

Advantages of using set -o vi

Asked 10 years, 11 months ago Modified 4 years, 1 month ago Viewed 46k times



70



I have seen many developers using this command to set the option to vi. I never understood the real use of this?

When using bash commands, what help does switching to vi provide?

bash

vim

vi

Share Improve this question Follow

edited May 14, 2014 at 14:10

asked Jan 31, 2012 at 6:43



Braiam

34.8k

25

107

165



Chander Shivdasani

1,461

3

13

11

8 Answers

Sorted by:

Highest score (default)



70



By setting your readline editing to either emacs (the default) or vi (`set -o vi`) you are essentially standardizing your editing commands, across the shell and your editor of choice¹.

Thus, if you want to edit a command in the shell you use the same commands² that you would if you were in your text editor. This means only having to remember one command syntax and (if that were not advantage enough) would probably make your editing in both environments faster and less error prone...

You can further leverage this relationship in vi-mode by pulling up any command from your shell history, hitting `Escape` to enter command mode and then hitting `v`, which will open your \$EDITOR with the command loaded for more complex editing with the full power of vim. Once you have finished editing the command to your satisfaction, `:wq` and the command is executed back in your shell.

1. Assuming, of course, that you use Emacs or Vi/m as your editor.

2. Or, more accurately, a subset thereof...

Share Improve this answer Follow

edited Oct 30, 2013 at 3:12

answered Jan 31, 2012 at 7:02



jasonwryan

69.9k


32

190

224

4 The Esc-v trick is one of the best features of `set -o vi` that most people don't know. This is especially true if you want to issue the same command multiple times with different arguments (as multiple command line commands).

– Michael Goldshteyn May 1, 2018 at 19:31

- 4 The Esc-v trick is not really an advantage specific to `set -o vi`. Under the default setting of `set -o EMACS`, C-x,C-e will bring up an emacs window (or the `$EDITOR`, if set) in which you can edit the command before running it.
– [John Gowers](#) Jul 31, 2018 at 17:43
- 2 with Esc-v, you have to be careful since doing `:q!` will also execute the command. You wouldn't want to have `rm -rf precious_directory` on the command line and accidentally hit Esc-v and then reflexively hit `:q!` thinking you're safe. I set up my vim to immediately add a `#` at the start of the line when I do Esc-v to protect myself from this possibility. – [mattb](#) May 31, 2021 at 18:26 

Vi mode is a huge usability improvement if you are using a mobile SSH client like ConnectBot for Android.

12 This is due to a reduced reliance on modifier keys.

Vim is much easier to use with a virtual keyboard on a smartphone or tablet than ... anything else, including the native editing methods built into the Android UI. Ironically, it is easier to edit C sources with Vim in an SSH session than to edit, say, an instant message with the platform's own editing widget for that purpose.

Shell vi mode brings a similar benefit.

Share Improve this answer Follow

edited Dec 18, 2018 at 16:39

answered Nov 28, 2013 at 0:15



[Kaz](#)

6,820

1

22

42

I've been hoping there was vi mode on my phone keyboard. You mention this is as if it already existed. How do I get vi mode on my phone? – [meh](#) Apr 25, 2019 at 22:10

It lets you edit stuff at the command line using the vi modes and operations.

7 An example will help make it much clearer:

You type `cp tmp/some_other_long_directory/file1.txt /tmp2/some_other_xtra_long_dir/`

but you get an error - you should have typed `file1.txt` not `file1.xt`

Without this option set, you press up-arrow and then press left arrow and let it repeat for... 35 times, until you get to the `.xt` and then you type the extra `t`. Total keystrokes: **37**.

With this option set you can (for example) press arrow up once, then Escape for command mode, 0 to go to the start of the line and then `/xt[return]` to get to the `xt` and then you can type `i` for insert mode and type the missing `t`. This may seem insanely complicated in some respects but if you are a vim user these commands are already very well known. Total keystrokes: **9**

Share Improve this answer Follow

edited Mar 31, 2014 at 0:57

answered Nov 28, 2013 at 2:44




[Michael Durrant](#)

39.9k

69

158

230

- 1 You could use a few `Alt+B` s or a `Ctrl+Alt+]` in Emacs mode too, so this doesn't seem like a strong example to me. – [Mikel](#) Apr 1, 2014 at 0:11 

2 Seeing as we're golfing: Emacs mode: Up , Ctrl+Alt+J , . , Right , t , Enter = 8. Vi mode: Up , F , . , a , t , Enter = 7. :) – [Mikel](#) Apr 1, 2014 at 0:44

@Mikel you forgot that vi mode starts in Insert mode so you have Esc in there too, it's a dead heat. – [dragon788](#) Oct 15, 2017 at 5:44

I'm not sure if there is a direct advantage. I've been a vi user for more than 20 years. I'm also a screen user for even longer, and of other programs that use vi keys. It's natural for me to prefer to set "vi" mode in bash. But I also work on hundreds of servers in my job, most are set to the default "emacs" mode. So I need to use both modes. But it is really just a matter of preference.

Share Improve this answer Follow

edited Jan 31, 2012 at 13:18

answered Jan 31, 2012 at 7:18



Kevin

39.1k

16

85

112



Arcege

22k

5

55

63

Similar situation for me - I use vim a lot but I've always found it less work to become proficient at the default emacs-like readline keys (which can be just as convenient as the vi subset) than it is to add a line switching to the vi mode on every new system I come across. – [jw013](#) Jan 31, 2012 at 20:52

1 I don't work on hundreds of servers, but I do use [sshrc](#) to keep my bash setup similar between different computers when ssh ing. Of course, that doesn't help when I'm using a machine locally that doesn't have my .bashrc stuff on it. – [Kyle Strand](#) Jan 5, 2016 at 20:48

The main advantage is modal editing of your command line. If you're familiar with Vim and likes its philosophy the benefits must be obvious. If you are experienced with it, your finger's muscle memory will make you edit your bash commands in lightning speeds.

NB: If you don't like modal editing, you should still learn to take advantage of the (default) emacs-mode. [Here are some nifty keyboard shortcuts](#) that will work on any process with readline , like bash .

Share Improve this answer Follow

edited Apr 13, 2017 at 12:36

answered Jan 31, 2012 at 8:12



Community Bot

1



rahmu

19.1k

27

83

127

If you get used to vi , so you set it to vi mode as your shell editor. That'd be the obvious reason. The other one is when the bash is not available by default in some Oses (mostly UNIX like AIX, Solaris), so the shell history feature is not available, so the way you get the past commands by setting the shell editor to vi , and

Esc , Ctrl + K OR Ctrl + L

Share Improve this answer Follow

edited Nov 28, 2013 at 8:37

answered Oct 30, 2013 at 3:20



Anthon

76.8k

42

159

217



Shâu Shắc

919

1

10

11

It can let you easily navigate and edit the command line using vim's shortcuts, e.g. quickly move to one word right, delete a word.

By the default shortcuts, when you need to go to the end of the line, you need to `Ctrl + e`, whereas with `set -o vi`, you just hit `$`, like in vim.



Share Improve this answer Follow

edited Nov 9, 2014 at 4:47

answered Nov 9, 2014 at 4:10



HalosGhost

4,644

10

32

40



Hahn

111

3

Probably late to the party, but for me using the vi mode is more about when creating interactive scripts.. for example

1

```
for i in `ls | grep -v gz`  
do  
    echo $i  
    gzip $i  
done
```



a very simple example of what could be quite complicated.. using `ESC` `k` `v` puts you into a vi session where you can modify the script, and then `:wq` and it runs.

Share Improve this answer Follow

edited Sep 9, 2015 at 23:22

answered Sep 5, 2015 at 4:04



don_crissti

76.4k

30

208

236



Walter Werner

11

1

1 This may be a good example of using bash in `vi` mode, but it's a bad example of bash code. (1) Using the output from `ls` as the input to any other sort of processing is a bad idea. (2) The `$(...)` syntax for command substitution is widely considered to be more readable than the ``...`` syntax. (3) You should always quote shell variables unless you have a good reason not to, and you're *sure* you know what you're doing. – G-Man Says 'Reinstate Monica' Sep 5, 2015 at 18:13