

vi Editor

SYNOPSIS

This section provides a summary of the basic vi commands. Basic operations such as cursor movement, inserting and deleting text, and searches are covered to give the user a working knowledge of the vi editor. **vi** is an extensive editor so not all of its commands are covered.

SETTING TERMINAL TYPE

Before you can start *vi*, you must tell UNIX what kind of terminal you are using. On most systems, you will have already set your terminal type as part of your login process. However, in case you need it, the commands to tell UNIX the terminal type are:

```
% set term=terminal - type<CR>
```

```
% setenv TERM terminal - type<CR>
```

for C and T shells where "terminal - type" is the type of the terminal being used.

```
% set term=terminal - type; export term<CR>
```

for Bourne and Korn shells where "terminal - type" is the type of terminal being used.

Substitute the "terminal - type" of your terminal in the command above. If you are using communications software on a personal computer, determine which terminal type your software emulates. "vt100" is one of the more widely supported. Additional terminal types are listed in `/etc/termcap` file or `/usr/lib/terminfo`.

If you make a mistake in your command(s) for setting your terminal type, you may find that *pi co* will not allow you to proceed.

ENTERING VI

To enter *vi*, type:

```
% vi filename<CR>
```

If the file does not already exist, a new one will be created for you.

The vi editor has two modes of operation: edit mode and input mode. When you enter vi you are automatically put into edit mode. This means that you can issue commands to the editor such as scrolling through a file, moving the cursor, searching for a particular word, or deleting text. Most of these commands are not echoed on the screen.

When in input mode (see Entering Text) all text is echoed to the screen. When you want to exit input mode and return to edit mode, press the <ESC> key.

One potential area of confusion in vi is that there are no prompts to tell you which mode you are in (edit or input). It takes time to get used to the silent operation of the edit mode commands, and it is not always easy to remember to type <ESC> to get out of input mode. One clue is that if your "edit commands" are "echoed" to the screen you are in input mode.

SAVING THE FILE/EXITING VI

To save the file without exiting, type

`:w<CR>` saves updated file (note: the colon is part of the vi command)

`:w filename<CR>` Allows you to save the updated file under another name

To save the file and then exit type

`:wq<CR>`
or
`ZZ`

To exit vi without saving the file type

`:q!<CR>` Exits vi without saving the file.

ABOUT VI COMMANDS

vi commands are usually one letter mnemonics and may be combined with a count if the command is to be performed more than once.

Some vi commands require an operand as well (i.e., "d", for "delete" requires an operand such as "w" for "word" so that vi knows what it is you are deleting).

EXAMPLE

<code>4dw</code>	delete the next 4 words
<code>d4w</code>	delete the next 4 words
<code>5w</code>	move forward 5 words

ENTERING TEXT

Commands to enter text will put vi into input mode: everything typed in from that point will be interpreted as text. To exit input mode and return to command mode press the escape key `<ESC>`.

Here are the six basic insertion commands. Prefixes repeat the insertion.

<code>a</code>	Append text immediately following the cursor.
<code>A</code>	Append text at the end of the line.
<code>i</code>	Insert text immediately before the cursor.
<code>I</code>	Insert text at the beginning of the line.
<code>o</code>	Open a new line below the current one.
<code>O</code>	Open a new line above the current one.

Remember to press the ESCAPE key `<ESC>` to exit insert mode and return to command mode.

EXAMPLE

`aThis is input text<ESC>`

This will append the string "**This is input text**" after the cursor. Pressing the carriage return `<CR>` while in input mode makes room for (opens) a new line below the current one.

There is also a replace mode in which the characters typed overwrite any text at the cursor position.

`R` Enter replace mode.

To leave replace mode and return to command mode use the `<ESC>` key.

CURSOR CONTROL

There are quite a few commands to move the cursor within *vi* and **ALL** of them can be used as operands. Here are just a few to get you started.

Character positioning:

<i>l</i> or <i><SPACE BAR></i>	Move cursor to the next character.
<i>h</i> or <i><BS></i>	Move back a character.
<i>j</i>	Move down one line, same column.
<i>k</i>	Move up one line, same column.
<i>+</i> or <i><CR></i>	Move to beginning of next line.
<i>-</i>	Move to beginning of preceding line.

Word positioning

<i>w</i>	Move to the next word.
<i>b</i>	Move back a word.
<i>e</i>	Go to the end of a word.

DELETING TEXT

Text deletion is done in edit mode. Delete is a command that always requires an operand (see About Vi Commands). There are two basic deletion commands:

<i>x</i>	Delete a character.
<i>d{operand}</i>	Delete an object (Ex: " <i>w</i> " for "word").

EXAMPLES

<i>8x</i>	" <i>x</i> " deletes the current character, so this command deletes 8 characters.
<i>db</i>	" <i>b</i> " means "before" so this command deletes the previous word.
<i>8dd</i>	Deletes 8 lines (doubling the command affects lines).
<i>d<SPACE BAR></i>	Deletes the current character.

SCREEN CONTROL

The screen control commands manipulate the screen to view different parts of your file within the editor.

<i>Ctrl - F</i>	Forward a full screen.
<i>Ctrl - B</i>	Backward a full screen.
<i>Ctrl - D</i>	Scroll down a half screen.
<i>Ctrl - U</i>	Scroll up a half screen.

SEARCHING AND CHANGING TEXT

To search for a particular word or phrase in the file type a slash "*/*". Note that, unlike most *vi* commands, the slash is echoed at the bottom left-hand corner of the screen. After pressing the slash key, type in the pattern you want to search for followed by a carriage return *<CR>*. The editor will search down through your file to match the pattern; *vi* does have wrap-around capabilities. Even a pattern can be used as an operand.

EXAMPLES

<i>/pattern<CR></i>	Locate the next occurrence of " <i>pattern</i> "
<i>d/dog<CR></i>	Delete up to the next occurrence of the word " <i>dog</i> "

A "?" instead of a "/" will cause the editor to search backwards. The command "n" will repeat the last search.

EXAMPLE

<code>?pattern<CR></code>	Search back in file for pattern
<code>n</code>	Repeat last search

EX COMMANDS

Unix also has a line editor called *ex*, which is actually the line mode of *vi*. One may use any *ex* command from *vi* by using the `:` command. This is especially helpful for substitution commands. Note that the colon will be echoed to the screen.

EXAMPLE

```
:s/dog/cat/<CR>
```

This will substitute the next occurrence of the word *dog* with *cat* in the current line. To switch to *ex* at any time type *Q*, and type *vi* at the `:` prompt to return to *vi* mode.

ERROR RECOVERY

A. EDITING MISTAKES

The *undo* command, *u*, will undo only the previous command. Successive *undo*'s undo the undo.

B. SYSTEM CRASHES

If the system crashes, you can recover the work you were doing to within a few changes by the command

```
%vi -r filename<CR>
```

COMMAND SUMMARY

A. SETTING TERMINAL TYPE

```
setenv TERM terminal - type<CR> Set your terminal type first.
```

B. ENTERING VI

```
vi filename<CR> Start the vi editor.
```

C. SAVING THE FILE/EXITING VI

<code>:q<CR></code>	Quit the vi editor.
<code>:q!<CR></code>	Quit without saving changes.
<code>:w<CR></code>	Save changes.
<code>:w filename<CR></code>	Save the updated file under a different name.
<code>:wq<CR></code>	Quit, saving changes.
<code>ZZ</code>	Same as <code>:wq<CR></code> .

D. ENTERING TEXT

<code>a</code>	Append text immediately following the cursor.
<code>A</code>	Append text at the end of the line.
<code>i</code>	Insert text immediately before the cursor.
<code>I</code>	Insert text at the beginning of the line.
<code>o</code>	Open a new line below the current one.

O
R

Open a new line above the current one.
Enter replace mode.

E. CURSOR CONTROL

l or *<SPACE BAR>*

Move cursor to the next character.

h or *<BS>*

Move cursor back a character.

j

Move down one line, same column.

k

Move up one line, same column.

+ or *<CR>*

Move to the beginning of the next line.

-

Move to the beginning of the previous line.

w

Move to the next word.

b

Move to the back of a word.

e

Go to the end of a word.

DOCUMENT CODE: UNIX-30201C

DATE REVISED: July 2, 1997

Title: VL90.EPS from
Creator:
CreationDate: Fri Dec 17 10:24:25 1993

Figure vi-1: The Least You Should Know About vi.