

VEDIT PLUS

## TUTORIAL



VEDIT PLUS TUTORIALNOTE

This is an in-depth Tutorial. The new user should first follow through the "30 Minute Tutorial" in the Introduction. That is enough to begin using VEDIT PLUS.

This in-depth Tutorial covers the entire "*Visual Mode*" of VEDIT PLUS, which is the primary editing mode. The Tutorial is organized into topics, starting with simple ones and proceeding to more complex ones. Most topics begin with a descriptive overview, followed by a more in-depth "Reference", followed by step-by-step directions for performing common editing operations. The "Reference" sections cover details about the Visual Mode that the new user need not be concerned with. The later part of the Tutorial gives step-by-step directions for performing common operations in the Command Mode.

Work through the Tutorial at your own pace. The basics of VEDIT PLUS are easy enough to use that you do not need to finish this entire Tutorial before using VEDIT PLUS productively. This Tutorial is intended to be learned gradually, over the course of days, where you can come back to it as you need additional capabilities during the course of your work. The first time through, you can ignore the "Reference" sections of each topic.

If you want to experiment on a file during the Tutorial, you will find it easier to use a file that already exists instead of creating a new one. If you don't have any such files available, you can copy one of the help files with a filename extension of ".HLP" from the copy of your VEDIT PLUS disk to your work disk. Make sure you are working with a copy of the ".HLP" file so that you don't accidentally alter the real ".HLP" file. (We assume, of course, that you have made a copy of the original diskette.)

When working through the Tutorial, keep in mind that not every possible text editing situation or sequence of commands is included here. In particular the Command Mode is barely scratched. As you gain more experience with VEDIT PLUS, you will find out that many editing tasks can be performed more easily in the Command Mode.

Remember that On-line help is available in both Command Mode and Visual Mode. Accessing the On-line help is described in the "30 Minute Tutorial" in the Introduction. Use the Index at the back of the manual to point you to additional information about any topic.

NOTATION

- [*function*] A Visual Mode edit function such as [CURSOR UP] or [INDENT]. The actual key(s) you press to perform the function are chosen during installation. If you are using one of our example keyboard layouts, refer to the layout sheet. The IBM PC layout uses predominately the function keys <F1> through <F10> and the cursor keys. A few control characters such as <CTRL-U> are also used.
- [*menu-function*]-*x* Some editing functions such as [GOTO] prompt on the status line with a one line menu. For example, "[GOTO]-B" denotes that you press [GOTO] and answer the prompt with a "B". "Keystroke macros" allow you to assign a single function/control key to perform an operation such as "[GOTO]-B".
- <CTRL-*x*> A control character such as "<CTRL-U>" is typed by holding down the CONTROL key like a SHIFT key and typing the letter, in this case "U".
- RETURN indicates pressing the RETURN key (labeled "<---'" on the IBM PC, "ENTER" on many keyboards and "CR" on some terminals). This starts a new line in Visual Mode. In Command Mode, you normally press RETURN at the end of command lines.
- <ESC> is simply the <ESC> key. It generates a special control character. It is often used in Command Mode to mark the end of text "strings". This manual often uses the shorthand of "\$" for the <ESC> key because "\$" is what VEDIT PLUS displays on the screen when an <ESC> is pressed in Command Mode.
- \$ is a shorthand for "<ESC>". Wherever "\$" appears in a Command Mode example, press the <ESC> key.

INVOKING VEDIT PLUS

To invoke VEDIT PLUS from the operating system type **VPLUS** followed by the name of the file to be edited or created. For example:

**VPLUS LETTER.TXT**

VEDIT PLUS will then read in the file "LETTER.TXT" or, if you are creating the file, briefly display the message "NEW FILE". It will then normally go into the "Visual Mode" and display the beginning of the file on the screen. The "*Status Line*" at the top of the screen will display the filename "LETTER.TXT". Also visible will be the "*Cursor*" which indicates at what position on the screen you are editing. It will initially be in the upper left hand corner.

Invoking VEDIT PLUS (Reference)

The examples in this manual always assume that VEDIT PLUS will initially enter Visual Mode. However, if you prefer to have VEDIT PLUS enter Command Mode, you can choose this when you install VEDIT PLUS. See Task 8.7 of the installation.

VEDIT PLUS can also be invoked with two files - the form is:

**VPLUS *infile outfile***

This lets you take an existing file '*infile*', make changes to it and save it as the new file '*outfile*'. '*Infile*' is not altered in the process.

To edit a file which is more than half a disk long, you must invoke VEDIT PLUS with two files - '*infile*' is the file to be edited and '*outfile*' is specified to be on another disk drive with a nearly blank disk.

Invoking VEDIT PLUS with one file is equivalent to invoking it with no files, entering Command Mode and issuing the command:

**EB *filename***

Invoking VEDIT PLUS with two files is equivalent to invoking it with no files, entering Command Mode and issuing the commands:

**ER *infile*  
EW *outfile***

-- OR --

**EB *infile outfile***

INVOKING VEDIT PLUS

VPLUS FILENAME.EXT

You will enter Visual Mode with the beginning of the file displayed on the screen ready for editing.

If the file does not already exist, the message "NEW FILE" will be displayed briefly.

VPLUS

You will enter Visual Mode with an empty screen. Choose a file to edit with the [FILE] function. Experienced users often use this option when they immediately want to enter Command Mode.

VPLUS INFILE.EXT OUTFILE.EXT

You will enter Visual Mode with the beginning of "INFILE.EXT" on the screen ready for editing. "OUTFILE.EXT" will appear on the status line. Any edit changes will be saved in "OUTFILE.EXT. "INFILE.EXT" will remain unchanged.

### KEYBOARD CHARACTERS AND EDIT FUNCTIONS

In Visual Mode normal keystrokes are entered directly into the text, while function and control keys are interpreted as edit functions. All the letters, numbers and other normal characters on your keyboard can be directly entered as new text. Go ahead and try typing a few words in right now. Notice that as each character is typed, it appears at the cursor position and the cursor then moves to the right. If there already were characters on the line, you have just overwritten them. As described later, you can also enter "Insert" mode to insert characters without overwriting. The edit functions are performed by pressing function keys, cursor keys and control keys on the keyboard. The customized keyboard layout determines which edit function each control or function key performs. You can use the installation program to change the layout in any way you wish. You can also use "keystroke macros" to assign an entire sequence of commonly typed keys to a function or control key.

#### Keyboard Characters (Reference)

VEDIT PLUS allows complete flexibility in what keys are pressed to perform the edit functions. You can press function keys, cursor keys, control characters (such as <CTRL-U>), or type "escape sequences". For simplicity we refer to all of these as "control and function keys". Escape sequences are primarily used on keyboards that do not have many function keys. Escape sequences such as "<ESC>-B" are typed by first pressing <ESC> and then "B". Technically, function keys usually send either an escape sequence or an "8 bit" character - both are handled by VEDIT PLUS.

Non-customized function and control keys are usually ignored in Visual Mode. You can insert single control or "8 bit" characters by preceding them with [NEXT CHAR LITERAL]. However, for special applications, you can optionally have those characters generated by non-customized control and function keys entered into the text. This option is controlled by the "EP 8" command parameter.

#### Edit Functions

The most commonly used edit functions are the "cursor movements" which only move the cursor around on the screen and scroll the screen to display different parts of the file, but do not change the file in any way. Other edit functions perform operations such as deleting text, searching for text and much more. Seven edit functions display a "pop-up" menu of possible operations - you type a letter to select an operation. (Keystroke macros can be used to setup "hot keys" which skip the menu prompt.)

### CURSOR MOVEMENT

The following two pages list all of the cursor movements and you should briefly try them all out. Don't be concerned about remembering them all now. Some are more important than others and you will get along quite well knowing only [CURSOR UP], [CURSOR DOWN], [CURSOR RIGHT], [CURSOR LEFT], [ZIP], [PAGE UP] and [PAGE DOWN]. These movements are all labeled on an IBM PC; [ZIP] moves to the end of the line and is labeled "End".

#### Cursor Movement (Reference)

In general the cursor can only move to where there is text - it cannot move to empty parts of the screen. For example, pressing [CURSOR LEFT] with the cursor at the beginning of a line moves the cursor to end of the real text on the previous line. Note that spaces and tab characters count as "real" text.

For convenience, VEDIT PLUS offers three different modes of cursor movement. Two modes allow the cursor to move straight up and down, even past the ends of short lines.

MODE 0 : the cursor can never be positioned past the end of a line. For example, if you move the cursor down from the end of a long line to a shorter line, the cursor will also move left to the end of the shorter line.

MODE 1 : the cursor can be moved straight up and down from a long line past short lines to another long line. If you attempt any edit change, i.e. typing in new text, with the cursor past the end of a line, the cursor will first move left to its "correct" position. Since it would be annoying during horizontal scrolling to have the screen scroll back as the cursor is moved past short lines, Mode 0 is temporarily changed to Mode 1 when the screen is scrolled to the right.

MODE 2 : the cursor moves identically to Mode 1. However, if the cursor is past the end of a line and you type text, spaces are automatically inserted from the end of the line up to the newly entered text. This mode is handy for filling out tables and other formatted text. Note, however, that the many spaces consume additional memory and disk space.

A little experimentation is best for understanding these modes and deciding which you like best. Mode 1 is the recommended setting. Other word processors generally operate in one of these three modes and you may want to pick one that you are already familiar with. The default mode is set during installation and may be changed with the "EP 9" command parameter.



CURSOR MOVEMENT

| <u>Operation</u>  | <u>Command Sequence</u> |
|---|-------------------------|
| Move cursor right   | CURSOR<br>RIGHT         |
| Move cursor left  | CURSOR<br>LEFT          |
| Move cursor up  | CURSOR<br>UP            |
| Move cursor down  | CURSOR<br>DOWN          |
| Move cursor to the<br>beginning of the<br>current line                | BACK<br>TAB             |
| Move cursor to the<br>next tab position                               | TAB<br>CURSOR           |
| Move cursor to the<br>end of current line                             | ZIP                     |
| Toggle cursor between<br>the beginning and end<br>of the current line | LINE<br>TOGGLE          |
| Move cursor to the<br>beginning of the next<br>line                   | NEXT<br>LINE            |

CURSOR MOVEMENT (continued)

| <u>Operation</u>  | <u>Command Sequence</u> |
|---|-------------------------|
| First character of<br>the previous word                 | PREV<br>WORD            |
| First character of<br>the next word                     | NEXT<br>WORD            |
| Beginning of current<br>paragraph                       | PREV<br>PARA            |
| Beginning of the next<br>paragraph                      | NEXT<br>PARA            |
| Toggle cursor between<br>top and bottom screen<br>lines | SCREEN<br>TOGGLE        |

### SCREEN AND HORIZONTAL SCROLLING

The screen scrolls automatically as the cursor is moved towards the very top or bottom of the screen. Generally the screen scrolls when the cursor reaches about the third line from the top or bottom of the screen. This ensures that you always see a few lines before and after the line you are editing.

For convenience you can also use [SCROLL UP] and [SCROLL DOWN] to scroll the screen without having to move the cursor. This allows you to view lines which are just barely off the screen.

#### Horizontal Scrolling

VEDIT PLUS has the ability to scroll the screen sideways for editing documents with long lines, such as spreadsheets or structured programs. Similar to the Up/Down scrolling, the screen automatically scrolls as the cursor is moved toward the beginning or end of long lines. You can also scroll the screen horizontally without moving the cursor with [SCROLL RIGHT] and [SCROLL LEFT].

#### Horizontal Scrolling (Reference)

The screen may be scrolled right up to a margin called the "*horizontal scroll margin*" (which is independent of the word wrap margin). The default horizontal scroll margin is set during installation and may be changed in Command Mode with the command "EP 10 nnn". It has a maximum value of 255.

Lines longer than the horizontal scroll margin are wrapped to the next screen (window) line. These additional screen lines are called "*continuation lines*" and are indicated with a special "*continuation character*", typically a reverse video "-", in the leftmost column of the window. The continuation character is selected during installation.

By setting a scroll margin of 78 when the screen width is 80 (the right and left most columns are reserved) the horizontal scrolling is effectively turned off and you can view an entire long line, since it will be displayed on multiple screen lines.

To reduce the amount of unwanted side to side scrolling, several functions such as [ZIP] do not force the screen to scroll.

SCREEN SCROLLINGOperationCommand Sequence

Display previous line  
of text by scrolling  
screen up.

SCROLL  
UP

Display next line of  
text by scrolling  
screen down.

SCROLL  
DOWN

Scroll screen toward  
the end of long lines

SCROLL  
RIGHT

Scroll screen toward  
the beginning of long  
lines

SCROLL  
LEFT

QUICK CURSOR MOVEMENT

Besides moving the cursor by a character, word or line at a time, you can also quickly access more distant parts of the file. The most commonly used functions are [PAGE UP] and [PAGE DOWN] which move the cursor up and down by about 20 lines at a time.

The menu-function [GOTO] allows direct movement to the beginning or end of the file or to a particular line number. Press [GOTO]. The status line will change into a menu prompt:

```
[B]egin [H]ome [S]et [J]ump [L]ine [Z]end [E]nd
```

Type "B" to move to the beginning of the file or "E" to move to the end of the file. Type "L" to move to a particular line in the file and the status line will prompt:

```
Enter Line Number:
```

Enter the desired line number and a RETURN. Notice that the selected line number is now displayed on the status line.

NOTE: As with all menu-functions, you can abort [GOTO] by pressing <CTRL-C> (or [CANCEL] on an IBM PC).

[GOTO] Function (Reference)

When editing a large file, which is larger than memory, the [GOTO] menu lets you select "Begin" to go to the very beginning of the file or "Home" to go to the beginning of what is currently in memory. Similarly, "Zend" is the end of what is currently in memory, while "End" is the very end of the file.

Setting and Jumping to Text Markers

You can set up to ten invisible markers in the text and later jump back to them. The markers are set with the [GOTO]-Set function (press [GOTO]; then press S). The status line will then prompt: **MARKER NUMBER 0-9**. Press a digit corresponding to the desired marker number. Similarly, you can jump back to a marked position with the [GOTO]-Jump function.

The marked positions are relative to the text - the markers adjust themselves as text is inserted and deleted. All markers are initially reset. Attempting to jump to a reset marker has no effect. Any markers within text which is written to disk are reset.

QUICK CURSOR MOVEMENT

Purpose: To rapidly access regions of the file not currently displayed on the screen.

| <u>Operation</u>                             | <u>Command Sequence</u>   |
|--|---|
| Previous Page of text                        | PAGE<br>UP  |
| Next Page of text                            | PAGE<br>DOWN  |
| Very beginning of the file                   | GOTO<br>Then type "B"   |
| Very end of the file                         | GOTO<br>Then type "E"   |
| Set invisible text marker at cursor position | 1. GOTO<br>Then type "S"  |
|  | 2. Type digit "0 - 9" to specify which marker to set. Or set marker "0" by pressing [GOTO] again or RETURN.     |
| Move cursor to previously set text marker    | 1. GOTO<br>Then type "J"  |
|  | 2. Type digit "0 - 9" to specify which marker to go to. Or go to marker "0" by pressing [GOTO] again or RETURN. |

### ENTERING NEW TEXT

The function [INSERT] toggles between the "Overstrike" and "Insert" modes. In Overstrike mode the text you enter over-writes any existing text. In Insert mode the **INSERT** message appears on the status line and the new text is inserted with the existing text shifting to the right. Try it to see the difference between the two modes.

To insert new lines of text, first move to the end of the previous line (with [ZIP]) and press RETURN. This opens up a blank line on the screen on which to enter text. You can enter multiple lines of text by simply pressing RETURN at the end of each line. If the cursor is in the middle of a line when you press RETURN, the line is split into two lines with the character at the cursor position and all following characters moving to the new line. As described later, [DELETE] can be used to merge lines back together.

#### Entering New Text (Reference)

If desired, you can customize VEDIT PLUS to start in "Insert Mode". (See Installation Task 7.3.)

When you press RETURN with the cursor in the middle of a line the line is split into two lines - technically a carriage return <CR> and line feed <LF> pair is inserted into the text.

#### Tab Character (Reference)

The [TAB CHARACTER] function is usually assigned to the <TAB> key on the keyboard ("-->|" on the IBM PC). Pressing the <TAB> key inserts the tab character into the text. The tab character is displayed with spaces on the screen to the next tab position, even though the spaces do not exist in the edit buffer. The tab positions are normally set to every eight positions, but can be changed from Command Mode with the "ET" command.

The cursor moves over a tab character as a unit, i.e. a single [CURSOR RIGHT] might move from column 1 to column 9. This is indicative of tabs being single characters. When the cursor is at the tab character, it is displayed at the left side of the displayed spaces. Tab characters are commonly used when writing programs and aligning tabular data. Paragraphs are best indented not by using a <TAB>, but rather by typing four or five spaces.

As a convenience, you do not have to be in Insert Mode to insert text on top of a tab character - the tab character will not be overwritten until you reach its last displayed position.

Don't confuse the [TAB CHARACTER] and [TAB CURSOR] functions. The latter is strictly a cursor movement and has nothing to do with tab characters. It only moves the cursor to the next tab position and is equivalent to repeatedly pressing [CURSOR RIGHT].

Optionally, the [TAB CHARACTER] function can insert spaces to the next tab position. This is equivalent to you typing in the spaces. While this uses up more disk space and is not normally recommended, it is useful in some applications. This option may be selected with the command "ES 1 1". This option is handy with FORTRAN and COBOL programs.

You may want to expand the <TAB> key with spaces when editing files which will be sent to a mainframe computer, because many mainframes do not support tab characters.



ENTERING NEW TEXTOperationCommand Sequence

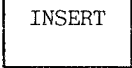
Entering text into the edit buffer --- beginning an empty file or continuing at the end of a file.

NONE - Move cursor wherever you like and begin typing. What you see is what you get.

Overstriking (typing over existing text)

1. Position cursor over first character to be overtyped.
2. Retype.

Inserting new characters between existing characters

1.  INSERT
2. Type new text

Watch for **INSERT** message on status line

3.  INSERT

"Insert" message disappears

Alternatively, you may prefer to stay in Insert Mode.

DELETING TEXT

You can delete text on a character, word, line or block basis. The delete functions are described on the next page. Go ahead and try out [DELETE], [BACKSPACE], [ERASE EOL], and [ERASE LINE].

To merge (concatenate) two lines together, move the cursor to the end of the first line and press [DELETE]. Or you can move to the beginning of the first line and press [BACKSPACE]. Go ahead and try splitting lines by typing a RETURN in the middle of the line and then merging the lines back together by pressing [BACKSPACE].

Provided that you have not moved the cursor off the current line and did not erase the entire line with [ERASE LINE], the line can be restored to its original contents (before any deletions or insertions were made) by pressing [UNDO].

Deleting Blocks of Text

The [BLOCK] menu-function can be used to delete a small or a large block of text such as a paragraph. To delete a block of text you must "mark" the block by pressing [BLOCK] twice - once with the cursor on the first character of the block and once with the cursor one position past the last character of the block. (Which end of the block you mark first is unimportant.) Each time the status line will prompt you with:

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

Each time type "D" to delete the block of text.

After the first time you type [BLOCK]-D you will notice that the status line displays the 1 END message. This means you have marked "one end" of the block. After the second time you type [BLOCK]-D the status line will display the number of lines about to be deleted and ask for confirmation.

You can abort the delete-block operation anywhere along the way. You can abort the [BLOCK] menu prompt by pressing <CTRL-C>. If the 1 END message is still on the status line you can abort the operation by pressing [CANCEL]. Or you can type "N" to the confirmation prompt.

Deleting text by the procedure above cannot be undone. A later section "Moving and Copying Blocks of Text" describes how to delete blocks of text with the ability to restore the text.

DELETING TEXTOperationCommand Sequence

Delete character at left  
of cursor; shift  
following characters left

BACK  
SPACE

Delete character at  
cursor; shift following  
characters left

DELETE

Erase from cursor to  
end of line

ERASE  
EOL

Erase entire line cursor  
is on and close up text

ERASE  
LINE

Delete word to left  
of cursor

DEL  
PREVIOUS  
WORD

Delete word to right  
of cursor

DEL  
NEXT  
WORD

Delete paragraphs and  
blocks of text

1. Position cursor over first character  
in the paragraph to be deleted.

- 2.

BLOCK

Type "D"

3. Position cursor past last character in paragraph to be deleted.

4. 

|       |
|-------|
| BLOCK |
|-------|

 Type "D"

5. The status line will display number of lines being deleted and ask for confirmation. Type "Y" if OK to delete the block of text.

#### CORRECTING MISTAKES MADE TO A LINE

This command restores the current line to its previous contents before any changes you just made to it.

|      |
|------|
| UNDO |
|------|

Once you move the cursor off a line you can no longer "undo" that line to its previous contents.

WARNING! [UNDO] will delete a newly typed line if the line was empty when you first started typing on it.

FILE OPERATIONS

To perform any of the more common file operations available from Visual Mode press [FILE]. The status line will prompt:

**[E]xit [Q]uit [N]ew [S]ave [D]irectory**

- Exit Exits VEDIT PLUS and saves your file. This is the normal way to leave VEDIT PLUS.
- Quit Quits VEDIT PLUS and abandons your file. If an existing file was edited, it is left on disk unchanged. This is used when you DO NOT WANT TO SAVE your edited text on disk.
- New Allows you to edit a new file. If you are already editing a file it prompts you whether you want to save or abandon the current file.
- Save Saves the file on disk and lets you continue to edit it. This should be used regularly if you are editing for a long time to protect yourself against power and hardware failures or a major mistake.
- Dirac. Displays the directory of any desired drive or subdirectory. Allows use of wildcard characters, i.e. "\*.TXT".

Selecting "Quit" will give the prompt:

**Abandoning File -- [S]tay in VEDIT [E]xit VEDIT [N]o - Resume**

This is a confirmation that you really want to abandon your file, and allows you to stay in VEDIT PLUS or go back to the operating system. Type "N" if you do not want to abandon your file and want to continue editing it. Type "S" to stay in VEDIT PLUS or "E" to exit back to the operating system. (Note: If you are simultaneously editing several files, the "E" exits only the current file and stays in VEDIT PLUS with another file now active; when only one file remains, it exits to the operating system.)

If you are currently editing a file when you select "New" you will get the prompt:

**OK to Save Current File? [Y]es [N]o - Abandon File**

The normal response is "Y" to save the current file. Typing "N" abandons the file just as "Quit" from the [FILE] menu will. The prompt "Abandon File Y/N" asks for confirmation if you want to abandon the file.

### The "FILE" Message

The "FILE" message on the status line is followed by the name of the file being edited. When the file is small enough for all of it to fit into memory at one time (typically 52K or smaller), the "FILE" message will appear in capital letters. However, when editing a larger file, some or all of the letters will be in lower case, corresponding to which part of the file is currently in memory.

When you are at the beginning of a large file the message will read "File". Since the "beginning" of the message is capitalized, the beginning of the file is in memory; the end of the file is not in memory. When you are at the end of a large file the message will read "fILE". Since the "end" of the message is capitalized, the end of the file is in memory; the beginning of the file is no longer in memory. When you are in the middle of a large file, with neither the beginning nor the end in memory, the message reads "file".

Although VEDIT PLUS normally handles files larger than memory automatically for you, the exact form of the "FILE" message can help with some editing operations. For example, it may help you decide whether you need to select the "global" option when searching for text (described later).

EXITING VEDIT PLUS

Purpose: When you are done editing you want to SAVE YOUR FILE and exit back to the operating system.

This is the normal way to save your file and exit VEDIT PLUS.

1. 

|      |
|------|
| FILE |
|------|

The status line prompts:

```
[E]xit [Q]uit [N]ew [S]ave [D]irectory
```
2. Type "E" to Exit. 

The file you were editing will be saved to disk and you will be returned to the operating system

NOTES:

1. If you are done editing the current file, but want to edit another file see the following section "Editing A New File".
2. If you are simultaneously editing several files (described later) the [FILE]-Exit function will save the file, but not yet exit VEDIT PLUS. Instead it will exit to another "edit buffer" containing another file.

QUITTING VEDIT PLUS

Purpose: When you DO NOT WANT TO SAVE THE EDITED TEXT to disk you can "Quit" (also called "Abandon") the current file. If you quit a file which already existed on disk, the file simply remains on disk unchanged - only the edit changes you made are abandoned.

It is common to load a file into VEDIT PLUS just to look at it or to copy parts of it into text registers. When you are done looking at the file it is then better to "Quit" the file. This way you can be sure that no edit changes accidentally got into the file.

When you "Quit" you have the choice of staying in VEDIT PLUS to edit another file or of returning to the operating system.

Quit VEDIT PLUS - Return to Operating System

1.



The status line prompts:

**[E]xit [Q]uit [N]ew [S]ave [D]irectory**

2. Type "Q" to Quit.

The status line prompts:

**Abandoning File -- [S]tay in Window [E]xit Window [N]o - Resume**

3. Type "E".

Any edit changes you made are lost and you are returned to the operating system.

If you DO NOT WANT TO QUIT, type "N" to resume editing.



Abandon File - Stay in VEDIT PLUS

1.



The status line prompts:

**[E]xit [Q]uit [N]ew [S]ave [D]irectory**

2. Type "Q" to Quit.

The status line prompts:

**Abandoning File -- [S]tay in Window [E]xit Window [N]o - Resume**

3. Type "S".

Any edit changes you made are lost, but you remain in VEDIT PLUS. Typically you would then use [FILE]-New to edit a new file.

If you DO NOT WANT TO ABANDON your file, type N" to resume editing.

See Notes for "Exiting VEDIT PLUS".

EDITING A NEW FILE

Purpose: It is not necessary to exit and invoke VEDIT PLUS again from the operating system in order to edit another file.

This procedure is for editing one file after another. If you want to edit two (or more) files at the same time see the later topic "Simultaneously Editing Two Files".

1. 

|      |
|------|
| FILE |
|------|

The status line prompts:

**[E]xit [Q]uit [N]ew [S]ave [D]irectory**

2. Type "N" to edit a new file.

If you are currently editing a file, the status line prompts:

**OK to Save Current File? [Y]es [N]o - Abandon File**

3. Type "Y" to save the current file to disk.

Type "N" to abandon the file and NOT SAVE IT.

You will be prompted for confirmation to abandon the file.

The status line will prompt for the name of the new file you want to edit:

**Enter Filename:**

4. Enter the desired file-name in the normal format and a RETURN. This can be an existing file or a file to be created.

SAVE EDITED TEXT AND CONTINUE

Purpose: You should make it a habit to regularly save your text on disk during a long "*edit session*". This way you will lose less work in case of a power or hardware failure, in case you make a major mistake, or if someone accidentally turns off the computer. Saving the text every hour and whenever you leave the computer is suggested.



FILE

The status line prompts:

**[E]xit [Q]uit [N]ew [S]ave [D]irectory**

1. Type "S" to "Save" your text to disk.

After the text is saved you can continue editing with everything else unchanged.

DISK DIRECTORY DISPLAY

Purpose: The disk directory of any drive or MS-DOS subdirectory can be displayed. This is useful when editing, merging or splitting multiple files. In the event that you run out of disk space, you can then also see if any files can be deleted.

1.



The status line prompts:

**[E]xit [Q]uit [N]ew [S]ave [D]irectory**

2. Type "D".

The status line prompts:

**Enter Filename:**

3. You now have several options. For example:

Press RETURN.

This displays the directory of the current drive and subdirectory.

Type B: and RETURN.

This display the directory of drive B. You can select any drive you want.

Type A:\*.TXT and RETURN.

The "\*" is a *wildcard* which displays all files with extension ".TXT" on drive A.

Type A:\BIN\ and RETURN.

This displays all files in the "\BIN" subdirectory of drive A.

NOTE: If you want to see the directory exactly as MS-DOS (PCDOS) displays it with all file sizes and creation dates you can use the Command Mode "OC" command to execute the DOS "DIR" command from within VEDIT PLUS.

MOVING AND COPYING BLOCKS OF TEXT

The 36 "Text Registers" and the [BLOCK] function allow very flexible "cut and paste" type operations. You can copy or move a block of text from one part of a file to another, repeatedly insert a piece of text, delete a block of text and much more.

The text registers are regions in memory in which VEDIT PLUS stores text which is independent of the main text you are editing. There are 36 text registers named "0" through "9" and "A" through "Z".

A "block" of text is any amount from one character to an entire file. You can COPY a block of text to a text register, which leaves your main text unaltered or you can MOVE a block of text to a text register, which also deletes it from your main text. Once the text is in a text register, you can then insert it anywhere in your file. Since this does not alter the text register, you can repeatedly insert it into the file.

To copy/move a block of text you must mark its exact beginning and end by pressing [BLOCK] twice - once with the cursor on the first character of the block and once with the cursor one position past the last character of the block. (If you want to include the end of a line in the block, the cursor must be at the first character of the next line.) Each time the status line will prompt with:

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

Type "C" to copy or "M" to move the block of text.

The status line then prompts for which text register you want to copy/move to:

**REGISTER [+], 0-9 A-Z**

Type a character "0" through "9" or "A" through "Z". As a convenience, you can press RETURN, [BLOCK] or any other function key to specify text register "0". This makes it easier when you are only using one text register.

Notice that the message **TEXT** now appears on the status line to remind you that you have something in a text register.

To insert a text register, move the cursor to the position at which you want the insertion made. Press [BLOCK]:

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

Type "I" to insert a text register. At the register prompt type the character corresponding to which register you want inserted.

Moving and Copying Blocks of Text (Reference)

Marking a block of text is the same as was done for "Deleting a Block of Text". It does not matter which end of the block you mark first. The status line displays **1 END** when you have only marked one end of the block and have yet to mark the second end.

The copy/move operations can also append the block of text to any existing text in the register. At the register prompt type "+" before the register number.

Although you can delete a block of text with the "Delete" option, it is safer to delete text by moving it to a text register. In case you deleted too much you can then insert it right back again.

If you type **[BLOCK]-C** or **[BLOCK]-M** twice at the same location, the specified text register will be emptied. If all registers are empty, the **TEXT** message will disappear from the status line.

In practice it doesn't matter whether you type **[BLOCK]-C**, **[BLOCK]-M**, **[BLOCK]-D** or even **[PRINT]-B** when marking the first end of a block. Only when marking the second end must you select the desired function.

If insufficient memory space exists for the copy/move operation, no operation is performed and the **FULL** message appears on the status line. (Moving the cursor to a different line erases the **FULL** message.) After emptying any unused text registers you can try the operation again.

If there is insufficient memory space in the edit buffer to insert a text register, nothing will be inserted and the **FULL** message will appear. The topic "Freeing More Memory Space" explains how to free more memory space in the edit buffer.

If you forget where you set the first block marker press **[BLOCK]** and type **S** to select "Swap". This swaps the position of the marker with the cursor to show you where the marker is set. Type **[BLOCK]-S** again to restore the marker and cursor position.

If you forget what is in a text register, you can use the Command Mode **"RT"** command to display it for you.

If you make a mistake and see the incorrect status line prompt, press **<CTRL-C>** to cancel the function. If the status line still reads **1 END** and you want to reset that first marker, press **[CANCEL]**.

MOVING TEXT WITHIN THE FILE

1. Position cursor at first character in block to be moved.

2. 

|       |
|-------|
| BLOCK |
|-------|

Type "M" to select "Move".  
Message **1 END** appears on status line

3. Position cursor one position past last character in block to be moved.

4. 

|       |
|-------|
| BLOCK |
|-------|

Type "M" to select "Move".  
(Or type "C" to copy a block.)  
Status line prompts with:  
**REGISTER [+] 0-9 A-Z**

5. Type the register name "0" - "9" or "A" - "Z" of desired register. Or press RETURN or [BLOCK] to use register "0".

Type optional + before name if new text is to be appended to any existing text in register, instead of overwriting it.

6. Move cursor to position at which to insert text.

7. 

|       |
|-------|
| BLOCK |
|-------|

Type "I" to select "INSERT".  
Status line prompts with:  
**REGISTER [+] 0-9 A-Z**

8. Type the same register name as you did in step 5 above.



See NOTES on next page.

NOTES:

1. If you get a **FULL** message at step 4, there is insufficient memory for the Text Register to hold the entire text block. Nothing was moved/copied to the Text Register. Try emptying other text registers. If you get a **FULL** message at step 7, your edit buffer does not have enough memory space. See task "Making More Memory Space".
2. Following the text insert in step 7, the cursor is positioned at either the beginning or end of the inserted text depending upon the Command Mode "ES 4" switch.
3. In step 3, in order to include the <CR>-<LF> at the end of a line, position the cursor at the beginning of the next line.
4. Alternatively, you can reverse steps 1 and 3, i.e. either end of the block may be set first.
5. Any function which is prompting you on the status line can be canceled by pressing <CTRL-C>. You can cancel the first block marker (i.e. the **1 END** message) by pressing [CANCEL].

EMPTYING A TEXT REGISTER

Purpose: It is best to empty a text register when its contents are no longer needed. This frees up more memory space.

1.  Type "C" or "M".
2.  Type "C" or "M".  
Performing a move or copy with the cursor at the same position empties the register.
3. Type the register name "0"- "9" or "A"- "Z" of the register to be emptied.



THE [REPEAT] FUNCTION

It is often desirable to repeat an edit operation such as inserting the same character, deleting many lines, or moving the cursor many "pages" forwards or backwards. With [REPEAT] you can perform these repeated operations without having to type the same key over and over again. Pressing [REPEAT] once gives a "repeat value" of "4", which is displayed on the left side of the status line. Pressing it again multiplies the value to "16", then to "64" and finally "256". Any other repeat value may be selected by typing a number between 00 and 256. Once the repeat value is correct, simply press the desired character or function/control key which is to be repeated.

For example, to create the top of a box consisting of 50 "\*", type [REPEAT] and "50". At this point the status line will display: (notice the "50" on the left hand side)

```
50 LINE: 1 COL: 1 FILE: SAMPLE .TXT
|
```

Then type a single "\*". The screen will change to:

```
LINE: 1 COL: 51 FILE: SAMPLE .TXT
|*****
```

[REPEAT] Function (Reference)

With VEDIT PLUS's interruptable screen updating, only the final screen will be shown when using [REPEAT]. Since some operations such as deleting a line may take a second to perform on a very large file, you may notice some delay when using [REPEAT]. (If you are deleting many lines, it may be quicker to delete them as a "block" of text.)

To abort a function while it is being repeated press [CANCEL].

REPEATING OPERATIONS

| <u>Operation</u>                | <u>Command Sequence</u> |            |                  |
|---------------------------------|-------------------------|------------|------------------|
| Delete four lines of text       | REPEAT                  | ERASE LINE |                  |
| Page forward by 16 pages        | REPEAT                  | REPEAT     | PAGE DOWN        |
| Delete 30 characters            | REPEAT                  | 3 0        | DELETE           |
| Insert 20 blank lines           | REPEAT                  | 2 0        | RETURN           |
| Insert 40 "*" characters        | REPEAT                  | 4 0        | *                |
| Reformat the next 10 paragraphs | REPEAT                  | 0          | FORMAT PARAGRAPH |

NOTE: You can cancel the [REPEAT] function while it is prompting you on the status line for the repeat value by typing "000" and any character, or by pressing [CANCEL].

SEARCH AND REPLACE

[FIND] is used to search for text, such as a particular word. [REPLACE] is used to first search for text and when found replace it with other text. You can automatically replace all occurrences in your document or for each occurrence be prompted whether you want it replaced.

To search for some desired text press [FIND]. The status line will prompt:

**FIND?**

Enter the desired "*search string*" followed by a RETURN. If found, the cursor will be positioned just past the first occurrence of the desired text. Each time you press [FIND] afterwards, the cursor will move to the next occurrence of the same text.

To search for something different first press [CANCEL], which cancels the current "*search string*". Then press [FIND] again and you will be prompted for a new search string.

If it cannot find another (or any) occurrence of the desired text, the status line will display:

**CANNOT FIND *search-string***      Type any key to continue.

This also cancels the search string, i.e. you don't have to press [CANCEL] to search for another.

[REPLACE] first prompts with **FIND?** similar to [FIND]. It then prompts:

**REPLACE WITH?**

Enter the desired "*replace string*" followed by a RETURN. VEDIT PLUS will search for the next occurrence of the desired text and when found prompt with:

**REPLACE? [Y]es [N]o [R]est [C]ancel**

Type "Y" to replace the text with the new "*replace string*", or type "N" to leave it unchanged. Type "R" to replace this and all subsequent occurrences without prompting for each one. As with [FIND], each time you press [REPLACE] it will locate the next occurrence of the string and prompt with the replace options. To cancel this process choose "C" from the options. Alternatively, to search and replace for different strings, press [CANCEL] before pressing [REPLACE].

Often you will want to search for all occurrences of a string and selectively replace some of them. This is done in conjunction with [REPEAT]. Press [REPEAT] four times for a maximum count of 256. Then press [REPLACE] and answer all prompts as above. Now the REPLACE? [Y]es [N]o [R]est [C]ancel prompt will repeat up to 256 times allowing you to make the selective replace. As before, "R" will replace the remaining occurrences without prompting, and "C" will cancel the replace operation.

When using [FIND] or [REPLACE] you can select from the "search options" by pressing RETURN in response to the FIND? prompt. The status line will then prompt:

**OPTIONS: [A]gain [B]egin [G]lobal [R]everse**

Select one or more options by typing the corresponding letters "A", "B", "G" and "R", followed by RETURN. For example, entering "AB <RETURN>" will select the "Again" and "Begin" options.

- Again This option causes the previous search/replace strings to be used again. This is convenient after you receive a "CANNOT FIND" message and want to reuse the strings on a new file or from the beginning of the file (with the "Begin" option).
- Begin This option starts the search or replace from the beginning of the edit buffer. If the "Global" option is also selected, the search or replace starts at the very beginning of the file.
- Global This option causes the search or replace to operate to the end of the file, if necessary, not just the end of the edit buffer. If the entire file is in the edit buffer, this option has no effect.
- Reverse This option causes the search or replace to go in the reverse direction, i.e. toward the beginning of the file. This is convenient for finding previous occurrences. Do not use this with the "Begin" option.

After you enter the desired options and RETURN you will get the FIND? prompt again, unless you selected the "Again" option, in which case the [FIND] or [REPLACE] will immediately be performed with the previous search/replace strings.

Search and Replace (Reference)

The search and replace strings are limited in length to 80 characters. If you need to search for a RETURN press <CTRL-N> in its place. If you need to search for a control character, which is used for line editing, you must precede it with <CTRL-Q>, which is the "next character literal" in command lines.

VEDIT PLUS is normally configured to equate upper and lower case letters when searching. Therefore, a search for "the" will find "the", "The" and "THE". To make a distinction between upper and lower case letters, first give the command "ES 5 0" from Command Mode. The **FIND?** prompt indicates when upper and lower case letters are equated by converting lower case letters to upper case as you type them. Note that the replace string must be entered verbatim - lower case letters are never converted to upper case, nor are they matched to the case of the original text.

All of the pattern matching available in Command Mode is also available in Visual Mode.

You can use a "trick" to find a capitalized word without bothering with the "ES 5 0" command - use the pattern "|U" in place of the upper case letter. The drawback is that it won't always work, especially for short words. For example, to search for the capitalized word "Tutorial" enter the following string after the **FIND?** prompt:

```
|Uutorial          Search string to find the capitalized
                    word "Tutorial".
```

SEARCHINGTo search for next occurrence:

1. 

|      |
|------|
| FIND |
|------|

      Status line prompts:  
**FIND?**
  
2. Type up to 30 characters that you are searching for and a RETURN. Pattern matching codes are allowed. Use <CTRL-N> to search for RETURN.      If found, cursor will be past desired text. Gives an error if not found - You must then press any key to continue.
  
3. 

|      |
|------|
| FIND |
|------|

      Press [FIND] again to search for the next occurrence. To find the "nth" occurrence use:  
**[REPEAT] n [FIND]**
  
4. 

|        |
|--------|
| CANCEL |
|--------|

      Allows another string to be searched. Automatically cancels if string not found.

To search for previous occurrence:

1. 

|      |
|------|
| FIND |
|------|

      Status line prompts:  
**FIND?**
  
2. Type immediate RETURN.      Status line prompts:  
**OPTIONS: [A]gain [B]egin [G]lobal [R]everse**
  
3. Type "R" for "Reverse" and RETURN      Alternatively, you can select other options.  
  
Status line prompts with **FIND?** again. Continue with step 2 at the top of the page.

REPLACING

1. 

REPLACE

Status line prompts:  
**FIND?**
  
2. Type up to 30 characters that you are searching for and a RETURN. Pattern matching codes are allowed. Use <CTRL-N> to search for RETURN.

Status line now prompts:  
**REPLACE WITH?**
  
3. Type up to 30 characters which are the replacement string and a RETURN.

If found, cursor will be past desired text. Gives an error if not found - You must then press any key to continue.

Status line now prompts:  
**REPLACE? [Y]es [N]o [R]est [C]ancel**
  
4. Type "Y" to make the replacement, "N" to leave text unchanged, "R" to replace this and all following occurrences, or "C" to cancel [REPLACE].
  
5. 

REPLACE

Press [REPLACE] again to search and prompt for the next replacement. To prompt for "n" replacements use:  
**[REPEAT] n [REPLACE]**  
in step 1.
  
6. 

CANCEL

Allows another string to be replaced. Automatically cancels if string not found.

PRINTING TEXT

Not only can you print the entire document from within VEDIT PLUS, but also any portion of it from a few words to a few pages. When VEDIT PLUS prints, it uses a left margin called the "*printer margin*" to keep the text from printing on the very left edge of the paper. It also skips over page perforations and leaves a top and bottom margin on each page. These margins are adjustable.

Press [PRINT]. The status line will prompt:

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

- [A]ll Prints the entire document. 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 of each page.
- [B]lock Prints a block of text. The block is specified similar to "[BLOCK]-Copy".
- [E]ject Advances the printer to the top of the next page. (Depending upon the "PP" command parameter, either line feeds or a form feed character is 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 changed 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.

The procedure to print a portion or "block" of text is very similar to moving/copying or deleting a block of text. Mark the block by pressing [PRINT] and "B" at each end. The text will then be printed.

To abort the printing before it is done, press <CTRL-C>.

VEDIT PLUS, as supplied, is set for a "*printer margin*" of 12 columns, which typically leaves a one inch left margin, and is set to print 60 lines on each page, leaving three lines blank at the top and bottom of each page.

To change the printer margin press [PRINT] and type "M" for "Margin". The status line will prompt:



**Enter Printer Margin**

Enter the desired printer margin, typically between 0 and 30, and a RETURN.

To change the number of text lines per page press [PRINT] and type "L" for "Lines". The status line will prompt:

**Enter # Lines**

Enter the desired number of lines, typically between 50 and 66, and a RETURN. The text will automatically be centered top to bottom on the page.

**Printing Text** (Reference)

The number of lines printed per page is related to the physical size of a page, assumed above to be 66 lines. This is the correct value for letter size sheets and regular computer paper. The value for the physical page size can be changed during installation or with the Command Mode "PP" command. You must change it when printing legal size sheets, European sized paper or pre-printed forms. You may want to change it when printing labels.

VEDIT PLUS can advance to a new page by sending multiple line-feeds or by sending a "*form-feed*" character. You can select this option during installation or with the "PP" command.

PRINTING TEXT

Printing Entire Document:

1. Be certain your printer is "on line" and/or selected.  
(See your printer manual).

2. 

|       |
|-------|
| PRINT |
|-------|

Type "A" to print entire document.

To abort printing before it is done press <CTRL-C>.

Printing Portion Of Document:

1. Be sure your printer is "on line".
2. Position cursor at first character to print.

3. 

|       |
|-------|
| PRINT |
|-------|

Type "B" to select Block print.

4. Position cursor past last character to print.

5. 

|       |
|-------|
| PRINT |
|-------|

Type "B" again.  
Printing should now start.

To abort printing before it is done press <CTRL-C>.

DISPLAYING AND ENTERING CONTROL CHARACTERS

Although most text files do not contain any control characters (other than <CR>, <LF> and <TAB>), you may occasionally encounter a file which does. Most common are special control codes to turn printer functions on and off. Many printers use control characters or escape sequences to control such things as character size, font style and overstrike.

Any control characters in the text, other than <CR>, <LF> and <TAB> are displayed on the screen in the common format by preceding the letter with a "caret" (^).

The function [NEXT CHAR LITERAL] allows any control character except <CTRL-Z> (which is not allowed by MS-DOS or CP/M) to be inserted into the text.

For example, to insert a <CTRL-C>, press [NEXT CHAR LITERAL]. Then press <CTRL-C>. A "^C" is now displayed just before the cursor. Notice that the cursor moves over the "^C" as a single character, similar to the way the cursor moves over tab characters.

On the IBM PC, graphics characters can be entered directly, without needing [NEXT CHAR LITERAL], by using the <ALT>-keypad feature. For example, to enter the graphics character with value 132 (German "a-umlaut"), hold down the <ALT> key, type 132 on the keypad, and release the <ALT> key.

Control Characters (Reference)

[NEXT CHAR LITERAL] lets you enter any character into the text that you can generate from the keyboard. Only <CTRL-Z> cannot be entered because it is used by MS-DOS and CP/M to mark the end of a file.

Control characters can also be inserted from Command Mode with the "EI" command. The "EI" command has the advantage of being able to insert any character with a decimal value between 00 and 255 (except 26 which is a <CTRL-Z>) into the text. In particular, lets you insert <CTRL-@> with value 00, which the IBM PC keyboard cannot generate.

If you are repeatedly inserting complex sequences of control characters, it is easier to copy the sequences into text registers and then just insert the appropriate text register when needed.

If you are doing a lot with printer control characters, you may find it easier to use our V-PRINT print formatter.

ENTERING CONTROL CHARACTERS INTO THE TEXT

Purpose: Sometimes you will need "control characters" in your text for special purposes such as controlling the print size and character style on your printer. Control characters typed on the keyboard usually perform edit operations or are ignored.

1.

|                         |
|-------------------------|
| NEXT<br>CHAR<br>LITERAL |
|-------------------------|

This enters the next character into the text, even a control character. You probably want to be in INSERT mode at this time.

2. Type the control character which is to be entered into the text at the cursor position.

ENTERING GRAPHICS CHARACTERS (IBM PC Only)

Purpose: The IBM PC graphics characters are often used for drawing boxes within your text.

Example: You can easily create the following box top with a 50 character width.



First type "<ALT>-201" (hold down the <ALT> key and type 201 on the keypad). Then press [REPEAT], type "48" and "<ALT>-205". Last type "<ALT>-187".

INDENTING TEXT (LEFT MARGIN)

There are several ways to indent text so that it doesn't begin in the first column. You can, of course, type spaces at the beginning of each line to be indented. This is the normal way of indenting the first line of a new paragraph. However, VEDIT PLUS can also automatically indent each new line of text for you. This is useful in word processing for indenting entire paragraphs, and for editing programs written in structured languages such as C, Pascal and PL/I.

The "*indent margin*" determines the left margin for newly entered lines. Normally the indent margin is at column 1. The [INDENT] function moves the indent margin to the right and [UNDENT] moves the margin back to the left. Each time you press [INDENT] or [UNDENT], the indent margin changes by the "*indent increment*" which is normally "4", but is configurable.

Try it. Press RETURN to position the cursor on a new line. Now press [INDENT]. Notice how the cursor moved to column 5. Now type a couple of words and another RETURN. This time the cursor started the new line in column 5. Press [INDENT] again; the cursor will now be in column 9. Press [UNDENT]; the cursor will move back to column 5.

The following section "Formatting Paragraphs" explains how to use the [FORMAT PARAGRAPH] function to change the indentation of an existing paragraph.

VEDIT PLUS has another mode of automatic indentation called "*Auto-Indent*". It is primarily intended for editing structured programming languages such as "C". In Auto-Indent mode each new line will be indented the same amount as the previous text line. You can then change the indentation of the new line by pressing [INDENT] or [UNDENT] - to start or end a block of program instructions. The main advantage of Auto-Indent mode is that you can jump around in a program and newly entered instructions will automatically fit the indentation of the current block of instructions.

To change either the "*indent increment*" or the "*Auto-Indent*" mode press [USER]. The status line will prompt:

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

To change the "*indent increment*" type "I" and the prompt will change to:

**Enter Indent Increment:**

Enter the number, generally between 1 and 10 and a RETURN.

To change the "Auto-Indent" mode type "A" and the prompt will change to:

**Auto-Indent -- (0 = Off) (1 = On):**

Enter "1" to enable Auto-Indent or a "0" to disable it.

Indenting Text (Reference)

VEDIT PLUS performs automatic indenting by padding the beginning of lines with tabs and spaces. You can confirm this by moving the cursor over these leading tabs and spaces and, if you like, you can delete them or over-write them. VEDIT PLUS only performs automatic indenting when you press RETURN, when "Word Wrap" is being used and when paragraphs are formatted.

If the cursor is at the beginning of a line when [INDENT] or [UNDENT] is pressed, the cursor and any following text will move to the new indent margin. Otherwise, only the indent margin is changed which will affect the next new line; the current line is not changed.

The exact number of tabs and spaces used to pad indented lines is related to the currently set tab positions. For example, assume that the "indent increment" is set to the common value of four (4) and the tab positions at every eight, i.e at 9, 17, 25, etc. When the "indent margin" is nine, the padding will consist of one tab; when the "indent margin" is 21 the padding will consist of two tabs and four spaces. On the other hand, if the tab positions were set to every four, only tabs would be used in the padding. Note that if the "Expand Tab with Spaces" switch is set, only spaces will be used for padding. This will use up additional memory and disk space.

INDENTING TEXTOperationCommand Sequence

Move the indent margin  
to the right

INDENT

New text lines will now  
be indented further to  
the right.

For example, with the  
cursor on a new line,  
the first [INDENT]  
moves the cursor to  
column 5, and all fol-  
lowing new lines will  
also start in column 5.

Move the indent margin  
back to the left

UNDENT

To change Indent/  
Undent increment

1. 

USER

Then type "I".

2. Enter the desired indent increment,  
generally between 1 and 10, followed by  
a RETURN.

To select Auto-  
Indent mode

1. 

USER

Then type "A".

2. Enter "1" and a RETURN to enable.  
Enter "0" and a RETURN to disable.

NOTE: You shouldn't indent your text to keep it from printing on the left edge of the paper. The print functions allow you to set a separate *left printer margin* for this purpose.

WORD PROCESSING FUNCTIONS

In addition to its wide range of features for general purpose editing, VEDIT PLUS has features specifically designed to assist with word processing. These include:

- Cursor Movement    The cursor may be moved forwards and backwards by a word or paragraph at a time.
- Indenting            Single lines or larger sections of text can be indented so that they don't start in the left-most column. This can be thought of as changing the left margin.
- Word Wrap            Enabling word wrap sets a right margin beyond which no text should appear. Words which would exceed the right margin are instead "*wrapped*" to the next line without breaking the word in half.
- Format Paragraph    If desired, each paragraph can have a different left and right margin. The margins for an existing paragraph can also be changed and the paragraph then "re-formatted" to fit the new margins.
- Justification        Notice how the paragraphs in this manual have a completely straight or "justified" right margin. Justification produces the straight margin by adding spaces between words on each line.

VEDIT PLUS has many other features useful to word processing such as printing, moving blocks of text and search and replace.

Definition of "Word" and "Paragraph" (Reference)

Words are allowed to have embedded periods in them, as in "i.e.". A comma "," always ends a word, even if the comma is not followed by a space. As a special case, numbers with embedded commas, such as "10,000" are treated as one word. The special characters ")", "]" and "}" also separate words from each other, as do spaces, tabs and carriage returns. All other characters are allowed in words.

VEDIT PLUS considers a paragraph to end when an empty line, a blank line or a print formatter command line is encountered. Print formatter command lines are considered to be any lines which begin with a ".", "!" or "@" in the first column.



WORD WRAP (RIGHT MARGIN)

To simplify word processing, "word wrap" may be enabled. Word wrap automatically starts a new line when text reaches the right margin which you have set. The entire word which would otherwise exceed the right margin is *wrapped* to the next line, without breaking the word in half.

Ordinarily word wrap is disabled. To enable it press [USER]. The status line will prompt:

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

To change the *Word Wrap right margin* type "W" and the prompt will change to:

**Enter Word Wrap Margin (0 = Off):**

Enter the column number for the desired right margin, generally between 50 and 80, and a RETURN.

A right margin of zero turns word wrap off. Word wrap should be turned off when editing programs. It is also usually easier to enter tabular data with word wrap off.

Word wrap may be thought of as setting the right margin. The left margin is set with [INDENT] and [UNDEMENT].

As you edit a paragraph, existing text will sometimes extend past the right margin. This is normal, because word wrap only occurs with new text entered past the right margin. You can get all the text back between the margins with [FORMAT PARAGRAPH].

You can set different margins when entering different paragraphs. You can also change the margins and then *reformat* an existing paragraph to the new margins. To reformat a paragraph, set the right (word wrap) margin with [USER], the left margin with [INDENT] and [UNDEMENT], and press [FORMAT PARAGRAPH].

Word Wrap (Reference)

The default value for the word wrap margin is set during installation (Task 5.7). Our preconfigured versions all have word wrap disabled (value of zero). The right margin for word wrap can also be changed from Command Mode with the "EP 7" command.

The right margin can be greater than the screen line length, in which case VEDIT PLUS will either scroll the screen horizontally or display a continuation line before the word wrap takes place.

WORD WRAP (RIGHT MARGIN)

NOTE: Before word wrap or [FORMAT PARAGRAPH] will work, the right margin for word wrap must be set. A right margin value of "0" turns word wrap off and disables [FORMAT PARAGRAPH]. The left margin is set with [INDENT] and [UNDEMENT].

- 1. Set right margin for word wrap at desired column. This is the last column in which text may appear.
 

|      |
|------|
| USER |
|------|

Then type "W".

At prompt enter new margin value and a RETURN.

- 2. Move left margin to the right.
 

|        |
|--------|
| INDENT |
|--------|

- 3. Move left margin to the left.
 

|          |
|----------|
| UNDEMENT |
|----------|

- 4. Set additional indent for the beginning of a paragraph with spaces.
 

|              |
|--------------|
| SPACE<br>BAR |
|--------------|

-- OR --

- Use a tab. See "ET" command to change tab positions.
 

|             |
|-------------|
| TAB<br>CHAR |
|-------------|

- 5. Type in the paragraph. Words will be wrapped as needed.

- 6. Due to further edit changes, some text may fall outside the margins.
 

|                     |
|---------------------|
| FORMAT<br>PARAGRAPH |
|---------------------|

This will reformat paragraph to fit current margins.

FORMATTING AND JUSTIFYING PARAGRAPHS

[FORMAT PARAGRAPH] is a word processing function which formats a paragraph so that all of the text appears between selected left and right margins. The left margin is set using [INDENT] and [UNDENT], while the right margin is the same as used for word wrap.

You can select whether you want paragraphs to have a straight right margin, as in this manual, or a ragged right margin. Justification creates a straight right margin by adding spaces between the words on the line.

[FORMAT PARAGRAPH] is usually used to make a modified paragraph fit the margins, or make existing paragraphs fit new margins. To format a paragraph, place the cursor anywhere in the paragraph and press [FORMAT PARAGRAPH].

After formatting, the cursor will be positioned at the beginning of the next paragraph. A series of paragraphs may therefore be formatted by just repeatedly pressing [FORMAT PARAGRAPH].

To enable/disable justification press [USER]. The status line will prompt:

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

Type "J" and the prompt will change to:

**Justify -- (0 = Off) (1 = On) (2 = Unjustify):**

Enter "1" to enable justification or "0" to disable it. (The "unjustify" option is described below.)

IMPORTANT NOTE:

An indented line is not enough to start a new paragraph.  
A new paragraph must be started with a blank line or, if you are using V-PRINT, a formatter command line such as ".br" or ".sp". Otherwise [FORMAT PARAGRAPH] will combine several paragraphs into one!

Formatting And Justifying Paragraphs (Reference)

[FORMAT PARAGRAPH] preserves the typical indentation of the first line of a paragraph. If the second line of a paragraph is indented more than the first line, VEDIT PLUS considers it an "*offset paragraph*" and also preserves this offset. Numbered paragraphs are often offset paragraphs as in the following example:

1. This is an example of an *offset paragraph*. Notice how the first line begins in column 1, while all following lines begin in column 5. This paragraph can be entered by pressing [INDENT] anywhere along the first line.

However, any indentation which all lines of a paragraph have is ignored by [FORMAT PARAGRAPH]. The left most line(s) will be positioned at the current left margin.

If you are using print formatter command lines, you should note that these lines separate paragraphs from each other and will not be formatted or changed in any way by [FORMAT PARAGRAPH].

NOTE: When using word wrap or [FORMAT PARAGRAPH], you may notice that each text line ends in a space. This actually is the space you typed between the words. Lines ending in a period "." are typically followed by two spaces. These spaces at the end of a line are allowed to exceed the right margin.

If you inadvertently set the left margin greater than the right margin, word wrap and [FORMAT PARAGRAPH] will not operate.

If you need to edit text after it has been justified, it is easier if you first "*unjustify*" the text. This removes the additional spaces between words, leaving the right margin ragged. (It will leave two spaces following ".", "?", ":", and "!".)

A justified paragraph may be reformatted and justified between new margins by first changing the margins and then pressing [FORMAT PARAGRAPH]. You do not have to unjustify it first. Just disabling justification will not unjustify a paragraph, because [FORMAT PARAGRAPH] will reformat the paragraph leaving the additional spaces in place.

This unjustify feature can also be used when converting files created with other word processors for use with VEDIT PLUS. For example, a document justified by WordStar (tm) can be unjustified to make further editing easier. (Note: the High bits in WordStar files must first be stripped with the "YS" command - see User Guide "WordStar Files".)

FORMATTING AND JUSTIFYING PARAGRAPHS

1. Set right margin in same way as for word wrap.

USER

Type "W".

At prompt enter new right margin.

2. Set left margin if needed with [INDENT] and [UNDEMENT].

3. Format a paragraph to fit within the current left and right margins.

FORMAT  
PARAGRAPH

To Justify:

1. Enable justification.

USER

Type "J".

Enter "1" to enable.  
Enter "0" to disable.  
Enter "2" to unjustify.

2. Set right and left margins as above.

3. Format the paragraph to current left and right margins and justify the right margin.

FORMAT  
PARAGRAPH

KEYSTROKE MACROS

"Keystroke macros" let you assign a repeatedly typed sequence of keys to an unused function or control key. Typing the single function/control key then performs the equivalent of typing the entire sequence of keys one after another - saving you time and reducing the chances for error.

For example, if you were repeatedly typing the phrase "attached and included herein by reference", you could define the key <CTRL-A> to type out the whole phrase each time it was pressed.

One simple use of keystroke macros is to set up "hot keys" for accessing functions within the menus. For example, (on an IBM PC) you can assign [BLOCK]-Copy to <ALT-C>, [BLOCK]-Move to <ALT-M>, and [BLOCK]-Insert to <ALT-I>.

Keystroke macros are created with the [DEFINE] function. The step by step example on the next page shows how to set up <ALT-C> to generate the two keystrokes [BLOCK]-C. Keystroke macros can also be used in Command Mode; this is described later.

Keystroke Macros (Reference)

Any desired number of keystroke macros can be built into VEDIT PLUS using Installation Task 3. Since key sequences can even be assigned to displayable characters, you have the freedom of completely changing the keyboard layout, even adopting a Dvorak layout. However, extensive re-assignments are best done during installation.

If you attempt to assign a key sequence to a function/control key which is already in use, you are prompted for confirmation to re-define the old use. Re-defining already used keys with [DEFINE] must be done with care. For example, if <F1> performs [CANCEL] and you assign something new to <F1>, you can no longer access [CANCEL] unless you remember to first assign [CANCEL] to another key.

The currently customized keyboard layout with all keystroke macros can be saved to disk with the "YK" command. A new keyboard layout can subsequently be loaded from disk with the "YL" command.

The [REPEAT] function can be used with keystroke macros to repeat the entire sequence of keystrokes. A [REPEAT] can also be part of the assigned "Key Sequence". The only limitation is that you cannot [REPEAT] a keystroke macro which itself contains a [REPEAT].

KEYSTROKE MACROS

This describes the exact steps necessary on an IBM PC to set up `<ALT-C>` to act as a "block key" for the "BLOCK" function copy.

1. 

|        |
|--------|
| DEFINE |
|--------|

The status line will prompt:

**Function/Control Key - [DEFINE] to end:**

2. Press `<ALT-C>` and `[DEFINE]` again.

The status line will prompt:

**Key sequence - [DEFINE] to end:**

3. Press `[BLOCK]` and `C`.

The status line will now display:

**Key sequence - [DEFINE] to end: \BL\C**

4. Finally, press `[DEFINE]` again to end the sequence.

`<ALT-C>` has now been defined to generate `[BLOCK]-C`.

NOTES:

1. The "Function/Control Key" can also be a sequence of keys as long as it begins with an unused function or control key.
2. If you make a mistake press `[BACKSPACE]` to cancel the function; you can then start over with step 1.
3. There is no absolute limit to how many keystroke macros can be built-in or added with `[DEFINE]`. Of course, keystroke macros take up memory space.

THE [MISC] FUNCTION

Four editing functions are available with the [MISC] menu-function:

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

"Match Parentheses" is primarily useful to programmers. With the cursor positioned at one of the eight characters {, }, [, ], <, >, (, ), it moves the cursor forward or backward to the matching pair. Since it supports "nested" characters, it is very helpful for finding syntactic errors in "C" and other programming languages.

"Upper case/lower case" switches the character at the cursor position from upper case to lower case and vice versa. It then moves the cursor to the next character.

If you use "Match Parenthesis" or "Uc/Lc" very often you will want to assign it to an unused function/control key using Keystroke macros. <ALT-M> and <ALT-U> are good choices. You can make these assignments "permanent" by using Installation Task 3.

"Insert" puts the editor into *Insert Mode*, regardless of what mode it was in. Similarly "Overstrike" puts the editor into *Overstrike mode*. Since [INSERT] already toggles between the modes, these are alternate functions for users who prefer a separate key for entering each mode.



SIMULTANEOUSLY EDITING TWO FILES

With VEDIT PLUS you can simultaneously edit two or more files. This can be done without "*windows*", where only one file is displayed on the screen at a time, or with windows, where the screen is split to display more than one file. We will first describe multiple file editing without windows.

When you first invoke VEDIT PLUS the actual editing of the file is done in the "*main edit buffer*". The contents of the edit buffer are what you see in Visual Mode. In addition to the main edit buffer you can also edit any of the 36 "*text registers*".

The first time you edit a text register, the text register is converted into an "*edit buffer*". Edit buffers are a little different from text registers - the main difference being that edit buffers can be used to edit a separate file. You can readily switch from one edit buffer to another. Once a text register is converted into an edit buffer it remains an edit buffer.

Editing multiple files is analogous to having VEDIT PLUS running on several computers at once - switching edit buffers moves your chair from one computer to another.

The edit buffer that you have switched to is called the "*current*" or "*active*" edit buffer. Except for the main edit buffer, the name of the current edit buffer is always displayed on the status line, i.e. "E0" - "E9" or "EA" - "EZ".

To switch to a different edit buffer press [WINDOW]. The status line will prompt:

```
[C]reate [D]elete [S]witch [Z]oom
```

Type "S" for "Switch" and the status line will prompt:

```
Enter Edit Buffer Name:
```

Enter the name of the edit buffer you want to switch to. The name of the main edit buffer is "@". As a convenience, any invalid buffer name or RETURN will also select the main edit buffer.

You may place text into a text register and convert it into an edit buffer in order to edit the text. However, more typically you will convert an empty text register into an edit buffer and then use it to edit an additional file. To edit a second file, first switch to an empty edit buffer. Then type [FILE]-New which will prompt for the name of the second file to edit. The name of the file being edited in the "*current*" edit buffer is always displayed on the status line.

Since it is easy to forget which text registers are being used as edit buffers, we suggest that you only use "0" through "9" as text registers and use "A" through "Z" as edit buffers. You can pick mnemonic letters.

### Exiting and Quitting Multiple Files

As already described, [FILE] can Exit or Quit VEDIT PLUS. These operations work a little differently when you are editing multiple files - they perform their operation on the current edit buffer and then automatically switch to another edit buffer. Only when no further edit buffers remain, do they exit VEDIT PLUS back to the operating system.

Exiting or quitting from an edit buffer also converts the edit buffer back to an empty text register. Only the main edit buffer will not convert into a text register.

Consider the example of where you are editing two files - one in the main edit buffer and one in edit buffer "D". You are currently in the main edit buffer and press [FILE]-Exit. The file will be saved and you will be switched to edit buffer D. You again press [FILE]-Exit to save the second file. Since only the main edit buffer now remains and it is empty, you are exited back to the operating system with the second file also saved.

You can save a few steps when exiting multiple files by switching to Command Mode and giving the "EXA" command. This exits all edit buffers (saving all files) and then returns to the operating system.

### Simultaneous Editing with Windows

A following topic describes how to split the screen into "windows". Windows can simultaneously display the multiple files that you are editing. However, it is not necessary to use windows to simultaneously edit several files. It is not always even advantageous to use windows since it is easier to edit a file over the full screen instead of just inside a window.

To display an edit buffer in its own window you need only to create a window with the same name as the edit buffer - the edit buffer will then automatically use that window. You can create the window before or after you create the edit buffer. For easier editing, you can also "zoom" a window to fill the full screen.

SIMULTANEOUSLY EDITING TWO FILES

This example gives the step by step directions to begin editing two files called "PROGRAM" and "DOCUMENT". It assumes that the files exist and that you have not invoked VEDIT PLUS yet.

## 1. VPLUS PROGRAM

Invoke VEDIT PLUS with the name of the first file. "PROGRAM" is loaded into the "main edit buffer", i.e. edit buffer "@".

Note: to edit with "windows", now follow the steps in the next topic "Creating Windows".

## 2.

WINDOW

Type "S" to switch to a new edit buffer. The status line will prompt:  
**Enter Edit Buffer Name:**

## 3. Type "D" to select text register D.

This converts text register "D" into an edit buffer.

There will now be no text on the screen and the status line will display "ED" to indicate that edit buffer "D" is "active".

## 4.

FILE

Type "N" to begin editing a new file. The status line will prompt:  
**Enter Filename:**

## 5. Enter "DOCUMENT" and RETURN.

The file "DOCUMENT" is loaded and displayed on the screen.

You can now edit the file "DOCUMENT" normally.

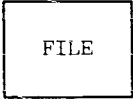
## 6.

WINDOW

Type "S" to switch back to the file PROGRAM in the main edit buffer. Press RETURN at the prompt for  
**Enter Edit Buffer Name:**

You can now edit the file PROGRAM normally. Use [WINDOW] to switch between files.

7.



Type "E" to "Exit" and save the file PROGRAM to disk.

By "exiting" the edit buffer you are automatically switched to the file DOCUMENT.

8.



Type "E" to "Exit" and save the file DOCUMENT to disk. Since no other edit buffers exist, this also exits back to the operating system.

COPYING TEXT FROM ONE FILE TO ANOTHER

Purpose: A common reason for editing two files is to copy portions, such as previously written paragraphs, from an existing file to a new file.


This example uses the two files DOCUMENT and PROGRAM as described on the previous page.


1. Follow steps 1 through 5 of the previous page. In this example, we will copy a block of text from DOCUMENT to the file PROGRAM.

2. Position the cursor to the beginning of the block to be copied. You are in the file DOCUMENT.


3.  Type "C" for "Copy".

4. Position the cursor past the end of the block of text.

5.  Type "C" for "Copy". Then press RETURN to copy the text into register "0".

6.  Use [WINDOW]-Switch to switch to the main edit buffer and the file PROGRAM.

7. Move cursor to position at which to insert text.

8.  Type "I" for "Insert". Then press RETURN to insert the contents of text register "0".

WINDOWS

The screen can be split into variable sized "*windows*" to display files in several edit buffers at the same time. The Command Mode can also be displayed in its own window(s).

To use a window, it must first be created. A new window is created by splitting an existing window into two windows. The split can be either vertical or horizontal. Each window is given a single character "*name*" when it is created. To display an edit buffer in its own window, the window must be given the same name as the edit buffer - the edit buffer will then automatically use that window. For example, to display the edit buffer "D" in its own window, you must first create the window "D".

Initially, the screen consists of a single window with name "@" corresponding to the main edit buffer "@". Windows are managed with the [WINDOW] menu-function:

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

- [C]reate Prompts for the additional information needed to create a new window by splitting the current window.
- [D]elete Prompts for the name of the window to be deleted. Switches to the main "@" edit buffer and window. Only the "@" window cannot be deleted. Exiting or quitting an edit buffer does NOT delete it corresponding window.
- [S]witch Switches to a different edit buffer. (If the selected text register is not already an edit buffer, it is converted into one.) If a window by the same name as the edit buffer exists, this corresponding window will be used; otherwise, the current window will be used by the new edit buffer.
- [Z]oom Zooms the current window to use the full screen. It is usually easier to do extensive editing in an edit buffer if the corresponding window is first zoomed. This zooming remains in effect until you switch to another window (edit buffer).

To create a new window type "C" for "Create" and the status line will prompt:

**[T]op [B]ottom [L]eft [R]ight**

This prompt asks where in the current window you want the new window to appear. For example, for a top-bottom split with the new

window at the top, type "T".

Next, the status line will prompt:

**Enter Window Name:**

Enter the single character name for the window. This will generally be the name of the edit buffer you wish to display in the window.

If you typed T or B for a top-bottom split the status line will prompt:

**Enter # Lines:**

Enter the number of lines you want in the new window. The remaining lines (at least one) will remain in the old window.

If you typed R or L for a right-left split the status line will prompt:

**Enter # Columns:**

Enter the number of columns you want in the new window. The remaining columns (at least 15) will remain in the old window.

To keep the windows on the screen separate, borders are drawn between the windows. Each window has a border line on the top which contains the message "WINDOW *name*", where '*name*' is its single character name. One **WINDOW** will be highlighted to indicate in which window (edit buffer) you are currently editing.

Command Mode Window

When you enter Command Mode (next topic), it will use the current window for its operation - the Visual Mode text will scroll off the window. Alternatively, you can create a window with the special name "\$" which will thereafter be used for the Command Mode. A major advantage of this is that you can observe in the Visual Mode window the effect of the Command Mode commands. The cursor displayed in the Visual Mode window will correspond to the position of the Command Mode "edit pointer".

For example, use the [WINDOW]-Create function to create the window "\$" at the bottom of the screen with five lines. Then enter Command Mode. The screen will then appear similar to:

```

COMMAND                                     FILE: DOCUMENT
=====
WINDOW 0:
=====
=====
=====
=====
=====
WINDOW $:
=====
=====
=====
=====
=====
COMMAND:

```



CREATING WINDOWS

This example gives step by step directions for splitting the screen into two windows, one to the right of the other.



The status line will prompt:

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

2. Type "C" to create a new window.

The status line will prompt:

**[T]op [B]ottom [L]eft [R]ight**

3. Type "R" to create the new window on the right of the screen.

The status line will prompt:

**Enter Window Name:**

4. Type "D" to create a window to be used with edit buffer "D".

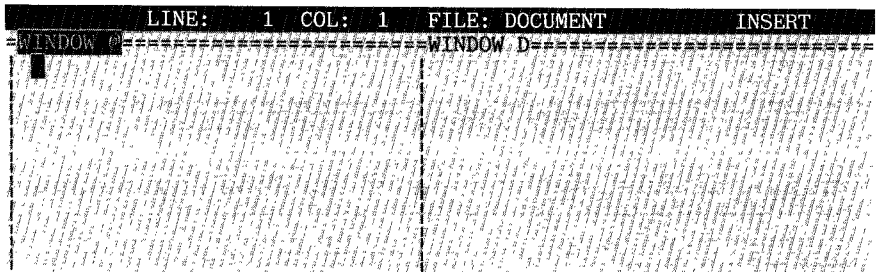
The status line will prompt:

**Enter # Columns:**

5. Enter "40" to split the screen horizontally in half.

Go to Step 2 in the example for "Simultaneously Editing Two Files" to actually use the window.

The screen will appear similar to:



ENTERING COMMAND MODE

Besides the "*Visual Mode*" in which all editing is done on the screen at the cursor position, VEDIT PLUS has a "*Command Mode*", where all editing is done by typing "*command lines*" consisting of single commands or sequences of commands. At the end of each command line you normally press RETURN, just as you do with command lines to the operating system. You do not have to use the Command Mode for simple editing, although its use will simplify many complicated editing tasks.

To enter Command Mode, press [VISUAL EXIT] or [VISUAL ESCAPE] (<ESC> on the IBM PC). The screen will scroll up and the Command Mode prompt "COMMAND:" will appear on the bottom line. The message **COMMAND** will also appear on the status line. The command to go back into Visual Mode is "V" (remember the RETURN). Notice that the cursor is at the same position in the text (but not necessarily on the screen) as it was when you exited visual mode.

SWITCHING FROM VISUAL MODE TO COMMAND MODE

Exit visual mode into command mode. "*Edit Pointer*" takes on last position of cursor. Any currently running macro will resume operation.

|                |
|----------------|
| VISUAL<br>EXIT |
|----------------|

Same as above, but also aborts any command, such as an iteration loop or a command macro.

|                  |
|------------------|
| VISUAL<br>ESCAPE |
|------------------|

SWITCHING FROM COMMAND MODE TO VISUAL MODE

Command to enter visual mode. Cursor takes on last position of command mode "*edit pointer*".

V (Remember the RETURN key)

ON-LINE CALCULATOR

You can use the Command Mode as a simple integer calculator by just typing in the algebraic expression you want calculated. For example type:

$12 + (169 / 13)$  and a RETURN

You may find it useful that the calculator can handle ASCII characters. Just precede the single character with a quote ("). For example, type:

"A and a RETURN

The value of "A is 65 because the letter "A" is the 65th character in the ASCII table. For the value of a control character, the common format of preceding a letter with a caret (^) will display its value. For example, for the value of <CTRL-S> type:

^S and a RETURN

Its value is 19.

You could compute the number of alphabetic letters with the expression:

("Z - "A) + 1

Its value is, of course, 26. This demonstrates that ASCII characters can be combined with numbers in expressions.

All of the different kinds of expressions which VEDIT PLUS can evaluate are covered later in this manual. Expressions can also include the numeric variables and the internal (read-only) values in VEDIT PLUS.

INSERTING A LINE RANGE OF ANOTHER FILE

Purpose: You will often want to insert a portion of another file, such as a paragraph or a subroutine, into the text being edited. The "EG" command is used to extract a specified line range of another file and insert it into your text. It can also insert an entire file. If you don't know the desired line range, you can display a file with line numbers with the "EL" command. This example shows how to insert a portion of "MYFILE.TXT" into the current text.

1. **EL MYFILE.TXT** The file MYFILE.TXT is displayed with line numbers. Press <CTRL-S> to stop/start the screen display. Simply note the beginning and ending line numbers of the text to extract. Let's say the line numbers are 235 and 272 respectively.
2. **EG MYFILE.TXT[235,272]** Lines 235 through 272 (inclusive) of the file MYFILE.TXT are inserted into the text at the edit (cursor) position. The command **EG MYFILE.TXT** would insert the entire file into the text.

NOTES:

1. Filenames may include a drive specifier and MS-DOS pathname.
2. You can use the command form "**EL file[m,n]**" to display a line range of another file. This lets you zero in on the correct lines without displaying the entire file.
3. You may find it easier to simultaneously edit the file from which you want to copy portions of text. This was described under "Copying Text From One File To Another".
4. If you find that the "EL" command types the file too fast, you can slow it down with the "**EP 6**" command parameter.
5. If you get a **\*BREAK\*** message, there was insufficient free memory to insert the entire text and as much as possible was inserted. See the upcoming topic "Freeing More Memory Space".

CONCATENATING TWO FILES

Purpose: You can append one file to the end of another. In this example the text in file '*file2*' is appended to the end of the text in '*file1*' and the combined text is written to the file '*file3*'. The three files can be on different disks.

Note: This assumes that the file '*file3*' will be written to the hard disk or to a floppy disk which will not be taken out in the middle of the operation. The files '*file1*' and '*file2*' can be on separate floppy disks which are swapped at step 5.

1. **VPLUS** Invoke VEDIT PLUS without a filename. Press [VISUAL ESCAPE] to enter Command Mode.
2. **EW *file3*** Set up the output file which will hold the combined text.
3. **ER *file1*** Set up to read the first file. The entire file need not fit into memory.
4. **\_Z** Go to the end of the first file.
5. If needed, you can now take out the floppy disk with '*file1*' and replace it with the disk for '*file2*'.
6. **ER *file2*** Set up to read the second file. The entire file need not fit into memory.
7. **EX** This writes out the complete file '*file3*' and exits VEDIT PLUS.

SPLITTING A FILE INTO TWO OR MORE FILES

Purpose: You can split a large file into several smaller ones. In this example the front, middle and end sections of the large file '*file1*' are split into their own files '*file2*', '*file3*' and '*file4*'. The original file is left unchanged. More flexible splitting can be done by using the text registers.

Note: In this example, "\$" is the <ESC> key, "OW" and "OA" are the digit zero followed by "W" or "A".

1. VPLUS Invoke VEDIT PLUS without a filename. Press [VISUAL ESCAPE] to enter Command Mode.
  
2. ER *file1*\$OA Set up and read the large input file. The entire '*file1*' need not fit into memory.
  
3. EW *file2* Set up the first output file.
  
4. V In Visual Mode, position the cursor on the first character of the second part of the large file. Return to Command Mode.
  
5. OW EF Write the first part of '*file1*' to '*file2*' and close it. OA reads in more of '*file1*'.  
 OA Set up the second output file.  
 EW *file3*
  
6. V See step 4. Not needed if only splitting into two parts.
  
7. (optional) Not needed if splitting into two parts. Write the 2nd part of '*file1*' to '*file3*' and close it. OA reads in more of '*file1*'.  
 OW EF Set up the third output file.  
 OA  
 EW *file4*
  
8. EX Write the rest of '*file1*' to the last output file and exit VEDIT PLUS.

FREEING MORE MEMORY SPACE

Purpose: When using the text registers extensively, you may run out of memory space for performing the desired operations. This is indicated by a \*BREAK\* in Command Mode, or a **FULL** message in Visual Mode. First try and empty any text registers which are no longer needed. If this does not give you enough memory space, you can use the "EN" command to free up more memory in the edit buffer - it will write some of the text out to disk.

Freeing More Memory for the Text Registers:

Use [BLOCK]-Copy to empty any unused text registers.

This was described earlier under "Emptying A Text Register".

-- OR --

Press [VISUAL ESCAPE] to enter Command Mode and empty unused text registers with the command "REr" where r is the text register to empty.

Freeing More Memory for the Edit Buffer:

1.

|                  |
|------------------|
| VISUAL<br>ESCAPE |
|------------------|

Go to Command Mode.

2. RU

If you tried to insert a text register and it didn't fit, use the "RU" command to display the size of the text register. Assume it is 12,000 bytes.

3. 12000EN

This frees up 12,000 bytes for the edit buffer, allowing up to 12,000 bytes from text registers or the "EG" command to be inserted.

RECOVERY FROM FULL DISK ERRORS

Purpose: If you run out of disk space while saving a file you will get a "NO DISK SPACE" error and enter Command Mode. Occasionally you may get a "NO DIR SPACE" error which means the disk has insufficient directory space to hold the rest of the file. You can recover from both of these errors by deleting old files on the disk, or by writing the rest of the file to another disk.

1. EX                                   You attempt to finish the edit session, but you get the disk full error.  
NO DISK SPACE  
\*BREAK\*
  2. ED                                   Look at the disk directory to see what files can be deleted.
  3. EK *oldfile*                       Delete one or more old files on the disk. DON'T DELETE ANY ".\$\$\$" OR ".\$R\$" FILES.
  4. EX                                   Continue saving the edit session.
- OR --
2. EF                                   Save whatever was already written to disk. Lets call this Part 1.
  3. EW *a:part2*                       Set up to write the rest of the text to the file '*part2*' on another disk drive, in this case drive "A:".
  4. EX                                   Write the rest of the text out to '*a:part2*', and exit VEDIT PLUS.

You will now have to concatenate the two parts on the two disks back into one file. See "Concatenating Two Files" discussed previously.

NOTE: See the section "Disk Write Error Recovery" if you do not have enough space on any drive to save all of the file. The procedure becomes more complicated, but you can still save all of your changes and additions to the file.



COMMAND MACROS IN VISUAL MODE

Sequences of Command Mode commands or "*command macros*" can be executed from Visual Mode in two ways. Command macros stored in text registers can be executed with [MACRO]. Alternatively, simple command macros can be built as "*keystroke macros*" which do not require the use of a text register.

To execute a command macro stored in a text register press [MACRO]. The status line will prompt:

**REGISTER 0-9 A-Z**

Type the single character name of the text register holding the desired command macro.

Simple command macros can be built as keystroke macros with the [DEFINE] function. The command macro is then executed by just pressing the function/control key. With the power of command macros, this allows virtually any edit function to be created. These custom edit functions can be "built-in" during Installation.

Visual Mode keystroke macros can be a combination of Visual Mode functions and Command Mode commands. The keystroke macro needs a [VISUAL EXIT] to enter Command Mode. Note: the keystroke doesn't need a "V" command to reenter Visual Mode; this happens automatically. Indeed, having a "V" command causes problems!

Here are a few examples:

Example: Function to "open" a blank line above the current one.  
(Similar to "open-line" in the EMACS editor)

```
[VISUAL EXIT] OL @I/<RETURN>/ -L
```

Example: Function to transpose two letters.

```
[VISUAL EXIT] .p..p+1RCT D -C RGT
```

Example: Function to delete characters from the beginning of the line up to the cursor.

```
[VISUAL EXIT] OK ("Zero" "K")
```

Example: Function to duplicate a line of text.

```
[VISUAL EXIT] OL 1RC9 L RG9 -L
```

ACCESSING COMMAND MACROS WITH [MACRO]

This gives the step by step directions for creating a simple "command macro" in a text register and executing it with the [MACRO] function. In practice, simple command macros should be built as keystroke macros. This macro duplicates the line of text the cursor is on. This saves time when entering lines of text which are very similar to the previous one. The macro will be created in text register/edit buffer "D", which is assumed to be empty.

1. 

|        |
|--------|
| WINDOW |
|--------|

 Type S and then D to switch to edit buffer "D". The text window should be empty.
  
2. Type:  
OL 1RC9 L RG9 -L These are the commands to duplicate the line. Note that "OL" has the digit zero.
  
3. 

|        |
|--------|
| WINDOW |
|--------|

 Type S and then RETURN to switch back to the main edit buffer.
  
4. Position the cursor to a line to be duplicated.
  
5. 

|       |
|-------|
| MACRO |
|-------|

 The status line will prompt:  
**REGISTER 0-9 A-Z**  
Type D to select register "D".  
  
The line will now be duplicated with the cursor positioned at the beginning of the duplicate line.
  
6. You can use the macro as often as you like.

You could create a keystroke macro out of [MACRO] and "D", perhaps assigning them to <ALT-D>. However, for this example, it is easier to build the entire command macro as a keystroke macro. This is described next.

CREATING COMMAND MACROS AS KEYSTROKE MACROS

Simple command macros can be created as keystroke macros. This has the advantage of not requiring a text register. The entire command macro can also be "built-in" during Installation.

This gives the step by step directions for creating the previous line duplicating macro as a keystroke macro.

1. 

|        |
|--------|
| DEFINE |
|--------|

The status line will prompt:  
**Function/Control Key - [DEFINE] to end:**
  
2. Press <ALT-D> and [DEFINE] again.

This assigns the macro to <ALT-D>.

The status line will prompt:  
**Key sequence - [DEFINE] to end:**
  
3. Type:  
[VIS EXIT] OL 1RC9 L RG9 -L [DEFINE]

These are the commands to duplicate the line. Note that "OL" has the digit zero.
  
4. Position the cursor to a line to be duplicated.
  
5. 

|         |
|---------|
| <ALT-D> |
|---------|

The line will now be duplicated with the cursor positioned at the beginning of the duplicate line.

NOTES:

1. Be sure to use [VISUAL EXIT]; [VISUAL ESCAPE] will not work.
2. DO NOT END THE MACRO WITH A "V"; otherwise "NESTING ERROR"s and other problems will occur! The macro will automatically reenter Visual Mode.

EXITING AND QUITTING FROM COMMAND MODE

You can also exit VEDIT PLUS from the Command Mode - in many cases it is even easier. The command "EX" saves the text and exits the current edit buffer; if it is the last edit buffer it returns to the operating system. "EX" is the normal way to exit VEDIT PLUS from Command Mode. As a convenience, the command "EXA" exits all edit buffers, saving all text, and returns to DOS. Similarly, "EQ" abandons the text and quits the edit buffer, and "EQA" quits all edit buffers and returns to DOS.

EXITING / QUITTING FROM COMMAND MODE

|   |          |   |
|---|----------|---|
| SAVE TEXT and Exit<br>Edit Buffer.      | EX       | After exiting from last<br>edit buffer it returns<br>to operating system.   |
|   | -- OR -- |   |
|   | EXA      | Exit ALL edit buffers,<br>save all text and return<br>to the operating system                                       |
| SAVE TEXT and Stay<br>in Edit Buffer    | EY       | Typically followed by an<br>"EB" command to edit a<br>new file.   |
| ABANDON TEXT and<br>Quit Edit Buffer    | EQ       | Requires Confirmation.<br>After quitting last edit<br>buffer it returns to the<br>operating system.                 |
|   | -- OR -- |   |
|   | EQA      | Requires Confirmation.<br>Quits ALL edit buffers,<br>abandoning all text and<br>returns to the operating<br>system. |
| ABANDON TEXT and<br>Stay in Edit Buffer | EZ       | Requires Confirmation.  |