

## Shell Escape Commands

|                       |                                                                                                                                                  |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>!: cmd</code>   | Execute shell command <i>cmd</i> ; you can add these special characters to indicate:<br>% name of the current file<br># name of last file edited |
| <code>!!</code>       | Execute last shell command                                                                                                                       |
| <code>:r! cmd</code>  | Read and insert output from <i>cmd</i>                                                                                                           |
| <code>:f file</code>  | Rename current file to <i>file</i>                                                                                                               |
| <code>:w !cmd</code>  | Send currently edited file to <i>cmd</i> as standard input and execute <i>cmd</i>                                                                |
| <code>:cd dir</code>  | Change the current working directory to <i>dir</i> (\$HOME is default)                                                                           |
| <code>:sh</code>      | Start a sub-shell (CTRL-d returns to editor)                                                                                                     |
| <code>:so file</code> | Read and execute commands in <i>file</i> ( <i>file</i> is a shell script)                                                                        |

## Shell Filters

|                              |                                                                                                                                              |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <code>!cursor_cmd cmd</code> | Send text from current position to <i>cursor_cmd</i> to shell command <i>cmd</i> . Replace original text in file with output from <i>cmd</i> |
| <code>!}sort</code>          | Example: Sort from current position to end of paragraph and replace text with sorted text                                                    |

## Macros and Abbreviations

|                              |                                                         |
|------------------------------|---------------------------------------------------------|
| <code>map key cmd_seq</code> | Define <i>key</i> to run <i>cmd_seq</i> when pressed    |
| <code>map</code>             | Display all created macros on status line               |
| <code>unmap key</code>       | Remove macro definition for <i>key</i>                  |
| <code>ab str string</code>   | When <i>str</i> is inserted, replace with <i>string</i> |
| <code>ab</code>              | Display all abbreviations                               |
| <code>una str</code>         | Unabbreviate <i>str</i>                                 |

`ap` allows you to define strings of `vi` commands. Place in `exrc` to run each time you enter `vi`. For long macros, set the `otimeout` option. If you embed control characters (e.g., keys `a` [ESC] in the macro, you need to precede them with [CTRL-v], you need to include quotes ("), precede them with \ (backslash). Used keys in `vi` are: `K V g q v * =` and the function keys.

Example:

```
:map v /I [CTRL-v] [ESC] dwIYou [CTRL-v] [ESC] [ESC]
When v is pressed, search for "I" (/I [ESC]), delete word (dw) and
insert "You" (IYou [ESC]). [CTRL-v] allows [ESC] to be inserted.
```

## Setting Options

Options shown here are default. To change them, either set them (:set option) or unset them (:set nooption). To run options each time you enter `vi`, place in `.exrc` file in home directory and omit preceding colons (:).

|                                 |                                                                              |
|---------------------------------|------------------------------------------------------------------------------|
| <code>:set all</code>           | Print all options                                                            |
| <code>:set nooption</code>      | Turn off <i>option</i>                                                       |
| <code>:set noai</code>          | Set automatic indentation                                                    |
| <code>:set ap</code>            | Print line after <code>d c J M : a t u</code> command                        |
| <code>:set bf</code>            | Discard control characters from input                                        |
| <code>:set eb</code>            | Precede error messages with bell                                             |
| <code>:set noic</code>          | Ignore case when searching                                                   |
| <code>:set dir=tmp</code>       | Set directory of buffer file                                                 |
| <code>:set lisp</code>          | Modify brackets for Lisp compatibility                                       |
| <code>:set magic</code>         | Pattern match with special characters                                        |
| <code>:set mesg</code>          | Allow other users to send messages                                           |
| <code>:set nolist</code>        | Show tabs (^I) and end of line (\$)                                          |
| <code>:set nonu</code>          | Prefix lines with line number                                                |
| <code>:set opt</code>           | Speed output: eliminate automatic [Return]                                   |
| <code>:set prompt</code>        | Prompt for command mode input with :                                         |
| <code>:set nore</code>          | Simulate smart terminal on dumb terminal                                     |
| <code>:set remap</code>         | Allow macros within macros                                                   |
| <code>:set noreport</code>      | Indicate largest size of changes reported on status line                     |
| <code>:set ro</code>            | Change file to read only                                                     |
| <code>:set scroll=n</code>      | Set <i>n</i> lines for [CTRL-d] and <code>z</code>                           |
| <code>:set sh=shell_path</code> | Set shell escape (default is /bin/sh)                                        |
| <code>:set showmode</code>      | Indicate input or replace mode                                               |
| <code>:set sw=8</code>          | Set the shift width                                                          |
| <code>:set term</code>          | Print terminal type                                                          |
| <code>:set noterse</code>       | Shorten error messages with <code>terae</code>                               |
| <code>:set notimeout</code>     | Eliminate one second time limit for macros                                   |
| <code>:set t1=0</code>          | Set significance of tags beyond this many characters (0 means all)           |
| <code>:set ts=8</code>          | Set tab stops for text input                                                 |
| <code>:set nowa</code>          | Inhibit normal checks before write commands                                  |
| <code>:set warn</code>          | Warn "No write since last change"                                            |
| <code>:set window=n</code>      | Set number of lines in a text window                                         |
| <code>:set wm=n</code>          | Set automatic wraparound <i>n</i> spaces from right margin (e.g., :set wm=8) |

# Quick Reference for vi

This quick reference lists commands you can use in the `vi` editor on Hewlett-Packard's UNIX™ System, HP-UX.

## Notations

- Commands beginning with `:` (colon) must end with [Return]. These commands represent escape commands to the `ex` editor. You can reference the `ex` tutorial for more details.
- file* is the name of file
- cursor\_cmd* is a cursor movement command (e.g., `G j w b`)
- char* is a single character
- str* is a character string (can contain pattern matching characters)
- [CTRL-x] means you press [CTRL], hold it down, and press the [x] key.
- n,m* can be two line numbers (e.g., 4,50), line marker (e.g., `.,$`), or search expression (e.g., `/string1/. /string2/`).
- (*a-z*) means you choose a letter from *a* through *z*

## Modes

**Command Mode** When you are not inserting or changing text, you can move the cursor and run commands (e.g., searching, deleting, saving). Pay attention to the case of the commands; check the [Caps] lock key if `vi` behaves strangely.

**Insert Mode** When you insert or change more than one character of text, you cannot use command mode commands. To leave the insert mode, press [ESC].

## Searching

% Search to beginning of balancing ( ) [ ] or { }  
 fchar Search forward in current line to char  
 Fchar Search backward in current line to char  
 tchar Search forward in current line to character before char  
 Tchar Search backward in current line to character after char  
 /str Return Find str  
 ?str Return Search in reverse for str  
 : set !c Ignore case when searching  
 : set !o!c Pay attention to case when searching (default)

## Global Search and Replace

: n,ms/str1/str2/opt Search from n to m for str1. Replace str1 with str2, using opt. opt can be g for global change, c to confirm change (press y to acknowledge, Return to suppress), and p to print changed lines.  
 & Repeat last : s command

g/str/cmd Run cmd on all lines that contain str  
 g/str1/a/str2/str3/ Find line containing str1, replace str2 with str3  
 v/str/cmd Execute cmd on all lines that do not match str

## Copying and Placing Text

ny or nY Yank n lines (place in buffer); omitting n yanks current line  
 ycursor\_cmd Yank from cursor to cursor\_cmd (e.g., yg yanks current line to last line in file)  
 "(a-z)ny or "(a-z)ndd Copy or delete n lines into named buffer a through z; omit n for current line  
 p (lower-case) Put yanked text after cursor (print buffer); also prints last deleted text  
 P Put yanked text before cursor; also prints last deleted text  
 "(a-z)p or "(a-z)P Put lines from named buffer a through z after or before current line

## Changing Text

Preceding these commands with n (a number) repeats the command n times.  
 rchar Replace current character with char  
 Rtext ESC Replace current character(s) with text  
 atext ESC Substitute text for current character  
 S or cc text ESC Substitute text for entire line  
 cwtext ESC Change current word to text  
 Ctext ESC Change rest of current line to text  
 ccursor\_cmd text ESC Change to text from current position to cursor\_cmd

## Joining Lines

J Join next line to end of current line  
 nJ Join next n lines

## Cursor Placement and Adjusting the Screen

H Move cursor to top line of screen  
 nH Move cursor to line n from top of screen  
 M Move cursor to middle of screen  
 L Move cursor to bottom line of screen  
 nL Move cursor to Line n from bottom of screen

CTRL-e

Move screen up one line

CTRL-y

Move screen down one line

CTRL-u

Move screen up 1/2 page

CTRL-d

Move screen down 1/2 page

CTRL-b

Move screen up one page

CTRL-f

Move screen down one page

CTRL-l (letter "l")

Redraw screen

z Return

Make current line top line on screen

nz Return

Make line n top line on screen

z.

Make current line middle line

nz.

Make line n middle line on screen

z-

Make current line bottom line

nz-

Make line n bottom line on screen