8 Essential Vim Editor Navigation Fundamentals

his article is written by SathiyaMoorthy.  
http://www.thegeekstuff.com/2009/03/8-essential-vim-editor-navigation-fundamentals/  
This article is part of the ongoing [Vi / Vim Tips and Tricks](http://www.thegeekstuff.com/tag/vi-vim-tips-and-tricks/) series. Navigation is a vital part of text editing. To be very productive, you should be aware of all possible navigation shortcuts in your editor. In this article, let us review the following 8 Vi / Vim navigation options.

1. Line navigation
2. Screen navigation
3. Word navigation
4. Special navigation
5. Paragraph navigation
6. Search navigation
7. Code navigation
8. Navigation from command line

### 1. Vim Line Navigation

Following are the four navigation that can be done line by line.

* k – navigate upwards
* j – navigate downwards
* l – navigate right side
* h – navigate left side

By using the repeat factor in VIM we can do this operation for N times. For example, when you want to  
go down by 10 lines, then type “10j”.  
  
Within a line if you want to navigate to different position, you have 4 other options.

* 0 – go to the starting of the current line.
* ^ – go to the first non blank character of the line.
* $ – go to the end of the current line.
* g\_ – go to the last non blank character of the line.

### 2. Vim Screen Navigation

Following are the three navigation which can be done in relation to text shown in the screen.

* H – Go to the first line of current screen.
* M – Go to the middle line of current screen.
* L – Go to the last line of current screen.
* ctrl+f – Jump forward one full screen.
* ctrl+b – Jump backwards one full screen
* ctrl+d – Jump forward (down) a half screen
* ctrl+u – Jump back (up) one half screen

### 3. Vim Special Navigation

You may want to do some special navigation inside a file, which are:

* N% – Go to the Nth percentage line of the file.
* NG – Go to the Nth line of the file.
* G – Go to the end of the file.
* `” – Go to the position where you were in NORMAL MODE while last closing the file.
* `^ – Go to the position where you were in INSERT MODE while last closing the file.
* g – Go to the beginning of the file.

### 4. Vim Word Navigation

You may want to do several navigation in relation to the words, such as:

* e – go to the end of the current word.
* E – go to the end of the current WORD.
* b – go to the previous (before) word.
* B – go to the previous (before) WORD.
* w – go to the next word.
* W – go to the next WORD.

WORD – WORD consists of a sequence of non-blank characters, separated with white space.  
word – word consists of a sequence of letters, digits and underscores.  
  
Example to show the difference between WORD and word

* 192.168.1.1 – single WORD
* 192.168.1.1 – seven words.

### 5. Vim Paragraph Navigation

* **{** – Go to the beginning of the current paragraph. By pressing { again and again move to the previous paragraph beginnings.
* **}** – Go to the end of the current paragraph. By pressing } again and again move to the next paragraph end, and again.

### 6. Vim Search Navigation

* **/i** – Search for a pattern which will you take you to the next occurrence of it.
* **?i** – Search for a pattern which will you take you to the previous occurrence of it.
* **\***– Go to the next occurrence of the current word under the cursor.
* **#**– Go to the previous occurrence of the current word under the cursor.

### 7. Vim Code Navigation

**%** – Go to the matching braces, or parenthesis inside code.

### 8. Vim Navigation from Command Line

Vim +N filename: Go to the Nth line of the file after opening it.

vim +10 /etc/passwd

Vim +/pattern filename: Go to the particular pattern’s line inside the file, first occurrence from first. In the following example, it will open the README file and jump to the first occurrence of the word “install”.

vim +/install README

Vim +?patten filename: Go to the particular pattern’s line inside the file, first occurrence from last. In the following example, it will open the README file and jump to the last occurrence of the word “bug”.

vim +?bug README

### Recommended Reading

[](http://www.thegeekstuff.com/vim-101-hacks-ebook/)**[Vim 101 Hacks](http://www.thegeekstuff.com/vim-101-hacks-ebook/), by Ramesh Natarajan**. I’m a command-line junkie. So, naturally I’m a huge fan of Vi and Vim editors. Several years back, when I wrote lot of C code on Linux, I used to read all available Vim editor tips and tricks. Based on my Vim editor experience, I’ve written Vim 101 Hacks eBook that contains 101 practical examples on various advanced Vim features that will make you fast and productive in the Vim editor. Even if you’ve been using Vi and Vim Editors for several years and have not read this book, please do yourself a favor and read this book. You’ll be amazed with the capabilities of Vim editor.

> [Add your comment](http://www.thegeekstuff.com/2009/03/8-essential-vim-editor-navigation-fundamentals/" \l "commentform)

### If you enjoyed this article, you might also like..

|  |  |
| --- | --- |
| 1. [50 Linux Sysadmin Tutorials](http://www.thegeekstuff.com/2010/12/50-unix-linux-sysadmin-tutorials/) 2. [50 Most Frequently Used Linux Commands (With Examples)](http://www.thegeekstuff.com/2010/11/50-linux-commands/) 3. [Top 25 Best Linux Performance Monitoring and Debugging Tools](http://www.thegeekstuff.com/2011/12/linux-performance-monitoring-tools/) 4. [Mommy, I found it! – 15 Practical Linux Find Command Examples](http://www.thegeekstuff.com/2009/03/15-practical-linux-find-command-examples/) 5. [Linux 101 Hacks 2nd Edition eBook](http://www.thegeekstuff.com/linux-101-hacks-ebook/) Linux 101 Hacks Book | * [Awk Introduction – 7 Awk Print Examples](http://www.thegeekstuff.com/2010/01/awk-introduction-tutorial-7-awk-print-examples/) * [Advanced Sed Substitution Examples](http://www.thegeekstuff.com/2009/10/unix-sed-tutorial-advanced-sed-substitution-examples/) * [8 Essential Vim Editor Navigation Fundamentals](http://www.thegeekstuff.com/2009/03/8-essential-vim-editor-navigation-fundamentals/) * [25 Most Frequently Used Linux IPTables Rules Examples](http://www.thegeekstuff.com/2011/06/iptables-rules-examples/) * [Turbocharge PuTTY with 12 Powerful Add-Ons](http://www.thegeekstuff.com/2008/08/turbocharge-putty-with-12-powerful-add-ons-software-for-geeks-3/) |

|  |  |  |  |
| --- | --- | --- | --- |
| [Bash 101 Hacks Book](http://www.thegeekstuff.com/bash-101-hacks-ebook/) | [Sed and Awk 101 Hacks Book](http://www.thegeekstuff.com/sed-awk-101-hacks-ebook/) | [Nagios Core 3 Book](http://www.thegeekstuff.com/nagios-core-ebook/) | [Vim 101 Hacks Book](http://www.thegeekstuff.com/vim-101-hacks-ebook/) |

Tagged as: [Vi Basic Navigation](http://www.thegeekstuff.com/tag/vi-basic-navigation/), [vi command](http://www.thegeekstuff.com/tag/vi-command/), [vi editor](http://www.thegeekstuff.com/tag/vi-editor/), [Vi Navigation Commands](http://www.thegeekstuff.com/tag/vi-navigation-commands/), [vi tips](http://www.thegeekstuff.com/tag/vi-tips/), [Vi Vim Tips and Tricks](http://www.thegeekstuff.com/tag/vi-vim-tips-and-tricks/), [Vim Basic Navigation](http://www.thegeekstuff.com/tag/vim-basic-navigation/), [vim command](http://www.thegeekstuff.com/tag/vim-command/), [Vim Commands](http://www.thegeekstuff.com/tag/vim-commands/), [vim editor](http://www.thegeekstuff.com/tag/vim-editor/), [Vim Editor Examples](http://www.thegeekstuff.com/tag/vim-editor-examples/), [Vim Navigation Commands](http://www.thegeekstuff.com/tag/vim-navigation-commands/), [vim tips](http://www.thegeekstuff.com/tag/vim-tips/)

{ 13 comments… [add one](http://www.thegeekstuff.com/2009/03/8-essential-vim-editor-navigation-fundamentals/" \l "commentform) }

* **vimer** March 11, 2009, 10:06 pm

Navigation using marks is very important without it vim is as good or bad as Emacs. A mark can be put using the letter m followed by any other latter. A small letter used for a mark is scoped within the file, while a capital letter used for a mark is scoped across files. To jump to a mark use `a or ‘a, I prefer the `a since it takes me to the exact position where I left the mark. To look at where the marks are use :marks in the ed mode.

overall steps for mark navigation are;  
Put mark on a line using ma go to some other place edit and then jump back using `a  
Say I am in some file I put a mark using mA go to some other file in same buffer and then jump back using `A, this is really cool I get to jump back to previous file with just 2 keystrokes.

Finally, ctags are also very important for code navigation without ctags vim is as bad as Eclipse, use Ctrl-] to jump to a function definition and Ctrl-t to comeback from the tag stack.

Cheers,

[LINK](http://www.thegeekstuff.com/2009/03/8-essential-vim-editor-navigation-fundamentals/" \l "comment-7014" \o "permalink to this comment)