

<i>Document Number:</i>	<i>Copy Number:</i>
-------------------------	---------------------



**TATA CONSULTANCY SERVICES
GUWAHATI**

vILP – Unix – Basics of Editors and vi

Content Manual

Version 1.1

May 2014

TATA CONSULTANCY SERVICES

A Division of **Tata Sons Limited**



DOCUMENT RELEASE NOTICE

Notice No.

Client

Project

Document details

Name	Version No.	Author	Description
Basics of Editors and vi	1.1	Progoti Choudhuri	Helps understanding use of editor in unix-based system

Revision details:

Action taken (add/del/chg	Preceding page No.	New page No.	Revision description

Change Register serial numbers covered:

The documents or revised pages are subject to document control.

Please keep them up-to-date using the release notices from the distributor of the document.

These are confidential documents. Unauthorised access or copying is prohibited.

Approved by : _____

Date: _____

Authorised by: _____

Date: _____

Client
Project
Document Name Basics of **Editors and
vi in Unix**
Release Notice Reference (for release)

Rev. No.	Revision date	Revision description	Page No.	Pr ev pa	Action taken	Addenda/New page	Release Notice reference
1.1	09-Jul-2014	Reviewed and incorporated					

CONTENTS

CHAPTER 1- BASICS OF EDITORS AND VI	4
1.1 Objective.....	4
1.2 UNIX Text Editors.....	4
1.3 The vi Editor	4
1.3.1 Edit Mode Commands	5
1.3.2 Save and Exit from vi	6
1.3.3 Cursor Movement Commands	6
1.3.4 Search and Repeat-Search Commands.....	6
1.3.5 Deletion and Modification Commands	7
1.3.6 Commands to Move or Copy Lines	7
1.3.7 Working with vi.....	8
1.3.8 Customizing vi editor environment	13
1.4 Command Summary.....	14
1.5 Video	15
1.6 Quiz.....	16

CHAPTER 1- BASICS OF EDITORS AND VI

1.1 Objective

- To understand the editors in UNIX
- use of vi editor

1.2 UNIX Text Editors

To write a program or a paper even to edit a mail message, you need to create, write, store, retrieve, view, and edit files. In unix several text editors available for this . They are ed, nedit, emacs and vi .

One of the first end-user programs hosted on the system and standardised in UNIX-based systems ever since is the ed editor. **ed** is a line editor for the UNIX operating system.

1.3 The vi Editor

The vi editor is available on almost all Unix systems. vi can be used from any type of terminal because it does not depend on arrow keys and function keys--it uses the standard alphabetic keys for commands.

vi (pronounced "vee-eye") is short for "vi"sual editor. It displays a window into the file being edited that shows 24 lines of text. vi is a text editor not a "what you see is what you get" word processor. vi lets you add, change and delete text, but does not provide such formatting capabilities as centering lines or indenting paragraphs.

This help note explains the basics of vi:

- 1.opening and closing a file
- 2.moving around in a file
- 3.elementary editing

vi has many other commands and options not described here. Online resources could guide you here.

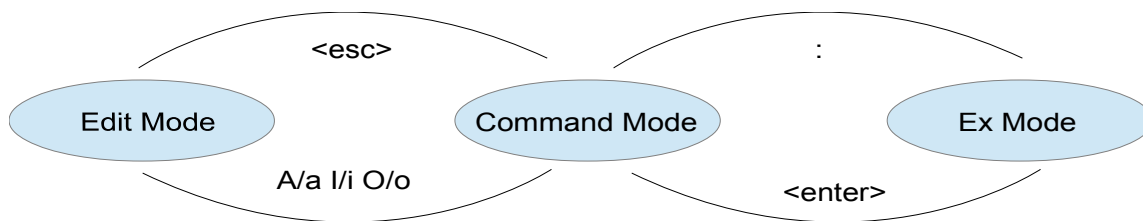
It is a visual editor used to enter and edit text files. Invoking vi with/without filename puts it in command mode:

vi [<file name>]

vi works in three different modes:

- Edit Mode - where any key is entered as text
- Command Mode - where keys are used as commands
- Ex Mode - ex commands can be entered in last line to act on text

Relationship between 3 modes



1.3.1 Edit Mode Commands

Command	Action
a	Appends text to right of cursor
A	Appends text at end of line
i	Inserts text to left of cursor
I	Inserts text at beginning of line
o	Inserts blank line below + inserts text
O	Inserts blank line above + inserts text
rx	Replace current character with char x
Rtext<Esc>	replaces character with text

1.3.2 Save and Exit from vi

Save and Exit Commands of ex Mode

Command	Action
:w	save file and remain in edit mode
:wq	save file and quit edit mode
:w <filename>	write buffer to file name
:q	quit editing mode when no changes are made
:q!	quit editing mode, abandoning changes
:x	save file and quit editing mode
ZZ	save file and quit edit mode

1.3.3 Cursor Movement Commands

Command	Action
h or backspace	Moves Cursor Left
l or spacebar	Moves Cursor Right
j	Moves Cursor Down
k	Moves Cursor Up
nG	Go to line number n
\$	Moves cursor to end of line
w	Moves cursor to next word
b	Moves cursor back to previous word
e	Moves cursor forward to end of word

1.3.4 Search and Repeat-Search Commands

Command	Action
/pat	Search forward for pattern pat
?pat	Search backward for pattern pat
n	Repeat search in previous search direction
; or ,	Repeat last character search in same or opposite direction (by f/F/t/T) only in current line (repeat factor works)

1.3.5 Deletion and Modification Commands

Command	Action
dw	Delete Current Word
dd	Delete Current Line
d\$ or D	Delete from cursor to end of line
cw	Change Current Word
cc	Change Current Line
c\$ or C	Change from cursor to end of line
x	Delete character in current cursor position
X	Delete character before cursor
.(dot)	Repeat last editing instruction
u	Undo single last change
U	Restore all changes to line since cursor moved to it

1.3.6 Commands to Move or Copy Lines

[<named-buffer>][n]dd
Cut (delete) n lines starting from current line

[<named-buffer>][n]yy
Copy n lines starting from current line

[<named-buffer>]p
Put yanked text after current cursor position

[<named-buffer>]P
Put yanked text before current cursor position

Note: Named-buffer is useful for copying an area from one file into another. Open some files simultaneously by vi. Mark an area in one file by dd or yy; move to another file (by :next) to paste (by p or P) the named area; then say :rewind to return to the parent file.

1.3.7 Working with vi

You may use vi

- to open an already existing file by typing
vi filename, where "filename" is the name of the existing file. If the file is not in your current directory, you must use the full pathname.
- To create a new file by typing
vi newname, where "newname" is the name you wish to give the new file.

To open a new file called "testvi," enter vi testvi. In the screen, you will see blank lines, each with a tilde (~) at the left, and a line at the bottom giving the name and status of the new file:

```
~  
"testvi" [New file]
```

Modes in vi : vi works in three modes:

- Command mode
- Insert mode
- Ex mode or Colon mode (:))

In command mode, the letters of the keyboard perform editing functions (like moving the cursor, deleting text, etc.). To enter command mode, press the escape <Esc> key.

In insert mode, we can type texts to form words and sentences. Unlike many word processors, vi starts up in command mode.

Entering Text

In order to begin entering text in an empty file, you must change from command mode to insert mode. To move to the insert mode type i.

Nothing appears to change, but you are now in insert mode and can begin typing text. In general, vi's commands do not display on the screen and do not require the Return key to be pressed.

Type a few short lines and press <Return> at the end of each line. If you type a long line, you will notice the vi does not word wrap, it merely breaks the line unceremoniously at the edge of the screen.

Moving the Cursor

To move the cursor to another position, you must be in command mode. If you have just finished typing text, you are still in insert mode. Go back to

command mode by pressing <Esc>. If you are not sure which mode you are in, press <Esc> once or twice until you hear a beep. When you hear the beep, you are in command mode.

The cursor is controlled with four keys: h, j, k, l.

Key	Cursor Movement
---	-----

h	left one space
j	down one line
k	up one line
l	right one space

When you have gone as far as possible in one direction, the cursor stops moving and you hear a beep. For example, you cannot use l to move right and wrap around to the next line, you must use j to move down a line. See the section entitled "Moving Around in a File" for ways to move more quickly through a file.

Basic Editing

Editing commands require that you, to be in command mode. Many of the editing commands have a different function depending on whether they are typed as upper-case or lower-case. Often, editing commands can be preceded by a number to indicate a repetition of the command.

Deleting Characters

To delete a character from a file, move the cursor until it is on the incorrect letter, then type

x

The character under the cursor disappears. To remove four characters (the one under the cursor and the next three) type

4x

To delete the character before the cursor, type

X (uppercase)

Deleting Words

To delete a word, move the cursor to the first letter of the word, and type

dw

This command deletes the word and the space following it.

To delete three words type
3dw

Deleting Lines

To delete a whole line, type
dd

The cursor does not have to be at the beginning of the line. Typing dd deletes the entire line containing the cursor and places the cursor at the start of the next line. To delete two lines, type
2dd

To delete from the cursor position to the end of the line, type
D (uppercase)

Replacing Characters

To replace one character with another:

1. Move the cursor to the character to be replaced.
2. Type r
3. Type the replacement character.

The new character will appear, and you will still be in command mode.

Replacing Words

To replace one word with another, move to the start of the incorrect word and type

cw

The last letter of the word to be replaced will turn into a \$. You are now in insert mode and may type the replacement. The new text does not need to be the same length as the original. Press <Esc> to get back to command mode.

To replace three words, type

3cw

Replacing Lines

To change text from the cursor position to the end of the line:

1. Type C (uppercase).
2. Type the replacement text.
3. Press <Esc>.

Inserting Text

To insert text in a line:

1. Position the cursor where the new text should go.
2. Type i
3. Enter the new text.
The text is inserted BEFORE the cursor.
4. Press <Esc> to get back to command mode.

Appending Text

To add text to the right of the cursor:

1. Position the cursor on the last letter of the line.
2. Type a
3. Enter the new text.
This adds text AFTER the cursor.
4. Press <Esc> to get back to command mode.

To add text at the end of the line

1. Position the cursor on the last letter of the line.
2. Type A
3. Enter the new text.
4. Press <Esc> to get back to command mode.

Opening a Blank Line

To insert a blank line below the current line, type
o (lowercase)

To insert a blank line above the current line, type
O (uppercase)

Joining Lines

To join two lines together:

1. Put the cursor on the first line to be joined.
2. Type J

To join three lines together:

1. Put the cursor on the first line to be joined.
2. Type 3J

Undoing

To undo your most recent edit, type
u

To undo all the edits on a single line, type
U (uppercase)

Undoing all edits on a single line only works as long as the cursor stays on that line. Once you move the cursor off a line, you cannot use U to restore the line.

Moving Around in a File

There are shortcuts to move more quickly through a file. All these work in command mode.

Key	Movement
---	-----
w	forward word by word
b	backward word by word
\$	to end of line
0 (zero)	to beginning of line
H	to top line of screen
M	to middle line of screen
L	to last line of screen
G	to last line of file
1G	to first line of file
<Control>f	scroll forward one screen
<Control>b	scroll backward one screen
<Control>d	scroll down one-half screen
<Control>u	scroll up one-half screen

Moving by Searching

To move quickly by searching for text, while in command mode:

- Type / (slash).
- Enter the text to search for.
- Press <Return>.

The cursor moves to the first occurrence of that text.

To repeat the search in a forward direction, type

n

To repeat the search in a backward direction, type

N

Closing and Saving a File

With vi, you edit a copy of the file, rather than the original file. Changes are made to the original only when you save your edits.

To save the file and quit vi, type
ZZ

The vi editor is built on an earlier UNIX text editor called ex. ex commands can be used within vi. ex commands begin with a : (colon) and end with a <Return>. The command is displayed on the status line as you type. Some ex commands are useful when saving and closing files.

To save the edits you have made, but leave vi running and your file open:

- Press <Esc>.
- Type :w
- Press <Return>.

To quit vi, and discard any changes your have made since last saving:

- Press <Esc>.
- Type :q!
- Press <Return>.

1.3.8 Customizing vi editor environment

vi can be customized as per users' requirements using ex-mode commands.

Ex Commands	Action
:set all operative	Display all set options; those pre-fixed with no are not
:set autoindent(ai) lines	Extremely useful to programmers for indentation of
:set number (nu) preserved on saving file	Display all lines with numbers which are not
:set nonumber(nonu)	Reverses number setting
:set tabstop=6 (ts)	Changes default tab setting (8 spaces)
:set ignorecase(ic)	ignores case while pattern matching
:set showmatch (sm) input mode; correction	Helps locate matching brackets when) or } is entered in beeps when no match found to alert for
:set autowrite (aw)	Writes current file automatically whenever switching files with :n and escaping to shell with :sh
:next (n for short)	Moves to next file opened in vi
:rewind	Comes back to parent file

All sets can be stored in .exrc file used by vi for its startup instructions.

Equivalently an environment variable, EXINIT can be assigned to store the settings:

```
$EXINIT="set nu tabstop=6 ignorecase"
```

Assignment of these changes in the .profile will be available for all sessions.

1.4 Command Summary

STARTING vi

vi filename edit a file named "filename" if file already exists or it will create a new file.

vi newfile create a new file named "newfile"

ENTERING TEXT

i insert text left of cursor

a append text right of cursor

MOVING THE CURSOR

h left one space

j down one line

k up one line

l right one space

BASIC EDITING

x delete character

nx delete n characters

X delete character before cursor

dw delete word

ndw delete n words

dd delete line

ndd delete n lines

D delete characters from cursor to end of line

r replace character under cursor

cw replace a word

ncw replace n words

C change text from cursor to end of line

o insert blank line below cursor
(ready for insertion)

O insert blank line above cursor
(ready for insertion)

J join succeeding line to current cursor line

nJ join n succeeding lines to current cursor line

u undo last change
U restore current line

MOVING AROUND IN A FILE

w forward word by word
b backward word by word
\$ to end of line
0 (zero) to beginning of line
H to top line of screen
M to middle line of screen
L to last line of screen
G to last line of file
1G to first line of file
<Control>f scroll forward one screen
<Control>b scroll backward one screen
<Control>d scroll down one-half screen
<Control>u scroll up one-half screen
n repeat last search in same direction
N repeat last search in opposite direction

CLOSING AND SAVING A FILE

ZZ save file and then quit
:w save file
:q! discard changes and quit file

1.5 Video

<http://www.youtube.com/watch?v=njlQiSAbSA0>

ftp://172.26.48.31/ILP-Vivacious/C++/UNIX/Chapter6_V1/Vieditor V2.0

1.6 Quiz

Q 1. vi file1.txt will do what?

- a) open the file file1.txt
- b) create the file with file name file1.txt
- c) will open the file file1.txt if it exists , otherwise create it
- d) will open the editor in unix system.

Answer : c

Q 2. vi editor has the modes

- a) command and insert
- b) command, edit and insert
- c) command, insert and ex
- d) command and edit

Answer : c

Q3 . What does the character x do ?

- a) when in command mode , deletes the character beneath the cursor
- b) deletes the character beneath the cursor
- c) when in command mode , deletes the character next to the cursor
- d) deletes the character next to the cursor

Answer : a

Q4. In vi editor command to place the cursor at the beginning of line 4 is?

- a) 4g
- b) g4
- c) Go 4
- d) 4G

Answer : d

Q 5. To save a file in the vi editor and continue working, we have to type

- a) :w
- b) ;w
- c) :wg
- d) :s

Answer : a