Cutting text (d). See "Deleting text: (d)."

Copying text (y). When you yank text, vi copies it into the undo buffer, leaving the original file buffer text unchanged. Position the cursor on the first character of the text block you want to copy. Then press **y**, followed by a vi text string definition code. (See "Defining a string.") For example, to define to the end of the current word, press **w**. Vi copies the text from your file buffer into the undo buffer.

Pasting text (p). You can use the paste command to insert the contents of the undo buffer into your text at the cursor location. To use the paste command, first delete or yank the text you wish to paste (see above). Next, move the cursor to the place where you want to paste the text. Type **p**. Vi inserts the contents of the undo buffer into your file buffer at the cursor location.

Defining a string

The delete, yank, and change commands all follow the same general command syntax:

n command object

where *n* is an optional repeat count, *command* is the one-character command, and *object* is a code that defines the text to be affected. The object codes and their meanings include:

- **w** word
- **(** from cursor to start of sentence
- **)** from cursor to end of sentence
- **{** from cursor to start of paragraph
- **}** from cursor to end of paragraph
- ^ from cursor to first non-space character of line
- **$** from cursor to end of line

In addition:

- **dd** deletes the entire text line
- **yy** yanks the entire text line
- **cc** changes the entire text line

Here are some examples to help you learn vi editing command syntax:

- **dw** Delete word.
- **2dw** Delete two words.
- **3cw** Change three words.
- **y(** Yank from cursor to beginning of sentence.
- **c)** Change from cursor to end of sentence.
- **d{** Delete from cursor to beginning of paragraph.
- **y}** Yank from cursor to end of paragraph.
- **c^** Change from cursor to beginning of line.
- **d$** Delete from cursor to end of line.
- **5dd** Delete the next five lines.
- **3yw** Yank the next three words.
- **10X** Delete 10 characters to the left of the cursor.

Note: Vi considers a sentence to be a text string ending in a period, question mark, or exclamation point and followed by a new text line or two spaces. It considers a paragraph to be a series of text lines followed by a blank line.

Searching for a text string

Vi lets you define a text string and search your file buffer for it. Suppose you want to search the file buffer for every occurrence of the word "halyard."

In edit mode, press /. A cursor appears at the bottom of the screen. Now, type **halyard** and press ENTER. Vi searches the file for the first occurrence of "halyard," placing the cursor on the first character when it finds it.

You can now press **n** to advance to the next occurrence of "halyard," or enter a new search string. Vi search commands include:

- **/*x*** Search forward for *x*.
- **?*x*** Search backward for *x*.
- **n** Search forward for next occurrence of defined string.
- **N** Search backward for next occurrence of defined string.

The following special characters act as "wild cards," which allow you more flexibility in searching for a text string:

- **.** Match any character. For example, the search string ***m . d*** would match "mad," "mud," and "mod."
- * Match any string. For example, the search string ***hoo**** would match "hoosiers," "hoops," and "hook."

Search and replace

You can use vi to search your file buffer for a text string and replace each occurrence with new text. For example, to replace "halliard" with "halyard," in edit mode type:

:%/s/halyard/halliard/gc

Vi searches the file for "halyard." You'll see each occurrence in context. To replace the word and continue, press **y** and then ENTER, or to continue searching without altering the text, press **n** and then ENTER. (To stop a search in progress, press ESC.) When done, vi prompts you to press ENTER. Vi now displays the file buffer with the changes in place.

For more information

The online manual contains a vi summary. At the Unix shell prompt, type **man vi**.

You can check the UITS Knowledge Base for answers to the vi questions commonly asked at IU. Use your Web browser (such as Netscape or Internet Explorer) to open the URL:

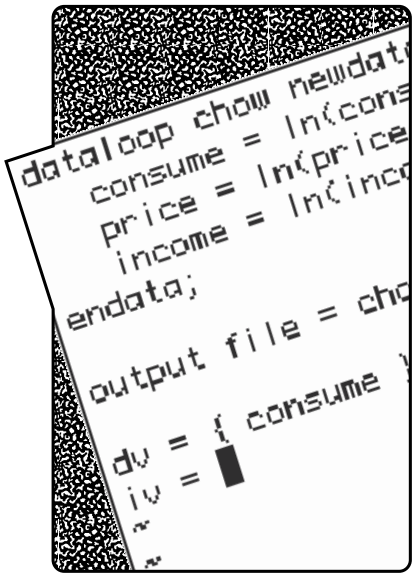
http://kb.indiana.edu/

Or, contact the UITS Support Center for help (IMU M084, 855-6789, Monday – Friday, 8am-5pm).

UC9712.b104

vi: A Unix text editor

This guide tells you how to use the vi text editor to create and edit text files on UITS Unix computers.



January 1998

For the most current version of this guide on the World Wide Web, set your browser to:
http://www.indiana.edu/~ucspubs/b104/

About vi

Full-screen text editors like vi, Emacs, and Pico enable you to create and edit text files, such as program code. The vi editor is available on all UITS Unix computers.

Before you run vi

Each time you connect to a central UITS computer to use vi, you need to set your terminal type. At IUB, your software will typically emulate a VT100 terminal. To find out what shell program you use, type:

```
echo $SHELL
```

Then, if you use ksh, bash, or sh, type:

```
export TERM = vt100
```

Or, if you use csh or tcsh, type:

```
setenv TERM vt100
```

You can automate this task by adding the appropriate command to your default command shell's configuration file, (e.g., .cshrc, .bashrc, etc.).

Invoking vi

You can invoke vi in different ways to suit your purposes.

Creating a new file. Suppose you want to create a new text file, *new.plans*, and then enter text into it. At your shell prompt, type: **vi new.plans**.

Vi detects that the *new.plans* file does not already exist. It creates a new empty file and opens a file buffer in which you can enter text. To indicate that the file is empty, vi displays tilde characters (~) down the left side of the file buffer.

Opening an existing file. Suppose you have already created and saved the file *old.plans.* Now you want to reopen and edit it. At your shell prompt, type:
vi old.plans.

Vi finds the file and displays the contents of the *old.plans* file for you to edit.

Using vi modes

Vi has three "modes": edit, insert, and colon.

Edit mode

Vi enters edit mode by default when it starts up. Edit mode allows you to move the cursor and edit the text buffer.

Insert mode

Insert mode "drops" the cursor at a specific point in the file, allowing you to insert text. To enter insert mode, position the cursor where you want to place text and press **i**.

Then type your text. If you make a mistake, press BACK-SPACE or DELETE to move the cursor backwards over the error. If the error is in a different line than the cursor, press ESC to return to edit mode, reposition the cursor at the error, and press **i** to get back to insert mode.

Here are some of the ways to enter insert mode:

i	Insert before cursor.	**I**	Insert at start of line.
a	Append after cursor.	**A**	Append at end of line.
o	Open new line below.	**O**	Open new line above.

Colon mode

Colon mode moves the cursor to the command line, allowing you to invoke program commands such as *write file* (**:w***)* and *quit* (**:q**). You enter colon mode from edit mode by typing a colon followed by a command. Some useful commands are:

:w	Write (save) current file.
:w *newname*	Write file to a new filename.
:r *oldname*	Read the file *oldname* into the current file.
:q!	Quit vi without saving changes to file.
:wq	Write and quit vi.
:e *filename*	Close current file and edit (open) *filename.*
:e #	Close current file and edit (open) previous file.

Quick quit (ZZ)

Want a fast way out? In edit mode, press SHIFT/**z** twice to save your file and exit vi.

Cursor movement

In addition to the arrow keys, you can use these edit mode commands to move the cursor:

h	Left one character.
l	Right one character.
j	Down one line.
k	Up one line.
b	Beginning of previous word.
e	End of next word.
w	Beginning of next word.
(	Backward one sentence.
)	Forward one sentence.
{	Backward one paragraph.
}	Forward one paragraph.
^	Go to first character of current line.
$	Go to last character of current line.
CTRL/**d**	Forward (down) one-half page.
CTRL/**u**	Backward (up) one-half page.
CTRL/**f**	Forward one page.
CTRL/**b**	Backward one page.
G	Go to the end of the file.
1G	Go to the first line of the file.
***n*G**	Go to line number *n*.

More edit mode commands

Removing characters: x, X

To remove the character under the cursor, press **x**. To remove the character to the left of the cursor, press **X**.

Restoring characters: u, U

To restore the last delete, press **u.** To undo all changes made to the current line, press **U**.

Replacing text: r, R

To replace the character under the cursor, press **r**, then type the replacement character.

To replace a portion of your current text line, position the cursor over the first character you want to change and press **R**. Vi is now in "strike-over" mode. It will allow you to type over the existing text to the end of the current text line. Press ESC at any time to return to edit mode.

Changing text: c

The change command lets you use the keyboard to mark a text block and replace it with new text.

Position the cursor on the first character of the text you want to replace. Type **c**, followed by a vi text string definition code. (See "Defining a string.") For example, to define to the end of the current word, press **w**.

Vi replaces the final character in the defined string with a dollar sign ($). You can now type the new text string. If your new text string has more characters than the original, vi will enter insert mode when it reaches the dollar sign. If your new text string has fewer characters than the original, vi will delete any remaining characters when you return to edit mode. To return to edit mode, press ESC.

Deleting text: d

The delete command lets you use the keyboard to define a text block and delete it from the file.

Position the cursor on the first character of the text you want to remove. Then press **d**, followed by a text string definition code. (See "Defining a string.") For example, to define to the end of the current word, press **w**.

Vi moves the text into the undo buffer. You can restore the deleted text at this point with the undo command (**u**). Or you can move the cursor and paste the contents of the undo buffer at a new location with the paste command (**p**). (See "Cut, copy, and paste functions.")

Cut, copy, and paste functions

Vi lets you delete (cut) or yank (copy) a section of text into the undo buffer. You can then reposition the cursor and paste the text from the undo buffer into the new location in your file buffer. The text you yank or delete remains in the undo buffer until you yank or delete again, overwriting the current contents.