

# 10 Easy Steps to Becoming a Vim Expert

Joe 'Zonker' Brockmeier  
openSUSE Community Manager



**Novell**®

# A little goes a long way

- You can do a lot as long as you understand the basics in Vim:
  - Movement
    - > h, j, k, l
    - > w, e, \$, 0, gg, G
  - Search and replace
  - Undo
  - Basic highlight / copy and paste
  - Using counts with operators (d5w, d0, etc.)

# You can do *more* if you dig in

- Vim has a steep learning curve
- If you're just starting, be sure to run through `vimtutor`
- If you spend a lot of time editing text, it's worth it!

# 10: Using Abbreviations

- You can replace text on the fly
- You can make sure that you spell things right (use abbreviations to correct poor spelling...)
- Avoid spelling out long words, names.

# To add an abbreviation

Enter command mode and use the ab command:

```
:ab zkr Zonker
```

```
:ab orly O RLY?
```

```
:ab yrly YA RLY!
```

# Using abbreviations

To use the abbreviation for O RLY? type `orly` and hit space.

Note that Vim will convert any string with “orly” into “O RLY?” unless you limit the abbreviation to insert mode, like so:

```
:iab orly O RLY?
```

To avoid using an abbreviation, type `Ctrl-v` after the string, which tells Vim not to expand the string.

## 9: Use tabs

- As of Vim 7, Vim has a tabbed interface for its GUI (gvim) and terminal mode.
- You can use tabs to consolidate multiple Vim instances into one Vim instance.

```
File Edit View Terminal Tabs Help
SaX.log Xgl.0.log NetworkManager X
Apr 19 21:59:19 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
Apr 20 00:38:41 winslow NetworkManager: <debug> [1208666321.967835] nm_hal_device_added(): New dev
ice added (hal udi is '/org/freedesktop/Hal/devices/usb_device_46d_c019_noserial').
Apr 20 00:38:41 winslow NetworkManager: <debug> [1208666321.999213] nm_hal_device_added(): New dev
ice added (hal udi is '/org/freedesktop/Hal/devices/usb_device_46d_c019_noserial_if0').
Apr 20 00:38:42 winslow NetworkManager: <debug> [1208666322.040158] nm_hal_device_added(): New dev
ice added (hal udi is '/org/freedesktop/Hal/devices/usb_device_46d_c019_noserial_if0_logicaldev_in
put').
Apr 20 03:04:42 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
Apr 20 06:28:45 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
Apr 20 07:59:22 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
Apr 20 08:45:59 winslow NetworkManager: <debug> [1208695559.050136] nm_hal_device_added(): New dev
ice added (hal udi is '/org/freedesktop/Hal/devices/usb_device_5ac_1290_2c9b0582c02c880dd4fef7e1df
d8f723360cf138').
Apr 20 08:45:59 winslow NetworkManager: <debug> [1208695559.078679] nm_hal_device_added(): New dev
ice added (hal udi is '/org/freedesktop/Hal/devices/usb_device_5ac_1290_2c9b0582c02c880dd4fef7e1df
d8f723360cf138_if0').
Apr 20 08:57:22 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
Apr 20 09:24:15 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
Apr 20 09:41:04 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
Apr 20 09:48:13 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
29099,1 99%
```



# How to use tabs

You can use tabs when you start Vim:

```
vim -p file1 file2 file3
```

You can open tabs within Vim:

```
:tabnew or :tabnew filename
```

You can use `tabf` to find files and open in a new tab:

```
:tabf file*
```

# Moving between tabs

Using Vim's GUI, just click on the tabs.

In the terminal mode use:

**:tabn** to move to the next tab

**:tabp** to move to the previous tab

**:tabfirst** to move to the first tab

**:tablast** to jump to the last tab

## 8: Mappings

Mappings are a way to assign a hotkey or key combination to an action.

You can reduce this:

```
:call functionName()
```

To this:

```
Ctrl-n
```

# How to use mappings

To set a mapping, use one of the mapping commands:

1. `cmap` handles command-line mappings.
2. `imap` handles insert-only mappings.
3. `map` maps keys in normal, visual, and operator-pending mode.
4. `map!` maps keys in Vim's command and insert modes.
5. `nmap` maps keys in normal mode only.
6. `omap` maps keys in operator-pending mode only.
7. `vmap` maps keys in visual mode only.

(Vim has plenty of modes...)

# How to use mappings, Part II

Run the following:

```
:map <F10> <Esc>:tabnew<CR>
```

Which assigns :tabnew to F10.

So, **:mapcommand <key> <command>**

Another example:

```
:map <S-A-F2> <Esc>:syn off<CR>
```

# Get rid of mappings

If you want to get rid of a mapping:

```
:unmap <F10>
```

## 7: Viewports

- You can split Vim's “viewport” to see multiple files or the same file in different locations.
- You can split vertically or horizontally.
- You can choose how many lines each viewport takes.

# Horizontal split

```
File Edit View Terminal Tabs Help
Apr 20 07:59:22 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
Apr 20 08:45:59 winslow NetworkManager: <debug> [1208695559.050136] nm_hal_device_added(): New dev
ice added (hal udi is '/org/freedesktop/Hal/devices/usb_device_5ac_1290_2c9b0582c02c880dd4fef7e1df
d8f723360cf138').
Apr 20 08:45:59 winslow NetworkManager: <debug> [1208695559.078679] nm_hal_device_added(): New dev
ice added (hal udi is '/org/freedesktop/Hal/devices/usb_device_5ac_1290_2c9b0582c02c880dd4fef7e1df
d8f723360cf138_if0').
Apr 20 08:57:22 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
Apr 20 09:24:15 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
~
~
NetworkManager 29099,1 Bot
Apr 20 06:28:45 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
Apr 20 07:59:22 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
Apr 20 08:45:59 winslow NetworkManager: <debug> [1208695559.050136] nm_hal_device_added(): New dev
ice added (hal udi is '/org/freedesktop/Hal/devices/usb_device_5ac_1290_2c9b0582c02c880dd4fef7e1df
d8f723360cf138').
Apr 20 08:45:59 winslow NetworkManager: <debug> [1208695559.078679] nm_hal_device_added(): New dev
ice added (hal udi is '/org/freedesktop/Hal/devices/usb_device_5ac_1290_2c9b0582c02c880dd4fef7e1df
d8f723360cf138_if0').
Apr 20 08:57:22 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
Apr 20 09:24:15 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
NetworkManager 29103,1 Bot
```



# Vertical split

```

File Edit View Terminal Tabs Help
Apr 19 21:37:24 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface eth1
Apr 19 21:39:48 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface eth1
Apr 19 21:42:04 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface eth1
@
NetworkManager 29083,1 99%
Apr 19 21:37:24 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface eth1
Apr 19 21:39:48 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface eth1
Apr 19 21:42:04 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface eth1
Apr 19 21:44:08 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface eth1
Apr 19 21:46:27 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface eth1
Apr 19 21:48:33 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface eth1
NetworkManager 29083,1 99% NetworkManager 29099,1 Bot
:10 split

```

# Using Viewports

Simple:

**:sp** or **:split**

Assign a number of lines:

**:15 split**

Open a file:

**:split filename**

Vertical split:

**:vsplit filename**

# Moving between viewports

Cycle between windows:

**Ctrl-w Ctrl-w**

*(All Ctrl-w commands are in command mode – Ctrl-w deletes a word in insert mode...)*

Normal movement keys work with viewports...

Move one viewport down:

**Ctrl-w j**

Move one viewport to the left:

**Ctrl-w h**

Reduce or enlarge viewport:

**Ctrl-w – or Ctrl-w +**

# More viewport goodness

“Rotate” the windows to the right

**Ctrl-w r**

“Rotate” the windows to the left

**Ctrl-w R**

Make viewports equal size:

**Ctrl-w =**

Close active viewport:

**Ctrl-w q** *or* **:q**

## 6: Marks

- Marks are bookmarks or placeholders in a file.
- You can use any letter in the alphabet.
- Lowercase letters are specific to a file.
- Uppercase letters are “global” and specific to a file.
  - Using a global mark will open the file and close the current file.

# Using marks

To set a mark, in command mode:

**ma**

To set a global mark:

**mA**

To use a mark:

**`a** (that's a backtick)

Marks can be used with commands:

**: 'a, 'z w *filename***

**: 'a, 'k s/*searchstring/replacestring/g***

# Using marks, part II

To display marks:  
**:marks**

```
File Edit View Terminal Tabs Help
th1
Apr 20 06:28:45 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
Apr 20 07:59:22 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew) for interface e
th1
Apr 20 08:45:59 winslow NetworkManager: <debug> [1208695559.050136] nm_hal_device_added(): New dev
ice added (hal udi is '/org/freedesktop/Hal/devices/usb_device_5ac_1290_2c9b0582c02c880dd4fef7e1df
d8f723360cf138').
Apr 20 08:45:59 winslow NetworkManager: <debug> [1208695559.078679] nm_hal_device_added(): New dev
ice added (hal udi is '/org/freedesktop/Hal/devices/usb_device_5ac_1290_2c9b0582c02c880dd4fef7e1df
d8f723360cf138_if0').
:marks
mark line col file/text
' 10 0 Feb 8 08:14:11 linux NetworkManager: <debug> [1202487251.453920] nm_system_device
g 10 0 Feb 8 08:14:11 linux NetworkManager: <debug> [1202487251.453920] nm_system_device
h 29089 0 Apr 19 21:50:55 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew)
A 29098 0 Apr 20 06:28:45 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew)
0 29099 0 Apr 20 07:59:22 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew)
1 218 18 /etc/sysconfig/network/config
2 1 0 /etc/networks
3 228 0 /etc/sysconfig/network/config
4 518 23 /etc/init.d/network
5 1 11 /etc/sysconfig/network/ifcfg-eth1
6 13 0 /etc/NetworkManager/dispatcher.d/netcontrol_global_hooks
7 1 0 /etc/NetworkManager/dispatcher.d/netcontrol_
8 8 1 /etc/NetworkManager/dispatcher.d
9 220 18 /etc/sysconfig/network/config
" 29098 0 Apr 20 06:28:45 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew)
[ 1 0 Feb 8 08:14:11 linux NetworkManager: <info> starting...
] 29112 0 Apr 20 09:59:50 winslow NetworkManager: <info> DHCP daemon state is now 3 (renew)
Press ENTER or type command to continue
```

## 5: Jumps

- Jumps are a recording of your movements through files.
- Much like Firefox's history.
- Jumps can include marks, but also include any other movements through Vim's history.



# Using Jumps

Use `:jumps` to see your movements within a file, like so:

```
:jumps
jump line  col  file/text
 100    4    0  nautilus-debug-log.txt
  99   44    0  nautilus-debug-log.txt
  98   26    0  Writing/opensuse.org/beta-annc.html
  97   30    5  Writing/opensuse.org/beta-annc.html
  96   36  195  Writing/opensuse.org/beta-annc.html
  95   17    0  Writing/opensuse.org/beta-annc.html
  94   83   33  Writing/opensuse.org/beta-annc.html
  93   13  378  Writing/opensuse.org/beta-annc.html
  92    4    0  Writing/opensuse.org/beta-annc.html
  91   11    0  Writing/opensuse.org/beta-annc.html
  90    1    0 .opendns.txt
```

# Cycling through your jumps

Using **Ctrl-o** and **Ctrl-i** will “cycle” through your jump list:

**Ctrl-o** will move you to the “last” jump (1<sup>st</sup> in the history list).

**Ctrl-i** will move you to the oldest jump.

This history is saved in your **.viminfo** file.

# Other useful Vim movement commands

Also be familiar with Vim movement commands:

**G** = bottom of file

**gg** = top of file

**H** = top of screen

**M** = middle of screen

**L** = bottom of screen

## 4: Folding

- Folding allows you to see more of a file in one screen.
- Great when working on code, HTML, etc.
- Folding works in command mode and visual mode.
- Folds can be “nested” so multiple folds can be folded into one big fold.

# Working with folds...

**zf** starts the folding command

Can then enter a movement command (like **12j**) so **zf12j** will “fold” 13 lines (12 + current line).

Can fold in visual mode using visual command and then enter **zf**.

**zo** will “open” a fold.

Deleting a folded “line” will delete the entire section. Same is true of Yank commands, etc.

**zk** and **zj** will move to next and previous folds, respectively.

**zd** deletes a fold

**zm** and **zM** close folds.

## 3: Use Vim scripts

- Vim.org has thousands of scripts that probably do what you want.
- Everything from converting Vim to a Bash IDE to a Twitter client.
- If you don't find what you want, write your own.
  - Scripts usually live in `~/ .vim/plugins`
- Find scripts at <http://www.vim.org/scripts/>
- Source scripts with `:source scriptname`
- Syntax files live in the `~/ .vim/syntax` directory.

## 2: Use Vim's “easy” mode

- You don't always have to do things the hard way!
- Vim's easy mode acts (mostly) as a standard text editor.
- Easy mode works in text and GUI modes, but I recommend GUI if you're going to go with that...
- You can still enter normal Vim mode from the “easy” mode, but have all the advantages of Vim in easy mode.
- Best way to introduce newbies to Vim!

# Using easy mode

- Run `evim` or `gvim -y` from the command line.
- Provides a context menu
- Provides familiar shortcuts (like `Ctrl-v` for paste, rather than Vim's traditional paste commands)
- Still offers Vim's tools & features.
- Displays Vim's usual shortcuts in menus.
- Leave easy mode by going to:
  - `Edit` -> `Global Settings` -> `Toggle Insert Mode`
  - or using `:set im!`



# 1: Hack your .vimrc

- If you use commands frequently, add them to your .vimrc.
- Set Vim up exactly as you like it.
- Commands are used in the vimrc mostly as they are when entered in Vim, just skip the : first...
- Comments start with “
- Can use mappings, set / unset commands, and much more.
- Save time – start with an existing vimrc (see /etc/vimrc on most distros... or use Google...)

# Thanks!

**Joe 'Zonker' Brockmeier**  
openSUSE Community Manager  
<http://zonker.opensuse.org/>  
zonker@opensuse.org



**Novell®**