# VI (and Clone) Editor Reference Manual

**Miles O'Neal**
*(meo@rru.com)*

**Susan Liebeskind**
*(shl@cc.gatech.edu)*

**Systems & Software Solutions**

December 12, 1998

ABSTRACT

This manual gives a brief overview of the more common vi commands. It is a companion to the Vi Introductory Guide.



**Revision History:**

*Rev. 1.1 Jan 9, 1990 meo*
*Rev. 1.2 Jan 1, 1992 meo*
*Rev. 1.3 Mar 22, 1994 meo*
*Rev. 1.4 Aug 3, 1998 meo*
*Rev. 1.5 Dec 12, 1998 meo*

# Table of Contents

## 1. Introduction

This is a brief, introductory reference for vi. It is supplemental to the VI Introductory Guide and your vendor documentation. The information here also applies to all the clones of vi, such as vim, elvis, and stevie.

Vi is actually a very powerful editor, and is organized in a logical fashion. It is different from other editors or word processors, which may make learning it just a little bit tricky for some people. If you approach it analytically and fearlessly, you should become proficient with it in a short period of time.

I highly recommend experimentation. For obvious reasons you should restrict early experimentation to test files.

Some non-display character keys and other keys have special functions. These keys will be shown by placing their name (such as RETURN) in square brackets, like this: **[RETURN]**.

Unlike most non UNIX[1] products, vi is case-sensitive. Be careful with the **[CAPS LOCK]** key (except in input mode). If things are acting strange, make sure **[CAPS LOCK]** is off. If you are unsure about this, a good test is to press the 'j' key; if it moves the cursor down a line, **[CAPS LOCK]** is not on; if it joins the next line to the current line, **[CAPS LOCK]** is on.

Not all of the commands mentioned here are covered in the VI Introductory Guide. There are also far more commands available in vi than are covered here - for further information check your system documentation or get one of the books available on vi.

| **Special Keys** | | |
|---|---|---|
| **Key** | **Command Mode** | **Text Mode** |
| **Arrow Keys** | cursor movement | N/A |
| **[CTRL]** | used with other keys for extra commands | insert control characters in text |
| **[ENTER]** or **[RETURN]** | down 1 line | normal function |
| **[ESC]** | N/A | leave text mode |
| **Space Bar** | move right 1 character | normal function |
| **[TAB]** | N/A | normal function |

When a command is shown as a combination of the '^' (caret) and another character, as in **^V**, this means press the **[CTRL]** key first, and hold it down while you press and release the other key.

## 2. Command Mode vs Text Mode

Vi is always either in command mode or text (input) mode. In text mode any character key pressed is enetered into the file. Most other typewriter keys have their normal effect. On some systems to enter a **[CTRL]** key sequence, you may first have to enter **^V** before the sequence you want.

In command mode nearly every key on the keyboard performs some command or modifies the next command. Some of these commands may be difficult to recover from, so be careful in command mode to enter only the commands you wish to enter.

The **[RETURN]** key is not needed in command mode except with commands that begin with a ':' (colon) and with the search comands.

| **Mode Change Commands** | |
|---|---|
| **a** | append text after cursor |
| **i** | insert text before cursor |
| **o** | open new line after current line & add text |
| **O** | open new line before current line & add text |
| **[ESC]** | leave text mode |

## 3. Cursor Motion

Cursor motion should nominally include use of the cursor keys. If your terminal lacks or has problems with arrow keys in vi command mode, the 'h', 'j', 'k', and 'l' keys will perform the same functions, as noted below.

| **Cursor Motion Commands** | |
|---|---|
| h | move back 1 character |
| l | move forward 1 character |
| j | move down 1 line |
| k | move up 1 line |
| | |
| b | back to beginning of word |
| e | forward to end of word |
| w | forward to beginning of next word |
| ^ | go to first displayable character of line |
| 0 | go to beginning of line |
| $ | go to end of line |
| | |
| <CTRL>F | forward 1 screen |
| <CTRL>B | backward 1 screen |
| <CTRL>D | down (forward) 1/2 screen |
| <CTRL>U | up (backward) 1/2 screen |

## 4. Editing Commands

These are the commands that actually manipulate text. the commands listed here include those which delete, replace, search, cut, and paste text, as well as those used for saving text and abandoning an edit session. Finally, a few miscellaneous commands are included which don't fit into the other categories.

### 4.1. Delete (Cut) Commands

In some editors the delete commands may be called cut commands. The last item deleted is saved in a buffer and may be put (or pasted) elsewhere in the file as noted later.

| **Delete Commands** | |
|---|---|
| x | delete character |
| dw | delete rest of word |
| d$ | delete rest of line |
| D | delete rest of line |
| dd | delete line |

### 4.2. Replace, Change and Substitute Commands

Some editors provide commands to substitute new text occurances of search strings; this is not the same thing. These following vi commands allows you to replace some number of characters, words, lines or parts of lines with new text.

The change and substitute commands, the editor puts you into insert mode until you press **[ESC]**. The replace command replaces 1 (or more) characters with the next character you type.

| **Replace Commands** | |
|---|---|
| r | replace character |
| cw | change rest of word |
| c$ | change rest of line |
| C | change rest of line |
| Ns | substitute text for N characters |

### 4.3. Search Commands

These commands let you search for a text string, which may include regular expressions. They must be followed by a **[RETURN]**.

| **Search Commands** | |
|---|---|
| /text | search forward for text |
| ?text | search backward for text |
| n | search in same direction for next occurance of last-searched-for text |
| N | search in other direction for next occurance of last-searched-for text |

### 4.4. Undo Command

IN the real vi, the undo command only undoes the last command, even if that is an undo command. Repeated undo commands simply toggle the effect of the last command before the series of undo commands.

Some vi clones, such as **vim**, allow multiple undo's. These usually offer a way to set the undo level to the standard vi mode. See your editor's man page or reference manual for details.

| Undo Command | |
|---|---|
| u | undo last command (BE CAREFUL WITH THIS) |

### 4.5. Saving & Exiting Commands

These commands must be followed by a **[RETURN]**.

| Saving & Exiting Commands | |
|---|---|
| :w | write (save) the file |
| :w NAME | write the file and name it NAME |
| :w! NAME | rewrite the file named NAME |
| :wq | write the file and quit the editor |
| :q | quit the editor |
| :q! | quit the editor (abandoning any changes) |

### 4.6. Search & Replace Command

To find & replace all occurances of a particular text string, use the command:

```
:%s/text1/text2/g[RETURN]
```

which replaces all occurances of text1 with text2.

### 4.7. Yank (Copy) Commands

The vi yank command is similar to the copy command in many editors. It copies the text into a buffer. The text may be put (pasted) elsewhere in the file as described under the Put command.

| Yank Commands | |
|---|---|
| yw | yank rest of word |
| y$ | yank rest of line |
| yy | yank entire line |
| Y | yank entire line |

### 4.8. Put (Paste) Commands

The vi put command is similar to the paste command in many other editors. It will paste whatever is in the buffer from either the previous delete or yank command.

| Put Commands | |
|---|---|
| p | put yanked/deleted text before cursor |
| P | put yanked/deleted text after cursor |

If you yanked/deleted a whole line (or group of lines), p and P paste the text before or after the current line, respectively; otherwise, they paste the text before or after the cursor, respectively, on the current line.

### 4.9. Miscellaneous Commands

| Miscellaneous Commands | |
|---|---|
| . | repeat last command (BE CAREFUL WITH THIS) |
| J | join next line to end of this line |

| NOTE |
|---|
| The period, or dot, command repeats most commands, but not quite all. It should work with all of the commands listed in this document; if you aren't sure of what you're doing, save your work before you try something. |

## 5. Miscellaneous Notes

Most commands (other than those beginning with a colon (':') may be preceded by a repeat count. For example,

```
3dd
```

would delete 3 lines, starting with the current one.

Vi will not let you delete more lines than are in the file, so the above example would not work on the last line; vi would beep at you instead.

---

**Notes:**

1. *UNIX is a trademark of Western Electric, AT&T, SCO, or whoever bought it this week.*

---

Last updated: *12 December 1998*

*Miles O’Neal <meo@XYZZYrru.com> [remove the XYZZY to make things work!] c/o RNN / 11501 Johnson Rd / Leander, TX / 78641-5823*