

## QUICK REFERENCE



ERROR AND OTHER MESSAGES

VEDIT PLUS displays a message to notify you of errors or special conditions. The messages fall into three categories: disk errors, command errors and other messages. Disk errors are caused by conditions described below. Command errors usually are caused by syntax errors or overlooking some small detail. Most of the remaining messages are confirmation prompts to perform an operation which may be hard to undo.

See also "Status Line Messages" for messages which appear on the status line.

MESSAGES

ABANDON FILE (Y/N)?

This is the normal confirmation prompt for the "EQ" and "EZ" commands, or when quitting from the [FILE] function. Type "Y" or "y" if you really want to abandon the file, without saving any changes; type any other character to cancel the command. To skip this prompt when quitting, enter "EQY" or "EZY" instead of "EQ" or "EZ".

ABANDON ALL FILES (Y/N)?

This is the confirmation prompt for the "EQA" command to quit/abandon ALL edit buffers and exit VEDIT PLUS.

BAD FILENAME (Command Error)

The filename you gave does not follow the DOS conventions.

BAD PARAMETER (Command Error)

Something was specified wrong with your "EI", "EP", "ES", "PP" or "ET" command.

\*BREAK\* (Command Error)

Command execution stopped because insufficient memory space was free to perform the command. For the "I", "S" and "EG" commands, as much text as possible was inserted. For the "RC" and "RG" commands, no text at all was copied or inserted. The message is also displayed when you press <CTRL-C> in Command Mode or while printing.

CANCEL, IGNORE, RETRY (C/I/R)? (MS-DOS / PCDOS only)

This error prompt occurs when DOS detects a disk error (disk drive door open, read/write error) or a printer error. Type "C" to cancel the operation. This is safe because it returns to VEDIT PLUS. Type "R" to re-try the operation, such as after closing the drive door. You can type "I" to ignore, but this usually just leads to another error.

CANNOT FIND... (Command Error)

The specified search string could not be found. This is the normal return for iteration loops which search for all occurrences of a string. Use the ":" command modifier to suppress this error message.

CANNOT OPEN TWO FILES (Command Error)

You cannot open two files in one edit buffer - one file is already open due to invoking VEDIT PLUS or a "EW" or "EB" command. This message also occurs if an output file is open at the time of an "EC" command. Perhaps you want to save the current file with the "EY" command or switch to another edit buffer to simultaneously edit another file.

CLOSE ERROR (Disk Error)

The output file could not be closed. This is a very unusual condition, but may occur if the disk becomes write protected, or if you changed disks without first issuing an "EC" command.

DIRECTORY NOT FOUND (Command Error)

The MS-DOS subdirectory specified as part of a filename could not be found. Perhaps you mistyped it or specified the wrong drive.

DISK IN USE BY MP/M (Disk Error)

You cannot perform an "EC" command because the disk is being used by another program or user. See section on Multi-Tasking Operating Systems.

END OF EDIT BUFFER REACHED (Command Error)

An "L" command tried to move past either end of the edit buffer. Use "n:L" to suppress this error message.

END OF MACRO REACHED, STRING STILL OPEN (Command Error)

The end of a command macro was reached before the last text string delimiter was found. Usually caused by forgetting the final <ESC> or explicit delimiter on a text string.

END OF MACRO REACHED, STRUCTURE STILL OPEN (Command Error)

The end of a command macro was reached before the final "]" to end a flow control structure was found. Usually caused by not properly matching up "[" and "]" in complex nested flow control structures. It can also be caused by unpaired "[" and "]" occurring within text strings.

FILE IS R/O - OUTPUT FILE NOT OPENED (Disk Error)

The file you are trying to edit or write already exists and has a "Read Only" attribute. Your command is canceled. You will have to return to the operating system and change its attribute before you can edit it.

FILE NOT FOUND (Command Error)

The file you wanted to open for input does not exist. Maybe you specified the wrong drive.

FILE NOT OPENED (Disk Error)

This message follows another message and reminds you that your attempted disk operation was canceled. It follows the "DISK IS R/O" message. Also follows an operating system error message if you attempted to open a file which is in use by another program or user. See section on Multi-Tasking Operating Systems.

INSERT NEW DISK AND PRESS <RETURN>

This is the normal prompt for inserting a new disk with the "EC" command.

INVALID COMMAND (Command Error)

The specified letter(s) is not a command.

INVALID DELIMITER FOR JP COMMAND (Command Error)

You cannot use the "!" character as an explicit delimiter with the "JP" command. In general, the "!" character should never be used as an explicit delimiter.

## INVALID DRIVE (Command Error)

The drive you specified as part of a filename was either specified incorrectly or does not exist on your computer. Most likely you mistyped it.

## INVALID EDIT BUFFER OPERATION (Command Error)

You are attempting an operation which is valid for normal text registers, but is invalid for edit buffers. You cannot change the contents of an edit buffer except when it is the active edit buffer.

## INVALID [HELP] REQUEST (Command Error)

During the [HELP] function you are prompted to press the function/control key for the desired function. Instead you pressed a displayable character or an unused function/control key.

## INVALID NUMERIC REGISTER (Command Error)

You specified an invalid numeric register with a numeric command. The numeric register must be in the range "0" - "99" and must immediately follow the two letter numeric command. (The first letter of numeric commands is "X"). (CP/M versions only support 26 numeric registers "0" - "25".)

## INVALID SEQUENCE (Command Error)

While setting up a keystroke macro you entered the "New Function/Control Key" as an invalid "sequence" beginning with a displayable character. You may use a function/control key, a single displayable character or a "sequence" beginning with a control character. Also displayed when you cancel [DEFINE] by pressing [BACKSPACE].

## JUMPING INTO A FLOW CONTROL STRUCTURE (Command Error)

You cannot jump into a flow control structure with the "JP" command. Flow control structures can only be entered at their beginning. However, it is allowable to jump out of a flow control structure, or to jump within it.

## LABEL MISSING !label! (Command Error)

The "JP" Jump command could not find the 'label' it is to jump to.

## LABEL TERMINATOR "!" MISSING (Command Error)

The label for the "JP" Jump command must be enclosed in two "!" characters. This error occurs when either the first or second "!" is missing.

## MACRO ERROR IN r (Command Error)

The command macro in register 'r' attempted to modify itself or another text register which is currently executing (the "calling" macro). Usually caused by naming the wrong register with the "RC" command.

## NESTING ERROR (Command Error)

You cannot nest macros deeper than 20 levels. This error usually is caused by a macro which recursively calls itself. For example, register 'A' contains the command "MB" and register 'B' contains the command "MA".

## NEW FILE

The file specified with the "EB" command or with the invocation of VEDIT PLUS did not exist on disk and a new file has been created. If you typed the wrong filename, you may want to start over by issuing the "EZ" command.

## NO DIRECTORY SPACE (Disk Error)

There was no directory space left for the output file. Refer to "Disk Write Error Recovery".

## NO DISK SPACE (Disk Error)

The disk became full before the entire output file was written. As much of the output file as possible was written. Refer to "Disk Write Error Recovery".

## NO INPUT FILE (Command Error)

There is no input file open for doing a read or append operation.

## NO MEMORY TO AUTO BUFFER (Command Error)

There is insufficient memory available for the edit buffer to perform auto-buffering. Use the "U" command to see how much memory is available. Most likely, almost the entire memory is being used by the text registers, leaving less than 3 Kbytes for the edit buffer. You must empty a text register or issue your own read/write commands.

## NO OUTPUT FILE (Command Error)

There is no output file open for doing a write, a close or an exit with the "EX" command. If you have already written out the text buffer and closed the output file, exit with the "EQ" command. You must have an output file open when performing global file operations.

## NOT FOUND IN HELP FILE (Command Error)

The help topic you selected could not be found in the help file. Either you entered the topic name incorrectly, or the help file is not complete.

## OK TO CLOSE (ERASE) OUTPUT FILE (Y/N)?

This is the confirmation prompt for the "EF" command which closes the current output file. Type "Y" or "y" to close the file. (The word "erase" is to remind you that improper use of the "EF" command can erase the file being edited!) Use the command "EFY" to skip this prompt.

## OK TO DELETE nnnnn LINES (Y/N)?

This is the confirmation prompt for the [BLOCK]-Delete function. 'nnnnn' is the number of lines which are about to be deleted. Type "Y" or "y" to delete the lines; type any other character to cancel the operation.

## OK TO ERASE THESE FILES (Y/N)?

This is the normal confirmation prompt with the "EK" command after displaying the filenames which are about to be erased. Type "Y" or "y" if you really want ALL of the listed files erased; type any other character to cancel the "EK" command.

## OK TO OVERWRITE EXISTING FILE (Y/N)?

This prompt appears with the "RS" command if the designated file to be written already exists on disk. Type "Y" or "y" to overwrite the existing disk; type any other character to cancel the "RS" command. Note that the "RS" command does NOT create a "backup" of any existing file.

## PRINTING -- Press &lt;CTRL-C&gt; to Abort

Normal message anytime text is being printed. Reminds you that you can press <CTRL-C> to stop the printing.



## READ ERROR

(Disk Error)

An error occurred reading a file. This error should never occur, since the operating system itself normally gives an error if there was a problem reading the disk.

## REDEFINE DISPLAYABLE CHAR (Y/N)?

The "Function/Control Key" you entered while setting up a keystroke macro with [DEFINE] is a displayable character which is about to be assigned a new meaning. Type "Y" to redefine it, or "N" if you made a mistake and do not want to redefine it.

## REDEFINE EXISTING KEY (Y/N)?

The "Function/Control Key" you entered while setting up a keystroke macro is already assigned to an edit function or a keystroke macro. Type "Y" to redefine it, or "N" if you made a mistake and do not want to redefine it.

## REV FILE OPEN

(Command Error)

You cannot change disks with the "EC" command because the ".\$RS\$" file is open due to backward disk buffering.

## UNABLE TO ENTER VISUAL MODE

(Command Error)

The "V" command noticed that the keyboard decode table is invalid. The most likely cause is loading the wrong file with the "YL" command. Either load the correct file or give the command "EXA" to exit VEDIT PLUS.

## UNKNOWN INTERNAL VALUE

(Command Error)

The internal value designator "." was encountered, but the following characters are not a defined internal value.

## WAITING FOR PRINTER -- Press &lt;CTRL-C&gt; to Abort

VEDIT PLUS is waiting for another program or user to release the printer before it can begin printing. Message changes to normal printing message once printing begins. See section on Multi-Tasking Operating Systems.

- APPEND** This message follows the text register prompt after you type "+" to select that you want the text appended to any existing text in the register.
- C-N-S** On the IBM PC, the status of the Caps/Num/Scroll Lock is displayed on the status line with the single letters "C", "N" and "S".
- COL** Indicates in which column of the text the cursor is.
- COMMAND** Indicates that you are in Command Mode. Only the filename and edit buffer name are displayed on the status line in Command Mode.
- E#** The "E#" is followed by the name of the active edit buffer. It does not appear when the main edit buffer is active.
- FILE** Indicates the name of the file you are currently editing. Technically it is the name of the "output file" - the name that the file will have when you save it on disk. If the message is all in capital letters the entire file is currently in memory. Otherwise the message is displayed as follows:
- File** The beginning of the file is currently in memory, but the end of the file is on disk.
- FILE** The end of the file is currently in memory, but the beginning of the file has been written to disk.
- file** Neither the very beginning nor end of the file is currently in memory - they are both on disk.
- FULL** Indicates that the edit buffer has run out of memory space -OR- that there was insufficient memory for the text register operation you just attempted. The former should rarely happen unless "auto-buffering" is disabled.
- INSERT** Indicates that you are in "Insert" mode.

HORIZONTAL SCROLL OFFSET

When the screen is scrolled sideways the column number corresponding to the left side of the screen is displayed on the left side of the status line.

**LINE** Indicates on what line of the file the cursor is. Note that the line number for a particular line of text will decrease if some of the preceding lines are deleted, and will increase if lines are inserted into the preceding text.

MENU PROMPTS

Menu selections are made by typing the letter corresponding to the desired operation - do not follow the letter with a **RETURN**. However, prompts for numbers have to be followed with a **RETURN**.

**MORE** Indicates that at least one text register is also an edit buffer. Note: the **TEXT** message has higher priority and usually appears in its place.

**PLEASE WAIT FOR DISK**

Indicates that disk reading or writing is being performed. Although the IBM PC has no problems, other computers may not reliably accept keyboard input during disk operations.

REPEAT COUNT

The currently selected repeat count when using the **[REPEAT]** function is displayed on the far left of the status line.

**TEXT** Indicates that you have text in at least one text register, (which is not also an edit buffer). If you forget, use the "RU" command to see which registers contain text or are edit buffers. (It will be overwritten by the **1 END** message.)

**1 END** Some edit functions require that you mark the beginning and end of block of text. This message indicates that you have marked one end of the block (you can mark either the beginning or the end of the block first). Pressing **[CANCEL]** will reset the first marker and remove this message.

[CURSOR UP]	Moves the cursor up one line, to the same horizontal position if possible. If the position is in the middle of a tab, moves to the end of the tab. The "Cursor Positioning Mode" determines whether cursor can be positioned past shorter lines.
[CURSOR DOWN]	Moves the cursor down one line, to the same horizontal position if possible. The same rules as for [CURSOR UP] apply.
[CURSOR RIGHT]	Moves the cursor to the next character. At the end of the line, it moves to the beginning of the next line.
[CURSOR LEFT]	Moves the cursor to the previous character. At the beginning of a line, it moves to the end of the previous line.
[BACK TAB]	Moves the cursor to the first character of the screen line or, if already there, to the first character of the previous screen line.
[TAB CURSOR]	Moves the cursor to the character at the next tab position. Note that this only moves the cursor; use the <TAB> key to insert a tab character.
[ZIP]	Moves the cursor to the end of the current screen line or, if already there, to the end of the next text line.
[LINE TOGGLE]	Is a combination of [ZIP] and [BACK TAB]. Moves the cursor to the end of the line or, if already there, to the beginning of the line.
[NEXT LINE]	Moves the cursor to the beginning of next text line.
[SCROLL UP]	Similar to [CURSOR UP], except that the cursor remains on the same screen line and the screen moves down instead.
[SCROLL DOWN]	Similar to [CURSOR DOWN], except that the cursor remains on the same screen line and the screen moves up instead.
[SCROLL RIGHT]	Moves the screen window right to view long lines going off the right side of the screen.
[SCROLL LEFT]	Moves the screen window left to view the beginning part of long lines.

[PREVIOUS WORD]	Moves the cursor to the first character of the current word or, if already there, to the beginning of the previous word.
[NEXT WORD]	Moves the cursor to the first character of next word.
[PREVIOUS PARA]	Moves the cursor to the beginning of the current paragraph or, if already there, to the beginning of the previous paragraph.
[NEXT PARA]	Moves the cursor to the beginning of next paragraph.
[PAGE UP]	Moves to previous "page" of text - similar to typing [CURSOR UP] for 3/4 screen lines.
[PAGE DOWN]	Moves to next "page" of text - similar to typing [CURSOR DOWN] for 3/4 screen lines.
[SCREEN TOGGLE]	Moves the cursor first to the last allowed screen line, or if already there, to the first allowed screen line.
[INSERT]	Toggles the "Insert"/"Overstrike" mode to its opposite.
[DELETE]	Deletes the character at the cursor position. The cursor doesn't move. A lone <CR> or <LF> will also be deleted, but a <CR> <LF> pair will both be deleted as one.
[BACKSPACE]	Moves the cursor left and deletes the character at that position.
[DEL PREVIOUS WORD]	Deletes the previous word and any following spaces if the cursor is at the beginning of a word. Otherwise deletes only that portion of the word to the left of the cursor.
[DEL NEXT WORD]	Deletes the entire word and any following spaces if the cursor is at the beginning of a word. Otherwise deletes from the cursor position to the end of the word.

- [ERASE EOL] Deletes (erases) all characters from the cursor position to the end of the text line. If the text line is completely blank, it deletes the blank line. For example, the following sequence will delete an entire line:
- [BACK TAB] [ERASE EOL] [ERASE EOL].
- [ERASE LINE] Deletes (erases) the entire text line. This also closes up the screen line.
- [UNDO] Rewrites the screen and ignores the changes made to the text line the cursor is on.
- [TAB CHARACTER] Enters the tab character (<CTRL-I>) into the text. Optionally expands with space characters to the next tab position. Almost always assigned to the <TAB> key.
- [NEXT CHAR LITERAL] Enter the next character, whether a displayable character, a control character or an 8 Bit character, into the edit buffer.
- [REPEAT] The next typed character or edit function is repeated. This is either a multiple of four or the number typed in. Type "000" and any character to abort; or press [CANCEL] to abort.
- [FIND] Performs a search operation. Prompts for the search string - enter the search string of up to 80 characters and a RETURN. Use <CTRL-N> to search for RETURN. Press [FIND] again to search for the next occurrence.
- [REPLACE] Performs a selective replace. Prompts for string to be replaced, then prompts for replacement string. Each time [REPLACE] is again pressed, the next occurrence of the string is found and the user has the option of replacing it or not, and of replacing all further occurrences of the string or canceling the assignment.
- [CANCEL] [FIND] and [REPLACE] reuse the previous strings until they are canceled with [CANCEL] or the search string is not found. [CANCEL] will also abort any function taking place because of a [REPEAT] and cancel the "1 END" block marker. On the IBM PC (and many other machines) it also cancels any status line prompts.

- [INDENT] Increases the "Indent Position" by the amount of the "Indent Increment". Each new line is then automatically padded with tabs and spaces to the Indent position. The padding will also take place on the current line if the cursor is before any text on the line.
- [UNDENT] Decreases the "Indent Position" by the "Indent Increment", until it is zero. One [UNDENT] therefore cancels one [INDENT].
- [FORMAT PARAGRAPH] Formats the current paragraph so that all text appears between the left and right margins. The left margin is the current Indent Position, and the right margin is the current Word Wrap margin. At completion, the cursor is positioned at the beginning of the next paragraph. Print Formatter commands will not be formatted. The paragraph will also be justified if the justify option is set from the [USER] menu.
- [MACRO] Executes a command macro. Prompts for the name of the text register which contains the macro to be executed. If during the execution of the macro anything is displayed on the screen, you will be prompted to "Press any key to continue". This lets you see the displayed information before returning to Visual Mode.
- [VISUAL EXIT] Visual Mode is exited to Command Mode. The current cursor position in the edit buffer becomes the Command Mode edit pointer position.
- [VISUAL ESCAPE] This is identical to the [VISUAL EXIT], except that any current command execution is aborted.
- [DEFINE] Prompts for the strings necessary to set up "keystroke macros". First prompts for "Function/control key" - this is usually an unused function (or <ALT>) key to which a sequence of key strokes are then assigned. Next prompts for the sequence of key strokes. Pressing [BACKSPACE] at any time aborts the operation. Therefore, [BACKSPACE] cannot be part of a keystroke macro. However, you can perform the equivalent function with [CURSOR LEFT] and [DELETE]. Keystroke macros can be built-in with Task 3 of the Installation.

[HELP]	On-line help for Visual Mode using the file "VVHELP.HLP". The keyboard layout is first displayed in two menu screens. Press RETURN at the first menu prompt to see the second screen. For help on a particular edit function press the function/control key for that function. Alternatively, for the menu of Visual Mode topics, press RETURN again at the second menu prompt. Then type the name of the topic for which you want help.
[GOTO]	<b>[B]egin [H]ome [M]ark [J]ump [L]ine [Z]end [E]nd</b>
[B]egin	Moves cursor to first character in the file.
[H]ome	Moves cursor to first character in edit buffer.
[S]et	Prompts for a digit "0 - 9". Sets an invisible text marker to which you can later "jump" back.
[J]ump	Prompts for a digit "0 - 9". "Jumps" to a previously set text marker. If the marker has not been set or has been reset, it is ignored.
[L]ine	Prompts for a line number. Moves the cursor to the beginning of the specified line.
[Z]end	Moves cursor past last character in edit buffer.
[E]nd	Moves cursor past last character in file.
[FILE]	<b>[E]xit [Q]uit [N]ew [S]ave [D]irectory</b>
[E]xit	Exits VEDIT PLUS (window) and saves file. This is the normal way to leave VEDIT PLUS.
[Q]uit	Quits VEDIT PLUS (window) and abandons file - any edit changes to the file ARE NOT SAVED.
[N]ew	Sets up to edit a new file. Current file may be saved to disk or abandoned.
[S]ave	Saves current file to disk for continued editing. Protection against power failure or major mistakes.
[D]irectory	Displays the directory of any desired drive or subdirectory. Allows use of wildcard characters, i.e. "*.TXT".



**[BLOCK]****[C]opy [M]ove [I]nsert [D]elete [S]wap**

- [C]opy** First time, marks the beginning of the desired block of text and displays **1 END** on status line. Second time, prompts for the name of the desired text register. The block of text between the marked position and the current cursor position is copied to the register. The status line message changes to **TEXT**. Optionally, precede the name with a "+" to append the block of text to any text already in the register. Pressing **[BLOCK]-C** twice in a row (without moving the cursor) empties the specified text register. Note: either the beginning or the end of the block of text may be marked first.
- [M]ove** Similar to "[C]opy", except that the text block is deleted from the edit buffer after it is moved to the text register.
- [I]nsert** Prompts for the name of the text register which is to be inserted at the current cursor position. The register itself is not changed.
- [D]elete** Similar to "[C]opy", except the text block is simply deleted.
- [S]wap** Swaps the position of the cursor with the previously marked block position. Pressing it twice restores the cursor and block marker positions.

**[PRINT]****[A]ll [B]lock [E]ject [L]ines [M]argin**

- [A]ll** Prints the entire file. All text is offset from the left edge of the paper by the "*Printer margin*". Typically prints 60 lines of text on each page, with a 3 line margin at the top and bottom.
- [B]lock** Prints a block of text. The block is specified similar to "**[BLOCK]-Copy**".
- [E]ject** Advances the printer to the next page. (Depending upon the "**PP 4**" command parameter, either line feeds or a form feed are used.)
- [L]ines** Prompts for the number of lines to be printed on each page. Reducing the number increases both the top and bottom margin. (The physical number of lines per page is set with the "**PP**" command.)

[M]argin      Prompts for the "Printer Margin". This is the number of columns that all text is offset from the left edge of the paper.

**[WINDOW]      [C]reate [D]elete [S]witch [Z]oom**

[C]reate      Creates a new window in current window. Prompts for window position, size and name.

[D]elete      Prompts for name of window to be deleted.

[S]witch      Switches to another window. Prompts for name of desired window.

[Z]oom      Zooms current window to the full screen. Stays in effect until next window switch.

**[USER]      [W]ord Wrap [J]ustify [I]ndent [A]uto-Indent**

[W]ord Wrap      Prompts for the new Word Wrap or "right" margin for word processing and [FORMAT PARAGRAPH].

[J]ustify      Enables or disables the right margin justification option for [FORMAT PARAGRAPH].

[I]ndent      Prompts for a new "Indent Increment" used by [INDENT] and [UNDENT].

[A]uto-Indent      Enables or disables "Auto-Indent" mode - the indent for new lines is set to be the same as the previous text line.

**[MISC]      [M]atch Parentheses [U]c/Lc [I]nsert [O]verstrike**

[M]atch Paren      Performs nested parentheses matching for structured programming languages. With the cursor positioned at one of the eight characters {, }, [, ], <, >, (, ), it moves the cursor forward or backward to the matching pair. Otherwise, searches forwards for one of the characters.

[U]c/Lc      Changes upper case letter at cursor to lower case and vice versa. Moves cursor to next character.

[I]nsert      Changes to "Insert" mode if not already set.

[O]verstrike      Change to "Overstrike" mode if not already set.

'n' Denotes a positive number or expression (# represents 65535).  
 'm' Denotes a number or expression which may be negative.  
 'p,q' Denotes two edit buffer positions for a "block" of text.  
 'c' Denotes a conditional expression. Value of "0" is FALSE, value of "1" (or any other non-zero value) is TRUE.  
 'r' Denotes a register name. For text registers, 'r' is "0" - "9", "A" - "Z" or "@". For numeric registers, 'r' is "0" - "25".  
 'w' Denotes a window name. Can be any character, but is usually "0" - "9", "A" - "Z", "@" or "\$" (Command Mode window).  
 'ss' Denotes a search string and may contain pattern matching codes.  
 'text' Denotes a text string being inserted, substituted or displayed.  
 'file' Denotes a filename in the normal MS-DOS (CP/M) format with optional drive and pathname (CP/M user number).  
 <ESC> The <ESC> key. Alternatively, explicit delimiters may be used.  
 <term> Either an <ESC> or a <RETURN>.

## SPECIAL CHARACTERS

<CTRL-Q> Literal Character. Next character is taken literally.

@ Precedes commands to indicate that explicit delimiters are being used.

Precedes F, L, N, S commands to suppress error handling.  
 Precedes ER, EW, RQ, XT, YD commands to suppress <CR><LF>.

Precedes F, S commands to perform search to end of file.  
 Precedes B, K, L, T, Z, EM, RM, YM commands to indicate global operation on entire file.

Shorthand for ".erJL". Used with "ES 8 2" following the F, S and L commands. Does not clear the error flag ".er".

# Represents maximum positive number 65535. Signifies "forever" or "all occurrences of".

## BASIC COMMANDS

nA Append 'n' lines from the input file.  
 OA Append lines until edit buffer is nearly full.

-nA Read back 'n' lines from the output file.  
 -OA Read back lines until edit buffer is nearly full.

B Move the edit pointer to beginning of edit buffer.  
 \_B Move to the beginning of the file.

mC Move the edit pointer by 'm' characters.

mD	Delete 'm' characters from the text.
p,qD	Delete the block of text marked by 'p' and 'q'.
nFss<ESC>	Search for the 'n'th occurrence of 'ss' in the current edit buffer and position the edit pointer after it. 'ss' may be up to 80 characters long.
n_Fss<ESC>	Search for the 'n'th occurrence of 'ss', performing auto-buffering if necessary. The edit pointer is positioned after the 'n'th 'ss' if found.
-nFss<ESC>	Search backwards for the 'n'th previous occurrence of 'ss' and position the edit pointer at the beginning of it.
Gr	Same as "RG" for compatibility with older versions.
H <term>	On-line help for Command Mode commands using the help file "VPHELP.HLP".
H text<term>	Immediately search help file for "\text\" and display the associated help message.
Itext<ESC>	Insert 'text' into the edit buffer at the edit pointer. The edit pointer is moved past 'text'.
-Itext<ESC>	Overwrite existing characters with 'text'.
mK	Kill 'm' lines of text.
m_K	Kill lines, performing auto-buffering if needed.
p,qK	Delete the block of text marked by 'p' and 'q'.
mL	Move the edit pointer by 'm' lines.
m_L	Move, performing auto-buffering if needed.
Mr	Execute text register 'r' as a command macro. When finished, continue processing the commands following "Mr".
nNss<ESC>	Short hand for "n_Fss<ESC>".
Qr	Not a command! Access value in numeric register 'r'.
nSss<ESC>text<ESC>	Search for the next occurrence of 'ss', and if found, change to 'text'; perform 'n' times.
mT	Type (display) 'm' lines of text.
m_T	Type lines, performing auto-buffering if needed.
p,qT	Type the block of text marked by 'p' and 'q'.
U	Display # of free bytes in current segment / # bytes in current edit buffer / total # bytes in all text registers which are not edit buffers. Also displays name of the current edit buffer (except for the main '@' edit buffer).

V	Go into visual mode setting cursor position from current edit pointer.
-V	Perform an update-only of the visual mode window. Remains in Command Mode.
nW	Write 'n' lines to the disk from the beginning of the edit buffer and delete them from the buffer.
OW	Write out the edit buffer up to the current line.
-nW	Write last 'n' lines in the edit buffer to the ".\$R\$" file. This makes more memory space free.
-OW	Write the current line and the rest of the edit buffer to the ".\$R\$" file.
Z	Move the edit pointer past the last character in the edit buffer.
_Z	Move the edit pointer to the end of the file.

## EXTENDED COMMANDS

EA	Saves the file being edited on disk and allows further editing to continue.
EB <i>file</i> <term>	Begin editing the file ' <i>file</i> '. Equivalent to invoking VEDIT PLUS with ' <i>file</i> '. If the file does not exist, "NEW FILE" is displayed.
EB <i>infile outfile</i>	Open ' <i>infile</i> ' for reading and ' <i>outfile</i> ' for writing.
EC	Allow user to change disks. Used for write error recovery, or to edit files on other disks.
ED <i>file</i>	Display disk directory. Optional drive, path (CP/M user number) and wildcard characters "?" and "*" may be used.
nED	Display the filenames in 'n' columns instead of 4.
-ED	Do not display header. Display names in one column.
:ED <i>file</i>	Test for existence of the file ' <i>file</i> '. Report result in ".rv".
EER	Change active edit buffer to text register 'r'.
EE#r	Change to the edit buffer specified by the ASCII value contained in numeric register 'r'.
-EER	Do not perform auto-buffering on current edit buffer before changing to edit buffer 'r'.
EF	Close the current output file WITHOUT writing additional text. The input file and edit buffer are unaffected. USE WITH CAUTION! INCORRECT USE CAN RESULT IN DELETED FILES!
EFY	Skip the confirmation prompt.

EG*file*[line range] Insert the line range of the file '*file*' into the edit buffer at the edit pointer. If no line range is specified, the entire file is inserted.

EH <term> On-line help for Command Mode topics using the help file "VPEHELP.HLP".

EH *text*<term> Immediately search help file for "*text*" and display the associated help message.

nEI Insert the character with decimal value 'n' into the edit buffer at the edit pointer. Values 0 - 25 and 27 - 254 are allowed.

-nEI Overwrite the existing character at the edit pointer with the character with value 'n'.

nEJ Jump the edit pointer to the 'n'th position in the edit buffer.

EK *file* Erase (kill) the file(s) '*file*'. This frees more space on the disk. Optional drive, path, (user number for CP/M) and wildcard characters "?" and "\*\*".

-EK *file* Skip the directory display and confirmation prompt.

EK *file* Y Same as "-EK *file*".

EL*file*[line range] Display the specified line range of the file '*file*' with line numbers. If no line range is specified, the entire file is displayed.

nEMss<ESC> Check 'ss' for match with text at edit pointer; move edit pointer if match. Must match 'n' times to be successful. Report results in ".rv", ".n" and ".er".

nEN Free 'n' bytes of memory space in edit buffer, if possible.

OEN Free customized "Free memory after auto-read".

EN Squeeze edit buffer down to customized "page" size.

mEO Same as "mPR" for compatibility with older versions.

EP <term> Display current values of all parameters.  
 EP n <term> Store value of parameter 'n' in ".rv".  
 EP n k Change the value of parameter 'n' to 'k' after saving the old value in ".rv". Affects the current and all subsequently created edit buffers.  
 -EP n k Affect only the current edit buffer.

1	Cursor type	(Mem mapped only)	(0 - 4)
2	Cursor blink rate	(Mem mapped only)	(10 - 255)
3	Indent Increment		(1 - 20)
4	Lower/upper case conversion		(0, 1, 2 or 3)
5	Conditional conversion character		(32 - 126)
6	Delay for Command Mode scrolling		(0 - 255)
7	Right margin for Word Wrap (0 = Off)		(0 - 255)
8	Bit 8 Allowed (See table)		(0 - 7)
9	Cursor positioning mode		(0 - 2)
10	Horizontal scroll margin		(40 - 255)
11	Horizontal scroll increment		(1 - 100)

EQ Quit (abandon) the edit session; any text changes are lost. Leaves disk files exactly as before the session started. If there are any more edit buffers, move to one of them; otherwise return to DOS.

EQY Skip the confirmation prompt.

EQA Quit all edit buffers without saving changes.

ER *file* Open the file '*file*' for input.  
 ER <term> Display the input (read) filename.  
 +ER <term> Display filename and include its drive and/or path.  
 :ER <term> Suppress "INPUT FILE:" message and final <CR><LF>.

ES <term> Display current values of all switches.  
 ES n <term> Store value of switch 'n' in ".rv".  
 ES n k Change the value of switch 'n' to 'k' after saving the old value in ".rv". Affects the current and all subsequently created edit buffers.  
 -ES n k Affect only the current edit buffer.

1	Expand <TAB> key with spaces	(0=NO 1=YES)
2	Auto buffering in Visual Mode	(0=NO 1=YES 2=BACK)
3	Auto-Indent mode	(0=NO 1=YES)
4	Point past text reg. insert	(0=NO 1=YES)
5	Equate UC/LC when searching	(0=NO 1=YES)
6	MS-DOS End-of-file padding	(0=NO 1=YES)
7	Reverse Upper and Lower case	(0=NO 1=YES)
8	Suppress search errors	(0=NO 1=YES)
9	Explicit string delimiters	(0=NO 1=YES)
10	Global file operations	(0=NO 1=YES)
11	Justify paragraphs	(0=NO 1=YES 2=UNJUSTIFY)

ET n1,n2,n3 ... Set new tab positions. "ET" is followed by up to 33 decimal numbers specifying the tab positions.

ET n Alternatively, if "ET" is followed by just one number, tabs are set to every 'n' positions.

ET <term> Display current tab positions.

EU d:subd Change to drive 'd' and/or subdirectory 'subd'.

EU du: Change to drive 'd' and/or user number 'u'. (CP/M)

EU <term> Display current drive/subdirectory.

EV Display the VEDIT PLUS version number.

EW *file* Open the file '*file*' for output. Any existing file by that name will be renamed to have the filename extension ".BAK" when the file is saved to disk.

EW <term> Display the output (write) filename.

+EW <term> Display filename and include its drive and/or path.

:EW <term> Suppress "OUTPUT FILE:" message and final <CR><LF>.

EX Exit edit buffer after saving entire file being edited on disk. If there are any other edit buffers, switch to one of them; otherwise, return to DOS.

EXA Exit all edit buffers, saving edit changes wherever an output file is open.

EY Save the entire file being edited on disk and remain in the current edit buffer. You can then use the "EB" command to edit another file.

EZ Quit (abandon) the edit session; any text changes are lost. Leaves disk files exactly as before the session started. Remain in current edit buffer.

EZY Skip the confirmation prompt.

## JUMP COMMANDS

cJL If 'c' is TRUE, exit the current REPEAT-UNTIL loop.

cJM If 'c' is TRUE, exit the currently executing macro.

cJN If 'c' is TRUE, start next iteration of current loop.

cJO If 'c' is TRUE, abort and return to Command Mode prompt.

cJLabel<ESC> If 'c' is TRUE, jump to "!label!".



## OPERATING SYSTEM COMMAND (For MS-DOS / PCDOS only)

OC <i>command</i>	Execute the specified DOS command or program. Returns to VEDIT PLUS after command/program is done.
OC  Rr	Execute the DOS command in text register 'r'.
OD	Display the current date. The commands "YI OD -YI" insert the date into the text.
:OD	Suppress the final <CR><LF>.
OS	Temporarily enter MS-DOS. Return to VEDIT PLUS with the MS-DOS command "EXIT".
OT	Display the current time.
:OT	Suppress the final <CR><LF>.

## PRINT COMMANDS

PE	Eject - advance printer to next page.
-PE	Reset only the internal line counter used for printing.
PP <term>	Display current value of print parameters.
PP n <term>	Store value of print parameter 'n' in ".rv".
PP n k	Change the value of parameter 'n' to 'k' after saving the old value in ".rv". The print parameters are:
1	Physical lines per page (5 - 100)
2	Printed lines per page (1 - 100)
3	Left margin for printing (0 - 100)
4	Use Form-Feed for new page (0 = NO, 1 = YES)
mPR	Print the specified 'm' lines of text.
m_PR	Print lines, performing auto-buffering if needed.
p,qPR	Print the block of text marked by 'p' and 'q'.

## TEXT REGISTER COMMANDS

R*	Treat this line as a comment.
+RAr	Auto-execute text register 'r' in place of the "COMMAND:" prompt. 'r' cannot be "0" (zero).
RA <term>	Disable auto-execution. Or use "RA0".
mRCr	Copy 'm' lines from edit buffer into text register 'r'.
mRC+r	Append text to any existing contents in register 'r'.
p,qRCr	Copy the block of text marked by 'p' and 'q' into 'r'.
RDr	Dump the contents of register 'r' onto the console. Control characters are not expanded.

RER	Empty text register 'r'.
+RER	Empty text register 'r', even if currently executing.
RGR	Get (insert) contents of register 'r' into edit buffer.
RIrtext<ESC>	Insert 'text' into register 'r'.
RI+r	Append 'text' to any existing contents of register 'r'.
RJR	Jump to macro register 'r' without "returning" to the current macro after 'r' is done executing. Don't follow "RJ" with other commands - they won't be executed.
RLr file	Load register 'r' from file 'file'.
RL+r	Append 'file' to contents of text register 'r'.
+RLr	Perform extended search for 'file'.
RMr	Compare contents of register 'r' with text at edit pointer. Move the edit pointers over as many characters as matched. Return # characters matched in ".n" and the match result in ".rv": {0,1,2} for {=,<,>}.
_RMr	Perform automatic disk buffering on current edit buffer, if necessary, to complete match. Often used when 'r' is also an edit buffer.
RPr	Print contents of register 'r'.
RQrtext<ESC>	Query for user input with 'text'. Get keyboard input line including terminating <CR><LF> or <ESC><ESC> and save in text register 'r'.
+RQr	As above, but prompt on status line.
:RQr	Get keyboard input line without the <CR><LF> or <ESC><ESC>.
RQ+r	Append input line to contents of text register 'r'.
RSr file	Save contents of register 'r' in file 'file'.
-RSr file	Skip the confirmation prompt when file already exists.
RTr	Type (display) contents of register 'r'. Control characters are expanded, <ESC> is represented as "\$". An embedded <CTRL-S> "stop character" causes a pause - type any character to continue.
RU	After displaying memory usage as in the "U" command, display the size of each text register. Asterisks indicate edit buffers.
RXr execfile	Load file 'execfile' into register 'r' and execute register 'r' as a command macro. Perform extended search for 'execfile'. Equivalent to "+RLr execfile\$ Mr".

## NUMERIC REGISTER COMMANDS

mXAr            Add 'm' to the value in numeric register 'r'.

XXR*text*<ESC>   Prompt for user input with '*text*'. Store next keyboard character in numerical register 'r'.

+XKr            As above, but prompt on status line.

:XKr            Allow <CTRL-C> as valid input character.

XQR*text*<ESC>   Query for user input with '*text*'. Store evaluated keyboard input in numerical register 'r'.

+XQR            As above, but prompt on status line.

mXSr            Set value of numerical register 'r' to 'm'.

XTr            Type (display) value of numeric register 'r' in decimal followed by <CR><LF>. Number is right justified.

-XTr            Display the number left justified.

:XTr            Suppress the <CR><LF> following the number.

## MISCELLANEOUS COMMANDS

nYD            Dump character with ASCII value 'n' to the console.

n:YD            Suppress the <CR><LF> following the character.

nYEA            Set window attribute to 'n' for displayed characters.

n1,n2YEA       Set window attribute to 'n1' for text characters and to 'n2' for characters used to erase window.

YEC            Clear entire window and home the cursor.

nYEH           Position cursor horizontally to column 'n'.

YEL            Erase from cursor to end-of-line in window.

YES            Erase from cursor to end-of-screen in window.

nYEV           Position cursor vertically to line 'n'.

nYF            Format the paragraph the edit pointer is in. Use 'n' as the left margin. If no 'n' is specified, use Visual Mode indent position as the left margin. Use word wrap column as right margin. Justify if justification switch is set. After formatting, position the edit pointer to the beginning of the next paragraph.

YI	Divert console output until next command prompt into the edit buffer at the edit pointer and update the edit pointer. (May work very slowly.)
-YI	Disable "YI", allow output to go back to the console.
YK <i>file</i>	Save current keyboard layout, including any keystroke macros, to the file ' <i>file</i> '.
YL <i>file</i>	Load new keyboard layout from the file ' <i>file</i> '.
+YL <i>file</i>	Perform extended search for the file.
YM	Search for the matching pair for "{" - "}", "[" - "]", "<" - ">" and "(" - ")". If edit pointer is not at one of the characters, search forwards for one.
_YM	Search for matching pair, performing auto-buffering if necessary.
YP	Divert console output to the printer until next command prompt.
-YP	Disable "YP", allowing output to go back to the console.
mYS	Strip the 8th bit from all characters in the specified 'm' lines.
m_YS	Strip characters, performing auto-buffering if needed.
YT <i>text</i> <ESC>	Type ' <i>text</i> ' on the console.
+YT <i>text</i> <ESC>	Type ' <i>text</i> ' on the status line.
YWD	Delete the current window.
YWDw	Delete the window 'w'.
YWI	Initialize screen - delete all windows, reset attributes.
YWS	Switch to the default window "@".
YWSw	Switch to the window 'w'.
YWZ	Zoom current window to full screen.
YWBw n	Create window 'w' of 'n' lines at bottom.
YWLw n	Create window 'w' of 'n' columns at left.
YWRw n	Create window 'w' of 'n' columns at right.
YWTw n	Create window 'w' of 'n' lines at top.

## SEARCH PATTERN MATCHING CODES

A	Match any alphabetic letter, upper or lower case
B	Match a blank - single space or tab
C	Match any control character
D	Match any numeric digit - "0" - "9"
F	Match any alphanumeric - a letter or a digit
L	Match end-of-line: <CR> and/or <LF>, Form-feed or EOF
M	Multi - match any sequence of zero or more characters
N	Match any character except following character or pattern
Pr	Access contents of text register 'r' as pattern set
Rr	Access contents of text register 'r' as variable string
S	Match any separator - not a letter or digit
T	Match selected separators (terminators)
U	Match any upper case letter
V	Match any lower case letter
W	Match white space - single or multiple spaces or tabs
X	Match any character
Y	Match multiple characters until the next pattern matches
<	Match beginning of line (zero length match)
>	Match end of line (zero length match)
	Use "  " when you need to search for a " "

## INTERNAL VALUES

.b	Name of current edit buffer (in ASCII)
.c	ASCII value of character at the edit pointer
.ef	End-of-file flag for input file (or file not yet opened)
.er	Error flag. Set by F, L, S and EM
.es	Search error flag. Set/reset by F, S, EM, RM
.ew	Write error flag. Set/reset by last write operation
.f	Number of free memory bytes
<del>.m</del> .m /	Current left margin (indent position)
.m	Name of currently executing text register (in ASCII)
.n	Number of characters matched by F, S, EM, RM
.of	Output file open flag ("1" if output file is open)
.p	Edit pointer's offset in edit buffer (0 = begin of buffer)
.rm	Absolute value of the remainder from last division
.rtr	"Type" for register r (unused, text reg, edit buffer)
.rv	Value returned by :ED, EM, EP, ES, PP, and RM
.t	Next tab position based on ".x"
.ur	Number of bytes in register 'r'
.v	Numerical value of expression at edit pointer
.vm	Visual mode "1 END" marker's offset in edit buffer
.wa	Attribute for displayed text in current window
.wd	Current display type (CRT, MM, IBM Mono, IBM Color)
.we	Attribute for erased parts of current window

(Continued on next page)

.wh	Horizontal size (in columns) of current window
.wn	The name of the current window (in ASCII)
.wt	Total number of windows on the screen
.wv	Vertical size (in lines) of current window
.wx	Horizontal ("x") cursor position in window
.wy	Vertical ("y") cursor position in window
.x	Horizontal column position for character at edit pointer
.y	Line-number in the file for line the edit pointer is on

**NUMERICAL OPERATORS**

+	Addition
-	Subtraction (also performs unary minus function)
*	Multiplication
/	Division
%	Remainder of division
&	Bitwise AND
^	Bitwise OR
'	Bitwise complement (also called 1's complement)

**RELATIONAL OPERATORS**

<	Less than
<=	Less than or equal to
=	Equal to
>	Not equal to
>=	Greater than or equal to
>	Greater than

**LOGICAL OPERATORS**

&	AND - TRUE only if both operands are TRUE
^	OR - TRUE if either operand is TRUE
~	NOT - Flips the truth value of the following operand

**OPERATOR PRECEDENCE**

Highest	'	Complement
	* / %	Multiplication, Division, Remainder
	+ -	Addition, Subtraction
	< > = etc.	Relationals
	~	NOT
	&	AND (Both Bitwise and Logical)
Lowest	^	OR (Both Bitwise and Logical)