

# Vi

## What is vi?

Scattered all around your computer are thousands of text files. To a new user, this may seem inconsequential, but almost everything in Slackware Linux uses a plain-text file for configuration. This allows users to make changes to the system quickly, easily, and intuitively. In chapter 5, we looked at a few commands such as **cat** and **less** that can be used to read these files, but what if we want to make changes to them? For that, we need a text editor, and **vi** is up to the task.

In short, **vi** is one of the oldest and most powerful text editors still used today. It's beloved by system administrators, programmers, hobbyists, and others the world over. In fact, nearly this entire book was written using **vi**; only the next chapter on **emacs** was written with that editor.

A little further explanation is needed to learn exactly what **vi** is today though, as Slackware Linux technically doesn't include **vi**. Rather, Slackware includes two vi "clones", **elvis**(1) and **vim**(1). These clones add many additional features to vi such as syntax highlighting, binary editing modes, and network support. We won't go too deeply into all these details. By default, if you execute **vi** on Slackware Linux, you'll be using **elvis**, so all examples in this chapter will assume that is what you are using. If you've used another Linux distribution before, you may be more familiar with **vim**. If so, you might wish to change the symlink for /usr/bin/vi to point to /usr/bin/vim, or add an alias to your shell's startup scripts. **vim** is generally considered to be more feature-rich than **elvis**, but **elvis** is a much smaller program and contains more features than most users will ever need.

**vi** is very powerful, but also somewhat cumbersome and challenging for a new user to learn. However, mastering **vi** is an important skill for any self-respecting system administrator to learn, as **vi** is included on nearly every Linux distribution, every BSD system, and every UNIX system in existence. It's even included in Mac OS X. Once you've learned **vi**, you'll not have to learn another text editor to work on any of these systems. In fact, **vi** clones have even been ported to Microsoft Windows systems, so you can use it there too.

## The Different Modes of vi

New users are often frustrated when using **vi** for the first time. When invoked without any arguments, **vi** will display a screen something like this.

```
~
~
~
~
~
~
~
~
~
~
~
```

~  
Command

At this point, the user will be typing and expect the keys he presses to appear in the document. Instead, something really strange happens. The reason for this is simple. **vi** has different operation "modes". There is a command mode and an insert mode. Command mode is the default; in this mode, each keystroke performs a particular action such as moving the cursor around, deleting text, yanking (copying) text, searching, etc.

## Opening, Saving, and Quitting

Ok, so you've decided that you want to learn how to use **vi**. The first thing to do is learn how to open and save files. Opening files is actually pretty easy. Simply type the filename as an argument on the command-line and **vi** will happily load it for you. For example, `vi chapter_11.xml` will open the file `chapter_11.xml` and load its content onto the screen, simple enough. But what if we've finished with one document and wish to save it? We can do that in command mode using the `:w` command. When in command mode, pressing the `⓪` key temporarily positions the cursor on the very bottom line of the window and allows you to enter special commands. (This is technically known as ex-mode after the venerable **ex** application which we will not document here.) The command to save your current work is `:w`. Once this is done, **vi** will write your changes to the buffer back into the file. If you wish to open another document, simply use the `:e other_document` command and **vi** will happily open it for you. If you've made changes to the buffer but haven't saved it yet, `:e` will fail and print a warning message on the bottom line. You can bypass this with the `:e!` command. Most ex-mode commands in **vi** can be "forced" by adding `!` to them. This tells **vi** that you want to abandon any changes you've made to the buffer and open the other document immediately.

But what if I don't like my changes and want to quit or start over? That's easily done as well. Executing the `:e!` command without any arguments will re-open the current document from the beginning. Quitting **vi** is as simple as running the `:q` command if you haven't made any changes to the buffer, or `:q!` if you'd like to quit and abandon those changes.

## Moving Around

Moving around in **vi** is perhaps the hardest thing for a new user to learn. **vi** does not traditionally use the directional arrow keys for cursor movement, although in Slackware Linux that is an option. Rather, movement is simply another command issued in command-mode. The reason for this is rather simple. **vi** actually predates the inclusion of directional arrow keys on keyboards. Thus, movement of the cursor had to be accomplished by using the few keys available, so the right-hand "home row" keys of `h`, `j`, `k`, and `l` were chosen. These keys will move the cursor about whenever **vi** is in command mode. Here's a short table to help you remember how they work.

Command	Result
h	Move the cursor one character left
j	Move the cursor one line down
k	Move the cursor one line up
l	Move the cursor one character right

Moving around is a little more powerful than that though. Like many command keys, these movement keys accept numerical arguments. For example, **10j** will move the cursor down 10 lines. You can also move to the end or beginning of the current line with **\$** and **^**, respectively.

## Editing A Document

Now that we're able to open and save documents, as well as move around in them, it's time to learn how to edit them. The primary means of editing is to enter insert mode using either the **i** or **a** command keys. These either insert text at the cursor's current location, or append it after the cursor's current location. Once into insert mode, you can type any text normally and it will be placed into your document. You can return to command mode in order to save your changes by pressing the **ESC** key.

## Vi Cheat Sheet

Since **vi** can be difficult to learn, I've prepared a short cheat sheet that should help you with the basics until you begin to feel comfortable.

Command	Result
h	Move the cursor one character left
j	Move the cursor one line down
k	Move the cursor one line up
l	Move the cursor one character right
10j	Move the cursor ten lines down
G	Move to the end of the file
^	Move to the beginning of the line
\$	Move to the end of the line
dd	Remove a line (and store it in the copy buffer)
5dd	Remove 5 lines (and store them in the copy buffer)
dw	Remove a single word (and store it in the copy buffer)
5dw	Remove five words (and store them in the copy buffer)
yy	Yank (copy) a line (and store it in the copy buffer)
yw	Yank (copy) a single word (and store it in the copy buffer)
5yw	Yank five words (and store them in the copy buffer)
p	Paste the contents of the copy buffer at the cursor's location
P	Paste the contents of the copy buffer above the cursor's location
r	Replace a single character
R	Replace multiple characters
x	Delete a character
X	Delete the previous character
u	Undo the last action
:s'old'new'g	Replace all occurrences of 'old' with 'new' (current line only)
:%s'old'new'g	Replace all occurrences of 'old' with 'new' (all lines)
/asdf	Locate next occurrence of asdf
:q	Quit (without saving)

Command	Result
:w	Save the current document
:w file	Save the current document as 'file'
:x	Save and quit

# Chapter Navigation

Previous Chapter: [Working with Filesystems](#)

Next Chapter: [Emacs](#)

# Sources

- Original source: <http://www.slackbook.org/beta>
- Originally written by Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson

[slackbook](#), [vi](#), [text editor](#)

From:  
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:  
<https://docs.slackware.com/slackbook:vi>

Last update: **2012/09/16 15:11 (UTC)**

