands and cursor movements can be preceded by a number to repeat the command that many times.

Text-Deletion Commands

The table shows commands that delete text in the vi editor.

Command	Function
R	Overwrites or replaces characters on the line at and to the right of the cursor. To terminate this operation, press Escape.
C	Changes or overwrites characters from the cursor to the end of the line
s	Substitutes a string for a character at the cursor
x	Deletes a character at the cursor
dw	Deletes a word or part of the word to the right of the cursor
dd	Deletes the line containing the cursor
D	Deletes the line from the cursor to the right end of the line

:n,nd	Deletes lines n -n (For example, :5,10d deletes lines 5-10.)
-------	--

Note: Output from the delete command writes to a buffer from which text can be retrieved.

You can use numerous commands to edit files by using the vi editor. The following sections describe basic operations for deleting, changing, replacing, copying, and pasting. Remember that the vi editor is case-sensitive.

Edit Commands

The table describes the commands to change text, undo a change, and repeat an edit function in the vi editor.

Command	Function
cw	Changes or overwrites characters at the cursor location to the end of that word
r	Replaces the character at the cursor with one other character
J	Joins the current line and the line below
xp	Transposes the character at the cursor and the character to the right of the cursor
~	Changes letter casing to uppercase or lowercase, at the cursor
u	Undoes the previous command
U	Undoes all changes to the current line
.	Repeats the previous command

Note: Many of these commands change the vi editor into the input mode. To return to the command mode, press the Esc key.

Search and Replace Commands

The table shows the commands that search for and replace text in the vi editor.

Command	Function
/string	Searches forward for the string
?string	Searches backward for the string
n	Searches for the next occurrence of the string. Use this command after searching for a string.
N	Searches for the previous occurrence of the string. Use this command after searching for a string.
:%s/old/new/g	Searches for the old string and replaces it with the new string globally

Copy and Paste Commands

The table shows the commands that cut, copy, and paste text in the vi editor.

Command	Function
yy	Yanks a copy of a line
p	Puts yanked or deleted text under the line containing the cursor
P	Puts yanked or deleted text before the line containing the cursor
:n,n co n	Copies lines n -n and puts them after line n (For example, :1,3 co 5 copies lines 1-3 and puts them after line 5.)
:n,n m n	Moves lines n -n to line n. For example, :4,6 m 8 moves lines 4-6 to line 8, line 6 becomes line 8, line 5 becomes line 7, and line 4 becomes line 6.

Save and Quit Commands

The table describes the commands that save the text file, quit the vi editor, and return to the shell prompt.

Command	Function
:w	Saves the file with changes by writing to the disk
:w new_filename	Writes the contents of the buffer to new_filename
:wq	Saves the file with changes and quits the vi editor
:x	Saves the file with changes and quits the vi editor
ZZ	Saves the file with changes and quits the vi editor
:q!	Quits without saving changes
ZQ	Quits without saving changes

Session Customization

You can customize a vi session by setting variables for the session. When you set a variable, you enable a feature that is not activated by default. You can use the set command to enable and disable variables. The set command variables include displaying line numbers and invisible characters, such as the Tab and the end-of-line characters.

To create an automatic customization for all your vi sessions, perform the following steps:

1. Create a file named **.exrc** in your home directory.
2. Enter any of the set variables into the .exrc file.
3. Enter each set variable without the preceding colon.

4. Enter each command on one line.

The vi editor reads the .exrc file located in your home directory each time you open a vi session, regardless of your current working directory.

Note: The same steps apply for customizing a session in the Vim editor. Except that, instead of creating an .exrc file, you need to create a **.vimrc** file.

Session Customization Commands

The table describes some of the variables of the **set** command.

Command	Function
:set nu	Shows line numbers
:set nonu	Hides line numbers
:set ic	Instructs searches to ignore case
:set noic	Instructs searches to be case-sensitive
:set noic	Displays invisible characters, such as ^I for a Tab and \$ for end-of-line characters
:set nolist	Turns off the display of invisible characters
:set showmode	Displays the current mode of operation
:set noshowmode	Turns off the mode of operation display
:set	Displays all the vi variables that are set
:set all	Displays all vi variables and their current values