VIM

David Orman

July 14, 2009

Contents

Chapter 1

VIM Basics

The goal of this course is teach and reinforce a solid foundation of VIM basics, for both beginners and advanced users alike. As I was writing this course material, I used VIM. Any time I found myself doing something that caused me to think *there must be a better way of doing this*, I went through the VIM documentation to find a more efficient way to perform the editing I was attempting. The results of my prior knowledge of VIM, as well as the knowledge I obtained while writing this document are contained within.

I welcome any and all feedback, if you have any suggestions for improvement please let me know. Also, if you have any difficulty digesting any of this information, let me know so that I can elaborate more on that topic. I will attempt to continue updating this documentation using your suggestions as time progresses, in order to provide more accurate and helpful documentation.

This document is available in PDF format, email david.orman@rackspace.com if you would like a copy.

1.1 Modes

VIM has three main modes, inherited from vi. The discussion of the extended modes VIM offers will be covered in later courses, as the three primary modes are all that required to successfully use VIM for most editing tasks.

Command mode allows you to perform operations on the text, without actually *inserting* the text directly. For example, in command mode, you can perform string substitution, write out the file, delete lines, and so forth.

When using command mode, you can use modifiers to perform operations on different numbers of objects, lines, paragraphs, and more. We will go into the different commands in more detail as we progress through this course, but a simple example would be the command *dd* which deletes a line.

Insert mode allows you to actually insert text, in various different ways. For example, you can *append* text after the cursor position and *insert* text at the current position.

Finally, visual mode permits you to perform operations on selected areas of text. This is quite useful in adding # comments at the beginning of multiple lines. Visual mode is used in combination with insert mode and command mode in order to perform the operations requested.

1.1.1 Command mode

To enter command mode, from insert or visual mode, press *esc*.

1. *:e filename* opens a file for editing.
2. *:w* writes the buffer to the file.
3. *:q* quits VIM.
4. *:q!* exits VIM without saving changes.
5. *:x* exits VIM, saving the changes that might have been made. If there are no changes, simply exits VIM without saving.

1.1.2 Insert mode

To enter insert mode from command mode, you can:

1. Press *i* to switch to insert mode before the current position.
2. Press *a* to switch to insert mode after the current position.
3. Press *I* to jump to the first non-blank character in the current line and switch to the insert mode.
4. Press *A* to jump to the last character of the current line and switch to the insert mode.
5. Press *o* to insert a new line after the line the cursor is positioned within.
6. Press *O* to insert a new line prior to the line the cursor is positioned within.

1.1.3 Visual mode

To enter visual mode from command mode, you have a few options:

1. Press *v* to switch to character-oriented visual mode.
2. Press *V* to switch to line-oriented visual mode.
3. Press *cntl-v* to switch into block-visual mode.

1.2 Navigation

1.2.1 Proper keyboard navigation

These commands are only functional in the command mode, otherwise they will enter text into the file open in the buffer.

Simple character-based navigation

1. Press *h* to move the cursor to the left.
2. Press *l* to move the cursor to the right.
3. Press *j* to move the cursor downwards.
4. Press *k* to move the cursor upwards.

Simple line and page based navigation

1. Press *:line#* to move to the specified line number.
2. Press *column#|* to move to the specified column number in the current line.
3. Press *gg* to move to the top of the file.
4. Press *G* to move to the end of the file.

Moving around within a line

1. Press *0* to move to the first column of the line.
2. Press ˆ to move to the first non-black character on the line.
3. Press *w* to move to the next word.
4. Press *W* to move to the next word, ignoring punctuation (such as . . .).
5. Press *e* to move to the end of the current word.
6. Press *b* to move to the beginning of the current word.
7. Press *B* to move to the beginning of the current word, ignoring punctuation (such as . . .).
8. Press *ge* to move to the previous word ending.
9. Press *gE* to move to the previous word ending, ignoring punctuation (such as . . .).
10. Press *g_* to move to the last non-blank character of the line.
11. Press *$* to move to the last character of the line.

1.3 Editing

1.3.1 Fundamental editing commands

1. Press *d* to delete the characters from the current cursor position to the position given by the next command. For example, *d0* will delete everything up to the beginning of the line. *d$* will delete everything until the end of the line.

2. Press *c* to change character from the current cursor position to the position indicated by the next command. For example, *c$* will change the characters from the current cursor position until the end of the line.

3. Press *x* to delete the character under the cursor.

4. Press *X* to delete the character before the cursor (this operates like backspace does).

5. Press *y* to copy the characters from the current cursor position until the position indicated by the next command. For example, *y$* will copy the characters from the position the cursor is at until the end of the line.

6. Press *p* to paste the previously deleted or yanked/copied text after the current cursor position.

7. Press *P* to paste the previously deleted or yanked/copied text before the current cursor position.

8. Press *r* to replace the current character with the character pressed after the *r*.

9. Press *.* to repeat the last insert or editing command.

10. Using *d*, *c*, or *y* performs the operation on the entire line. For example, *dd* will delete the entire line.

Using a numeric which I will call *X* in this case, prior to the above listed commands will perform the operation *X* number of times. For example, *3dd* will delete 3 lines.

1.3.2 Editing using visual block mode

Visual block mode allows you to insert characters on each line within/before/after a selected area. For instance, in order to insert a # in front of five lines, press *cntl-v*, select the area in which you wish to insert text before, then press *I*. Type the text you'd like to insert prior to the selected lines, and then press *esc* to exit insert mode. At this point, the text will be inserted prior to the area selected. Use *A* instead of *I* in order to insert text **after** the selected area, on the lines selected.

You can also do a substitution, in much the same way. This replaces the text in the selected area with the text you type, on each line in the selected area. To do this, press *cntl-v* to select the area you wish to replace. Then, press *s* and type what you wish to replace the text in the selected area with. Finally, press *esc* to leave insert mode, and VIM will replace the lines in the selected area with the text you have input.

One more example! You can insert text explicitly at the end of lines in the selected areas. Keep in mind, when using *cntl-v* to select text, you are selecting by column and row. If you use *cntl-v$*, then press *A* when making your selection, you are telling VIM to explicitly add text to the end of the lines, not at the exact location of the selection.

1.3.3 Editing using text objects in visual mode

Using text objects for operations allow you to perform editing functions more efficiently by operating on commonly encountered *objects*, such as *words*, *sentences*, and *paragraphs. Inner* objects just contains the characters of that object, not trailing spaces. They can also contain the object between *()*. The *a* object operations include both the characters of that object, as well as trailing spaces and *()*.

1. Press *viw* to operate on the inner word.
2. Press *vaw* to operate on a word.
3. Press *viW* to operate on the inner word, including trailing space.
4. Press *vaW* to operate on a word, including trailing space.
5. Press *vis* to operate on the inner sentence.
6. Press *vas* to operate on a sentence.
7. Press *vip* to operate on the inner paragraph.
8. Press *vap* to operate on a paragraph.
9. Press *vi(* or *i)* to operate on an inner block within the *(* or *)*, including trailing space.
10. Press *va(* or *a)* to operate on a block within the *(* or *)*.
11. Press *vi<* or *i>* to operate on an inner block within the < or >, including trailing space.
12. Press *va<* or *a>* to operate on a block within the < or >.
13. Press *vi{* or *i}* to operate on an inner block within the { or }, including trailing space.
14. Press *va{* or *a}* to operate on a block within the { or }.

15. Press *vi*¨ to operate on an inner block within the ¨ and ¨, including trailing space.
16. Press *va*¨ to operate on a block within the ¨ and ¨.
17. Press *vi*` to operate on an inner block within the ` and `, including trailing space.
18. Press *va*` to operate on a block within the ` and `.

1.4 Time saving functionality

1.4.1 Undo and redo

1. Press *u* to undo an operation.
2. Press *cntl-r* to redo an operation that has been undone.

1.4.2 Search and replace

1. To search and replace within a file, use *:%s/old/new/g*

Chapter 2

Exercises

These examples will be used during our class to familiarize us with actually performing these basic operations.

2.1 Navigation

2.1.1 Basic Navigation

1. Open httpd.conf.

2. Practice moving the cursor around, without using the arrow keys. Move right, left, up and down using the *h, j, k, and l* keys.

3. Practice moving to the top and the bottom of the example file. Use *gg* and *G*.

2.1.2 Advanced Navigation

1. Open httpd-vhosts.conf.

2. Practice moving around using object-based navigation. For example, moving from word to word, both ignoring and not ignoring punctuation. Use the commands *w, W, e, E, b, B, 0, and $*.

2.2 Editing

2.2.1 Basic Editing

1. Open httpd.conf.

2. Use the commands *d, c, x, X, y, p, P, and r* to modify text. Make sure to experiment with each one.

2.2.2 Advanced Editing

1. Open httpd.conf.
2. Use the commands from exercise one in combination with the object keywords *iw, aw, is, as, ip, and ap* in order to modify objects.

2.3 Time saving

2.3.1 Undo and Redo

1. Open httpd.conf.
2. Using the some of the commands practiced in exercises from the editing portion of this course, modify the file.
3. Use *u and cntl-r* to undo and redo your changes. Remember, you can undo repeatedly, to continue stepping back through your modifications, as well as stepping forward redoing the changes as well.

2.3.2 Search and Replace

1. Open httpd-vhosts.conf
2. Using search and replace, modify the file so that the virtual hosts listen on port 80.
3. Using search and replace, modify the file so that the virtual hosts have document roots of the format */var/www/docs/ . . .*