

BUDVISOR

BUDdy for VI editor's Seemingly OutRageous commands

VER 1.1
Copyright © 1993 Venkman Inc.

by
Venkman: Venkman1@ix.netcom.com

NOTE: In commands where word "move" or "movesrch" is mentioned the move commands or search commands can be combined to define the text to be operated on. e.g. y3w yanks 3 words, dG deletes till end of file, dfe deletes till it finds an 'e' in the line (e inclusive), etc. The # before the keyword move implies that a number may be given.

MOVES

<p>^D DOWN one half-screen ^U UP one half-screen ^F FORWARD one full-screen ^B BACKWARD one full-screen ^E Move window down one line without moving cursor ^Y Move the window up one line without moving cursor ((left paren) to the beginning of a sentence) (right paren) to the beginning of the next sentence [[to the beginning of a section]] to the end of a section</p>	<p>\$ to the end of the line ^ to the first non-white character on the line 0 to the first column on the line (column zero) # to an exact column on the line (column #) e.g. 5 1 G GO to the last line in the file #G GO to line #. (e.g., 3G, 5G, 175G) { Move to the beginning of a paragraph } Move to the end of a paragraph H to the HIGHEST position in the window M to the MIDDLE position in the window L to the LOWEST position in the window</p>	<p>h left one character j down one line k up one line l right one character w beginning of the next WORD (word is de-limited by non-al.num chr) e END of the next word b BACK to beginning of previous word W,E,B ditto, (word delimited by space) + to first non-white char of next line - to first non-white char of previous line line #move repeat move # times</p>
--	--	--

EDITS

<p>EXITING: :q! QUIT without saving changes ZZ Exit and save any changes :w (filename) WRITE without quitting (to "filename" if given) :efilename Begin EDITing the file called "filename" :x, :wq WRITE and QUIT :r filename READ file called "filename" UNDO: u UNDO last change U UNDO all changes made to the current line as long as your cursor has not moved off the line</p>	<p>CUTTING: dd DELETE one line dmovesrch DEL to where "move" or "search" specifies (e.g. dfe dta d20) d#move DELETE to where the #move specifies d/pattern delete till pattern is found (forward) d?pattern delete till pattern is found (backward) D delete till end of line x X-OUT one character under cursor X X-OUT char before the cursor</p>	<p>TEXT: ESC ESCAPE from insert mode . (dot) repeat last change o OPEN a line below the cursor O OPEN a line above the cursor i INSERT starting before the cursor I INSERT at the beginning of the line a APPEND starting after the cursor A APPEND at the end of the line J JOIN two lines #s SUBSTITUTE for # characters #S SUBSTITUTE for # whole lines</p>
<p>YANKING: ymovesrch YANK to where "move" or "search" specifies y#move YANK to where the #move specifies Y or yy YANK one line #Y, #yy YANK # lines below</p>	<p>PASTING: P (upper p) PASTE buffer before the cursor p (lower p) PASTE buffer after the cursor</p>	<p>r REPLACE character (NO need to press ESC) R enter over-type mode c(move) CHANGE to where the 'move' specifies C CHANGE till end of line</p>

SEARCHES

<p>f char FIND next occurrence of char on the line t char Move 'TIL next occurrence of char on the line F char FIND previous occurrence of char on the line T char Move 'TIL previous occurrence of char on the line ; Repeat the last f, t, F, or T , Reverse the last f, t, F, or T % Show matching () or {} or [] n Repeat last / or ? command N Reverse last / or ? command /string Find string looking forward ?string Find string looking backward / or ? Use string used for previous search</p>
--

: COMMANDS

<p>:#,<command>(!)(filenm) General fmt. One of the #s can be : +# #below -# #above . current ln. \$ last ln. :s/<search>/<replace>/(gc) SUBSTITUTE <search> with <replace> once in a line if (g) specified then replace all within line. 'c' is for confirmation :g/<search>/s//<replace>/(gc) SUBSTITUTE globally first occurrence of <search> with <replace> if (g) then all occurrences within line are replaced. 'c' is for confirm. :# go to LINE # !:<UNIX> run UNIX command outside the vi :d delete lines btn. line numbers & Do last search-replace again % same as !,\$</p>
--

NOTE: PATTERNS USED WITH / & ? SEARCHES: ^ beginning of line, \$ end of line, . any char, * any # of chars, \< beg of word, \> end of word, [str] any chr in str, [^st].. not in st, [x-y] btn x & y

SPECIALS

<p>m char MARK this location and name it char ' char (quote character) return to line named char '' or '' (quote quote) return from last movement "(a-z)(1-9)DEL DELETE, YANK or PASTE text from buffers 'a' thru 'z' or '1' thru '9' (one at a time only) "(a-z)(1-9)YANK "(a-z)(1-9)PASTE (e.g. "ad" "5dw, "by3y "2Y, "dp "5P) ~ (tilde) Convert case of current character z<RETURN> Position the current line to top of window</p>	<p>z. Position the current line to middle of window z- Position the current line to bottom of window ^G Show status of current file ^L Refresh screen ^R retype, clear @s >movement Shift right to where the movement command specifies <movement Shift left to where the movement command specifies !} run a UNIX command on the paragraph (e.g. !}sort sorts lines of a paragraph)</p>
--	---