SOFTWARE MANUAL AlphaVUE USER'S MANUAL

DWM-00100-15 REV. BOO



ALPHAVUE USER'S MANUAL

'Alpha Micro', 'AMOS', 'AM-100', 'AlphaBASIC', 'AlphaPASCAL', and 'AlphaLISP'

. ,

are trademarks of

ALPHA MICROSYSTEMS Irvine, CA 92714

©1980 - ALPHA MICROSYSTEMS

ALPHA MICROSYSTEMS 17881 Sky Park North Irvine, CA 92714

Table of Contents

CHAPTER 1

CHAPTER 2

GETTING STARTED

1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 1.10 1.11 1.12 1.13 1.14 1.15	THE KEYBOARD FILES CREATING A NEW FILE THE CURSOR AND DISPLAY WINDOW EDITING TEXT WITHIN A LINE THE MENU INSERTING AND DELETING CHARACTERS DELETING WORDS TABS FORM FEEDS LONG LINES ADDING NEW LINES TO THE FILE PAGE COMMANDS MANIPULATING LINES COMMAND MODE 1.15.1 Getting Help in Command mode 1.15.2 Finishing an Editing Session 1.15.3 Re-editing a File 1.15.4 Quitting an Editing Session 1.15.5 Searching 1.15.6 Replacing Text 1.15.7 Moving Blocks of Text	1-1 1-3 1-4 1-4 1-6 1-6 1-7 1-7 1-9 1-9 1-10 1-11 1-11 1-12 1-13 1-13
2.1	2.1.1 Forward: Control-L (or right-arrow	2-1
	key on some terminals)	2-2
	2.1.2 Backspace: Control-H (or left-arrow	- -
	2.1.3 Delete: RIBOIT	2-2
	2.1.4 Insert: Control-F	2-2
	2.1.5 Scrunch: Control-D	2-2
	2.1.6 Go to Beginning of Line: Control-U	2-2
	2.1.7 Go to End of Line: Control-N	2-3.
	2.1.8 Delete to End of Line: Control-Y	2-3
	2.1.9 Go to Next Word: Control-W	2-3
	2.1.10 Go to Previous Word: Control-A	2-3
	2.1.11 Delete Word: Control-V	2-3
	2.1.12 Erase Line: Control-RUBOUT	2-4
2.2	FILE EDITING COMMANDS	2~4
	2.2.1 Down: Control-J (or Down-Arrow Key	
	or Linefeed Key on some terminals)	2-4
	c.c.c up: control~k (or Up~Arrow Key	- /
	2.2.3 Home: Control=2 (on HOME Kay an accord	2-4
	and Lear Signler Torminale)	.
	and Lear Stegler Terminals)	2-4

CHAPTER 3

٨

 $\left(\begin{array}{c} & & \\ & & \\ & & \end{array} \right)$

 \bigcirc

2.3	2.2.4 Ending Page: Control-E 2-5 2.2.5 Page Down: Control-T 2-5 2.2.6 Page Up: Control-R 2-5 2.2.7 Center Cursor: Control-S 2-5 2.2.8 Insert Line (Split Line): Control-B 2-5 2.2.9 Delete Line: Control-Z 2-5 2.2.10 Concatenate Lines: Control-0 2-6 2.2.11 RETURN 2-6 2.2.12 Next Match: Control-X 2-6 2.2.12 Next Match: Control-X 2-6 2.3.1 ESCAPE 2-6 2.3.2 Break: Control-C 2-6 2.3.3 Toggle Entry Mode: Control-\ 2-7 2.3.4 Toggle Character Insert Mode: 2-7 2.3.5 Setting Block Markers: Control-P 2-7 2.3.6 Absolute Character Insert: Control-G 2-8 SCREEN-EDITING MODES 2-8 2.4.1 Character Insert Mode 2-9 2.4.2 Text Mode 2-9 2.4.4 Fields 2-10 2.4.4.1 Fields 2-10 2.4.4.2 Next
	2.4.4.5 Entry Mode with BASIC files
	2.4.5 Indent Mode
COMM	AND MODE
3.1 3.2 3.3	ENTERING COMMAND MODE 3-1 COMMAND FORMAT 3-1 COMMANDS THAT ACCESS THE DISK 3-2 3.3.1 Finish 3-3 3.3.2 Save 3-3 3.3.3 Go 3-3 3.3.4 Quit 3-4 3.3.5 Yank {n} or Yank {filename} 3-4 3.3.6 Unyank {n} or Unyank {filename} 3-5 3.3.7 Dir 3-6 3.3.8 Erase 3-6
3.4	STRING SEARCH AND STRING REPLACE COMMANDS3-73.4.1Search3-83.4.2Next3-83.4.3Whole3-83.4.4Replace3-93.4.5Global Replace3-103.4.6Wildcard3-10
3.5	BLOCK OPERATIONS 3-11 3.5.1 Copy Block 3-11 3.5.2 Delete Block 3-12 3.5.3 Move Block 3-12 3.5.4 Clear 3-12

3.6	FORMAT	TING COMMANDS	2
	3.6.1	Center	2
	3.6.2	Format 3-1	z
	3.6.3	Width 3-1	ž
3.7	ENVIRO	NMENT COMMANDS 3-1	2
	3.7 1	Puch 2_1	2
	372		4
7 0			4
2.0		SPLIT COMMANDS	4
			4
	3.0.2	Unsplit	4
3.9	EDIT .	3-1	5
3.10	PAGE N	•••••••••••••••••••••••••••••••••••••••	5
3.11	MARGIN	Ν	5
3.12	BLANKS	3-1	6
3.13	PARAME	TER SETTING COMMANDS	6
	3.13.1	Text boolean	6
	3.13.2	Wrap boolean	7
	3.13.3	Searchfold hoolean 3-1	2
	3.13.4	Shik boolean 7-1	י ד
	3 13 5	$\frac{1}{2}$	(
	3 13 6	Help2 boolean	5
	3 13 7	Commont character 7 4	<u>_</u>
	7 17 0	Column a	(
	7 17 0	Column n	8
	3.13.7	Field n 3-18	8
	3-13-10	U Insert boolean	8
	3.13.1	1 Fold boolean	9
	3.13.12	2 Space boolean	9
	3.13.13	3 Delta n	9
	3.13.14	4 Control boolean	ò.
	3.13.15	5 Smart boolean	7 0
	3 13 14	S Abbroviations	1
		3-2(J
EVECI		IE	
LVECC	UTING AC	JE	
41	TNVOKTA		
781	4.1 1	Options 4-1	
	7.1.2	Filonomo	
	7		

CHAPTER 4

INVOKING VUE		4-1
4.1.1 Options		4-1
4.1.2 Filename		4-2
4.1.3 Extension		4-2
ERROR RECOVERY		4-2
INITIALIZATION FILE (INI, VIE)		4-2
4.3.1 Default=extension		
4.3.2 Help=boolean		
4.3.3 Start=Location		4-5
434 Goscommand stripet		4-4
4 3 5 Text=booloop	•••••	4-4
		4-5
4.5.0 wrap=boolean		4-5
4.3.7 Comment=character		4-5
4.3.8 Column=comment column		4-5
4.3.9 Space=boolean		4-6
4.3.10 Insert=boolean		4-6
4.3.11 Fold=boolean	•••••	4-6
4.3.12 Field=character		4-6
4.3.13 Searchfold=boolean		4-7
4.3.14 Indent≃boolean		4-7
4.3.15 Width={number}		4-7
	INVOKING VUE 4.1.1 Options 4.1.2 Filename 4.1.3 Extension ERROR RECOVERY INITIALIZATION FILE (INI.VUE) 4.3.1 Default=extension 4.3.2 Help=boolean 4.3.3 Start=location 4.3.4 Go=command string\$ 4.3.5 Text=boolean 4.3.6 Wrap=boolean 4.3.6 Wrap=boolean 4.3.7 Comment=character 4.3.8 Column=comment column 4.3.9 Space=boolean 4.3.10 Insert=boolean 4.3.11 Fold=boolean 4.3.12 Field=character 4.3.13 Searchfold=boolean 4.3.14 Indent=boolean 4.3.15 Width=fnumber}	<pre>INVOKING VUE 4.1.1 Options 4.1.2 Filename 4.1.3 Extension ERROR RECOVERY INITIALIZATION FILE (INI.VUE) 4.3.1 Default=extension 4.3.2 Help=boolean 4.3.3 Start=location 4.3.4 Go=command string\$ 4.3.5 Text=boolean 4.3.6 Wrap=boolean 4.3.6 Wrap=boolean 4.3.7 Comment=character 4.3.8 Column=comment column 4.3.9 Space=boolean 4.3.10 Insert=boolean 4.3.11 Fold=boolean 4.3.12 Field=character 4.3.13 Searchfold=boolean 4.3.14 Indent=boolean 4.3.15 Width={number}</pre>

(.

	<pre>4.3.16 Wildcard={wildcard characters} 4-7 4.3.17 Entry=boolean 4-8 4.3.18 Sblk=boolean 4-8 4.3.19 Delta={number} 4-8 4.3.20 Control=boolean 4-8 4.3.21 smart=boolean 4-8 4.3.22 Dumb=boolean 4-8 4.3.22 Dumb=boolean 4-8 4.3.22 Dumb=boolean 4-8 4.3.22 Dumb=boolean 4-8 4.4 SAMPLE INI.VUE FILE 4-9</pre>
CHAPTER 5	HARDWARE AND SOFTWARE REQUIREMENTS
	5.1 HARDWARE REQUIREMENTS
APPENDIX A	VUE 2.4 RELEASE NOTES
	A.1NEW SCREEN MODE COMMANDSA-1A.2NEW COMMAND MODE FEATURESA-2A.2.1Disk Access CommandsA-2A.2.2SearchesA-2A.2.3Block CommandsA-3A.2.4Formatting CommandsA-3A.2.5Memory Split CommandsA-3A.2.6New Menu FeaturesA-3A.2.7Parameter CommandsA-4A.3NEW INITIALIZATION FEATURESA-4A.4IMPROVED HARDWARE SUPPORTA-4
APPENDIX B	NEW MENU FORMAT

INDEX

Quick Reference Guide

```
Screen Editing Commands:
 right.....(Control-L)
 left.....(Control-H or backspace)
 up.....(Control-K)
down.....(Control-J or linefeed)
page up.....(Control-R)
page down....(Control-T)
center cursor..(Control-S)
home......(Control-^ - also home key on Lear Siegler and Soroc terminals)
end......(Control-E)
return.....(Control-M or return key)
insert line....(Control-B)
delete line....(Control-Z)
concatenate....(Control-0)
insert char....(Control-F)
scrunch char...(Control-D)
delete to eol..(Control-Y)
delete.....(RUBOUT)
erase line....(Control-RUBOUT)
nextword.....(Control-W)
lastword.....(Control-A)
deleteword.....(Control-V)
beg of line....(Control-U)
end of line....(Control-N)
next match....(Control-X)
escape ..... (escape key)
break.....(Control-C)
toggle entry...(Control-\)
inscharmode....(Control-Q)
set marker....(Control-P)
```

Command Mode:

<pre>Finished Go Quit Yank {n} or {Filespec} Unyank {n} or {Filespec} Dir Erase {Filespec} Search {string} Next {string} Whole {string} Replace {string} Global {string} Wildcard Copy Delete Move Clear Center Format Width {n} Push Pop Split Unsplit Edit Page {n} Margin {n}</pre>	<pre>:finished editing, update files to disk :finish and go :return to EXEC without updating files :read in text :write out text :disk directory :erase a disk file :search for string (starting at cursor) :search for string (including portion of file on disk) :replace string with query :replace string :set wildcard characters :copy a block of text :delete a block of text :clear block markers :center a line of text :format a paragraph of text :set formatting width :save current environment :restore earlier environment :split memory image :inverse of Split :return to Screen Mode :go to new page :set new Left marcine</pre>
Margin {n}	set new left margin
Blanks {n}	·set new left margin
Heln	insert blank lines
leth	:get some assistance

Parameter Setting Commands

Text {boolean} Wrap {boolean} Insert {boolean} Fold {boolean} Space {boolean} Searchfold {boolean} Indent {boolean} Help? {boolean} Comment {character} Column {n} Field {n} Delta {n}	<pre>:enable/disable Text Mode :enable/disable Wrap Mode :enable/disable line insert in Entry Mode :enable/disable folding in Entry Mode :enable/disable generation of space in Entry Mode :enable/disable search case folding :enable/disable search case folding :enable/disable Indent Mode :enable/disable help menu :set comment character :set comment column :set next field character :set auto incomment wolve (() Dietech</pre>
Delta (N)	set auto increment value (for BASIC)

Page viii

CHAPTER 1

GETTING STARTED

This chapter is intended as an introduction for new users. It does not cover every command and option available, but is designed to get you flying quickly with the most commonly used commands. This chapter is designed to be read linearly, while the rest of the manual is intended for reference. Users familiar with computer operation should skip the first two sections. The emphasis of this chapter is on editing text (word processing).

The best way to learn how to use an interactive program like VUE is to sit down and experiment. Try the examples yourself. Don't worry about mistakes. It is impossible to damage the computer from the keyboard.

1.1 THE KEYBOARD

The user communicates with VUE via a keyboard very similar to a typewriter keyboard. However users not familiar with computer terminals will note that there are a few important differences between a typewriter keyboard and a computer terminal.

A normal typewriter keyboard has about 55 keys. However, the English character set is a combination of approximately 90 symbols, including the alphabet (represented in both upper and lower case letters), punctuation marks, and other symbols used very often. In order for the typewriter keyboard to represent the extra characters, most keys have two definitions. The definition of each key is changed by the action of the SHIFT key. The extra characters are produced by hitting a key at the same time as the shift key. The SHIFT key allows 55 keys to produce the English character set.

For computer applications, even the keys for these 90 characters are not enough. Additional keys are needed so that the user may control the computer in addition to entering text. On a typewriter, usually there are only one or two of these "control" keys available: carriage return and backspace. An advanced text editor like VUE, however, may need over 30 of these special keys, bringing the number of characters required to over 120. In order to allow these extra characters to be entered without doubling the size of the keyboard, most keys are given a third definition. The third definition is activated by a key that operates in a manner similar to the SHIFT key, called the CONTROL key. As with the SHIFT key, the CONTROL key is depressed simultaneously with another key to produce a character with a different meaning. For instance, to erase a word in VUE, the CONTROL and V keys are pressed simultaneously. From now on, all references to these "control-characters" will be made in the form "Control-V," where this means "press the CONTROL and V keys simultaneously."

Some control-characters are used so often that terminal manufacturers have added special keys for them so that they can be produced by hitting a single key. Unfortunately, some of the keys have not been standardized, so that a similar key on two different terminals may produce different results. The following control keys are implemented on nearly all terminals:

```
return = Control-M
linefeed = Control-L
backspace = Control-H
tab = Control-I
escape = Control-E or ESC
```

These keys are used often, so it is a good idea to familiarize yourself with their locations on your terminal. Many terminals provide other special control keys. The manual for your terminal should be consulted to find out what control-characters it produces.

Often it is desirable to repeat a given key many times in a row. Many terminals provide a repeat function for this purpose. This function is implemented in two ways. Some terminals, like SOROC IQ12Os, will repeat a key if it is held down for more than a second. Other terminals, like Lear Siegler ADM-3s, provide a separate REPEAT key. This REPEAT key works much like the SHIFT and CONTROL keys, in that it must be held down at the same time as the key you want repeated. As long as both keys are held down, the desired character will be repeated about 10 times a second.

Throughout this manual, we have adopted some graphic conventions to make our examples clearer. For instance, underlined characters indicate those characters that the computer prints on your terminal display. For example, you will often see examples that begin with an underlined dot (_). The underlined dot is the AMOS prompt symbol-- the symbol that the operating system displays when it is ready for you to enter a command. The characters in the examples that you are supposed to type are not underlined.

Another example is this symbol:

RET

This is the symbol that represents the special key usually labeled "RETURN" or "RET". At the end of a line that you are supposed to type in, it means, "press the RETURN key."

1.2 FILES

The floppy diskette or disk pack on which the computer stores information is capable of storing hundreds of thousands of characters. Usually, however, only a small percentage of that space is needed for any particular application. A letter, for example, may only be 20,000 characters long (about 5 pages). The computer, therefore, allows you to break up the disk into small sections, called "files." Each file contains a single set of related information; for instance, a single letter or computer program.

Since a single disk may contain many separate files, both you and the computer need a way to identify them so that they may be used later. In order to do this, each file is given a name. The name is assigned to the file by you. It is generally a good idea to assign a name to a file that relates somewhat to the contents of the file, so that you can remember what is in the file. Under AMOS, the filename can be up to 6 characters long. Some sample filenames are:

> LETTER TAX ORDER PLOTS

In addition to the filename, AMOS also maintains a zero to three character extension for each file. The extension tells the computer what kind of data is in the file. The extension may be specifically mentioned by following the normal filename with a period and the extension. For example:

LETTER_TXT	- text file with a letter in it
LETTER.BAK	 backup of letter
LETTER_LST	- formatted letter ready for printing
TAX.DAT	- tax data file (probably not
	readable directly)
PLOTS_BAS	– a BASIC language plotter program

1.3 CREATING A NEW FILE

We now have enough information to try our hand at some text editing. To do that we will need a file. Let's call our file "LETTER." Before we can create the file, the computer must be turned on, and the operating system must be running. The AMOS operating system signals that it is running by printing a period. When this prompt is issued, we can create our file by typing in "vue letter" followed by RETURN.

.vue letter (RET) AlphaVue Version 2.4 LETTER.TXT does not exist - create it?

After we type RETURN, VUE responds by telling us its name, and then indicates that the file LETTER does not exist yet. We now have a chance to decide if we want to create a new file. Since we do, type "Y" followed by RETURN. After a short delay, a new display appears. The entire screen except for the first line will be filled with asterisks. In the upper left hand corner of the screen, a small white block will appear. This is the CURSOR.

1.4 THE CURSOR AND DISPLAY WINDOW

The cursor is the visual indicator of where we are in our file. Its actual appearance varies with the terminal used. On SOROC IQ120s it appears as a small white block; on Hazeltine terminals it appears as a blinking polygon. The cursor corresponds to the carriage position on a typewriter. The next character entered always appears at the current cursor position, just as a character entered on a typewriter will appear on the paper at the current carriage position.

When editing with VUE, the file can be considered to be a giant roll of paper 510 characters wide and as long as the file. The cursor may be moved back and forth along a line just as the carriage can be moved on a typewriter. Moving up and down in the file is analagous to rolling the paper back and forth in the typewriter. Normally, the file cannot be "rolled" past the end of the last line, just as the typewriter cannot be advanced past the end of a sheet of paper. Whenever rows of asterisks appear, they indicate positions off the end of the file, which cannot be reached. As noted in the last section, when a new file is created, most of the display is asterisks. This is because a new file contains only one blank line. The rest of the screen is "off the end of the roll." Unlike ordinary paper, however, we can easily make the file longer, as if we were gluing an extra piece of paper onto the end.

Although the file may be as long as desired and up to 510 characters wide, only a small portion of the file may be displayed at any one time. Usually, the terminal limits us to 24 lines by 80 characters, but some are smaller and some larger. Therefore, we can only look at a small "window" of the file at a time. When an attempt is made to move the cursor past the edge of this window (for instance, down from the bottom line of the screen), the cursor "drags" the window along with it. Therefore, by moving the cursor around in the file, we can drag the window around in the file and look at any portion of the file.

1.5 EDITING TEXT WITHIN A LINE

Let's go back to our new file LETTER. It presently contains just one blank line, and the cursor is at the very first character of the entire file. Now enter the character A. The A appears on the screen at the cursor position and the cursor moves right one position. As we enter more characters, they appear on the screen and the cursor moves farther right.

Now suppose you make a mistake on the last character typed in. You typed in an F, but you really wanted to type in a G. Simply hit the "rubout" key, usually marked RUB. The cursor backs up one position, and the F disappears. You may now type in the G. Perhaps you decide the entire line is wrong. You could rubout the entire line. However, VUE has a command that will delete the line for you. First type Control-U. The cursor moves to the beginning of the line. Now type Control-Y. The entire line disappears.

Now that we have a blank line, type in "Now is the time to edit with computers". Then type Control-U to get the cursor to the beginning of the line. We can now experiment with moving the cursor around in the line.

First press Control-L. On some terminals, like SOROCs and Lear Siegler ADM-1s and 2s, a separate key is set aside for this function. It is marked with an arrow pointing right. Entering Control-L causes the cursor to move one position to the right. Now enter Control-H. This may also correspond to a key with a left arrow, or a key marked BACK. Entering Control-H causes the cursor to move one character to the left. In other words, the cursor backspaces.

Now continue pressing Control-Ls until you reach the end of the line. Enter Control-L again. Nothing happens. The cursor will not move past the end of the line. This is an important difference between VUE and a typewriter. When using VUE, blanks are not the same as nothingness. To prove this, type a few blanks. The cursor moves forward. The cursor may now be moved back and forth within the blank area as well as within the rest of the line, but it still may not be moved past the last blank.

We now know three methods for moving the cursor around in the file:

- 1. Control-U (cursor to beginning of line)
- 2. Control-L (advance cursor one position)
- 3. Control-H (backspace cursor one position)

There are three more commands for moving the cursor within a line. Control-W moves the cursor forward one whole word, and Control-A moves the cursor back one word. Control-N moves the cursor to the end of the current line. These commands are very useful for moving around quickly in your text.

1.6 THE MENU

We have already learned the functions of five control-characters. When first using VUE, it can be difficult to remember what all the control-characters do. Because of this, a built in "menu" has been provided. To display the menu, simply press the ESC key. The screen is cleared, and a menu of all available commands is displayed. Once you have found the command you want, hit ESC again to return to normal text viewing.

In addition to the control-character menu, additional menus may be available. More information on these additional menus is given in the section on Command mode.

í...

1.7 INSERTING AND DELETING CHARACTERS

Many times you will make a typing error and not notice it for quite some time. Or you may simply wish to change the middle of a line at a later date. With VUE this is no problem. For instance, suppose you typed in the string:

Now is the tme for all go od men

Of course, you really meant "time" and "good." To fix the problem, first move the cursor to the "m" in "tme." Then press Control-F. All the characters to the right of the cursor move right one character, leaving a blank at the cursor position:

Now is the t me for all go od men

You may now type in the "i", correcting the first error.

The second error may now be corrected. Move the cursor to the blank between "go" and "od". Then depress Control+D. The blank will disappear, and the entire line to the right shifts left to fill the space. The line is now correct.

Sometimes it is necessary to insert a lot of characters. It is then inconvenient to type in a lot of Control-Fs. A special mode is provided for this purpose called auto insert mode. When in this mode, every character typed in activates an automatic insert. This mode is toggled on and off by Control-Q.

Suppose in our previous example we had really wanted to say "the right time." We could, of course, just move the cursor to the first character of "time," press Control-F six times, and then type in "right." To use auto insert mode, first move the cursor to the first character of "time." Then press Control-Q. The terminal beeps, indicating that you are now in auto insert mode and you see a Q in the top right-hand corner of your terminal screen. Now type in "right." As you type in the word, the rest of the line moves over to make room. You can type in as many characters as you want, and the rest of the line will move to make room. When you are done, press Control-Q again to return to normal mode.

1.8 DELETING WORDS

After our last example, we decide that we really didn't want the word "right" in the sentence after all. One way of getting rid of it is with six Control-Ds. However, a single command will do the whole job. Simply move the cursor to the first character of the word, and press Control-V. The word and the blank following it disappear, and the rest of the line moves in to fill the space. If the cursor had not been at the beginning of the word, only the end of the word would have been erased. In other words, the word is truncated at the current cursor position.

The Control-Y function introduced earlier also truncates. We used Control-Y at the beginning of the line, and the entire line was erased. If Control-Y is used somewhere besides the beginning of the line, only the line to the right of the cursor is erased.

1.9 TABS

Tabs are used to columnize data. A tab is entered either with the TAB key or with Control-I. The tab stops are set at every eight characters.

When first using VUE, it may take a while to get used to the behavior of tabs. Although they appear as several spaces on the screen, they are stored internally as one character. Therefore, these spaces on the screen do not really exist. You may recall that the cursor cannot be moved to a position where where there is no character, such as beyond the end of a line. The cursor also may not move into the middle of a tab field. It may however, move around a tab field. As the cursor moves along a line with tab character in it, it will appear to jump across the tab areas. Therefore, a character may not be added in the middle of a tab field, only at either end. It's not really as confusing as it sounds. A few minutes of practice and you'll be an expert.

1.10 FORM FEEDS

When VUE encounters a form feed (a Control-L) in your text file, it replaces the form feed with a line of ^ symbols. This makes the page breaks easy to find. When you leave VUE by using the F (Finish) command, VUE converts the line of ^ symbols back into a single form feed. You can safely use the ^ symbol in your file; VUE only converts a line of ^ symbols into a form feed if the line is as wide as the screen.

1.11 LONG LINES

It was mentioned earlier that lines in the file could be up to 510 characters long. Unfortunately, very few terminals are this wide, so we can only look at a small portion of such a long line at one time. If we fill a line on the screen with characters and then keep typing, we find the characters we have already typed in moving to the left. The characters at the left end of the screen disappear. The column at the extreme left of the screen is no longer column zero. We can keep typing until the 510th character has been entered. At that point, the cursor no longer advances. By using the six cursor positioning commands we can move the cursor back and forth through our line. The "window" displayed will move around as necessary, so we can view the entire line.

1.12 ADDING NEW LINES TO THE FILE

Up to now, all our operations have taken place on a single line. Real files, however, are nearly always longer than one line! To add a new line to the file, press RETURN. The cursor moves down to the beginning of the next line, and a row of asterisks disappears. The file is now two lines long. We can make the file as long as we want by simply continuing to press RETURN.

Now that we have our new lines, how can we move around in them? Since the cursor is at the end of the file, we probably want to go up. To do so, press Control-K. (On some terminals the up-arrow key corresponds to this control character.) The cursor moves up one position. Each time Control-K is pressed, the cursor moves up one line, until the beginning of the file is reached.

Now to get back down. We have two choices. The first is Control-J. (On some terminals this may be the down arrow key or the FEED key.) This moves the cursor down one line. If possible, the cursor will stay in the same column. The second choice is the RETURN key. This also moves the cursor down one line, but the cursor always goes to column zero. Note that the RETURN key does not make the file longer unless you are already at the last line in the file.

When the cursor moves from one line to another, an important consideration is the fact that the cursor cannot wind up in a position where there is no character. If after changing lines, there is no character in the column the cursor was in before, the cursor moves until it is pointing to a character. For example suppose the file contains the text:

> This is an example of text editing | | | cursor before Control-J cursor after Control-J

Before the Control-J, the cursor pointed just past the end of the first line. Now we tell the cursor to move down a line. It does so, but notices that it is now beyond the end of its new line. Therefore, it moves backward until it reaches a character (the carriage return at the end of the line). If we now tell it to move up a line (Control-K), the cursor simply moves up, since there is a valid character in this position. (In this case, a blank.) These rules also apply to tab fields. If the cursor finds itself in the middle of a tab field after moving from one line to another, it moves left until it encounters a character.

If you have been trying the examples above on the computer, try some experimenting now. Get the feel of the commands already introduced. While you are at it, type in a couple of pages of text. You will need a couple of pages of text to try out the commands in the next section.

1.13 PAGE COMMANDS

We have already learned how to move the cursor up and down in the file one line at a time. Often, however, we want to move much faster than this. Four commands are available to move an entire page at one time. The first is Control-^{*}. (The HOME key produces this character on SOROC and Lear Siegler terminals.) This command moves the cursor to the beginning of the entire file. The Control-T command moves the cursor down a page (one screenful, usually 24 lines) at a time. Control-R moves the cursor up one page. The final page command is Control-E, which moves the cursor to the last line in the file. When you use a Control-E, the last 1/4 of the screen is filled with asterisks that signify the end of the file.

1.14 MANIPULATING LINES

We already know how to add lines to the end of a file, but suppose we need to add more lines in the middle of a file? VUE provides for this need. First, move the cursor to the beginning of the line. Then press Control-B. The cursor does not move, but the line the cursor was on and all the lines below it move down one, leaving a blank line behind. If more blank lines are needed, press Control-B again.

Of course, sometimes what you really want to do is delete lines. This function is also provided. Simply move the cursor to anywhere within the line to be deleted. Then press Control-Z. The entire line disappears, and all the lines below it move up to fill in the space previously used by the deleted line.

In addition to inserting or deleting lines, it is also possible to split a line in two or to concatenate (i.e. join) two lines. To split a line, move the cursor to the point in the line where the line is to be split. Then press Control-B. The line splits, with the right half of the line moving down to the next line. Note that Control-B is also the command that inserts a blank line. In fact, inserting a blank line is really splitting a line at its beginning.

To concatenate two lines, move the cursor to anywhere in the upper line to be concatenated. Then press Control-O. The line below the cursor moves up and joins the end of the line that the cursor is on, creating a new line consisting of the two old lines. This command is ignored if the new line would be longer than the maximum of 510 characters.

1.15 COMMAND MODE

Up to now, we have been using VUE in SCREEN MODE. In Screen mode, the text you are editing is always visible. Some commands, however, are available only in another mode, called COMMAND MODE. In this mode, the text you are editing is not visible. This mode is used for non text related commands, like reading from and writing to the disk. To get into Command mode when you are in Screen mode, simply type ESC. The screen is cleared; then VUE displays the name of the file it is editing and the amount of free space left in memory. (NOTE: If the amount of free space gets under approximately 500 bytes, some of your file should be written to the disk. See the Unyank command, Section 3.3.5, "Unyank".) VUE then prints the menu. Finally, a > will appear on the fifth line of the screen. The > is the VUE prompt symbol, which means that VUE is in Command mode and is waiting for your command.

If, while in Command mode, you decide to go back to Screen mode, simply press the ESC key. You are immediately returned to Screen mode.

The rest of this chapter discusses a few of the commands you can use while in Command mode. NOTE: These commands can be altered on your system within a certain range to help meet specific needs. For the most part, in this manual we will assume that you are using VUE with all parameters set as they came on your System Disk.

1.15.1 Getting Help in Command mode

Normally, when Command mode comes up, it prints a menu of screen control commands. After the first command (or just a RETURN) is entered, however, a new menu is displayed. Usually, this menu describes the VUE Command mode commands. However, additional menus may be available. To find out what menus are available, simply type HELP, followed by a RETURN:

AlphaVue 2.4 Status: space fold srchfold text

Editing INTRO.TXT 2630 bytes free margin 0 Page 20

Help is available for: SCREEN COMMAND TXTFMT BASIC

VUE responds with a list of the available menus. To get more information on one of the subjects listed, type HELP followed by the subject name:

AlphaVue 2.4 Status: space fold srchfold text

Editing INTRO_TXT 2630 bytes free margin 0 Page 20

>HELP TXTFMT (RET)

TXTFMT Text Processor Commands:

F)ormat - format mode U)nformat - unformat mode Number - set page number P)age - eject page

•••••••••••• continued list of TXTFMT commands

1.15.2 Finishing an Editing Session

Eventually, you will finish editing your new file. Up to now, your file has existed in your computer's internal memory. The file is in internal memory rather than on disk because the internal memory may be changed quickly, making fast screen editing possible. However, when the computer is turned off, all the contents of internal memory are "forgotten." In order to save the result of your work, the contents of internal memory must be saved on disk. To do this, enter Command mode and type in "F" or "FINISH" followed by a RETURN. VUE will save your text on the disk and then return you to the operating system. For example:

AlphaVue 2.4 Status: space fold srchfold text

Editing LETTER.TXT 24856 bytes free margin 0 Page 12

>FINISH...

-

1.15.3 Re-editing a File

After you have finished editing your new file and have returned to the operating system, you may need to re-edit the file again to correct errors or to expand or revise it. To do so, simply type VUE followed by the name of the file you want to edit. This is the same procedure as when you created a new file. However, since the file already exists, VUE does not ask if you want to create it. For example, to re-edit your letter, type:

_vue letter (RET) AlphaVue Version 2.4 Loading LETTER.TXT

VUE tells us it has found the file and is loading it into internal memory. As the file is read into internal memory, VUE prints a period as every 64th line is read in. This is simply to let you know that everything is proceeding normally.

When the entire file has been read in, the screen switches to a display of the first page of your file. You are now in Screen mode and may begin re-editing your file.

1.15.4 Quitting an Editing Session

Since VUE provides so many powerful functions for editing your file, it is possible to edit your file "to death." For instance, you may accidentally erase 50 lines of text you really wanted to keep. If this happens, the QUIT command can be used to restore those 50 lines. The QUIT command returns you to the operating system without saving internal memory. As a result, the disk file does not contain any of the changes you made the last time you edited. Note that the QUIT command is not selective. ALL changes you made

(...)

are deleted, even if you wanted some of them. For instance, if you typed in 50 lines of text before deleting the other 50 lines, you can not recover both the 50 deleted lines and the 50 new lines; you have to make a choice between them.

To invoke the quit command, simply type in Q or QUIT followed by a RETURN. For example:

AlphaVue 2.4 Status: space fold srchfold text

Editing LETTER.TXT 23458 bytes free margin 0 Page 9

>Q RET

1.15.5 Searching

A common need when editing is to find a particular word or string of characters in a file. For instance, you might want to find where you had used a person's name in a file, or where a particular section of a document is located. VUE does this searching for you automatically, using the SEARCH command. The SEARCH command consists of S or SEARCH followed by a blank followed by the string you are searching for. For example, suppose you were editing this manual, and wanted to find every reference to the cursor. You would type:

>S CURSOR BET

(At this point, it should be noted that the search command considers upper and lower case letters to be the same. Therfore, CURSOR is the same as cursor and Cursor.)

If VUE finds the word CURSOR anywhere in the file, it displays the page on which the word appears, with the cursor on the word itself. You are now in Screen mode, and can edit with the normal screen editing commands.

After looking at and possibly editing the file around the first occurrence of the word CURSOR, you may need to look at the next place in the file where the word CURSOR exists. To do so, press Control-X. If VUE finds another occurrence of the word CURSOR, it displays the new location of the word, and again allows you to look at and edit the area. You may repeat this sequence until you have located every occurrence of the word CURSOR in the entire file.

When VUE cannot find the word you are searching for, it returns you to Command mode so that another SEARCH command may be executed.

1.15.6 Replacing Text

Many times it is necessary to replace every occurrence of a word in a file with another word. For instance, you might want to change the word CRT to Terminal everywhere in the file. VUE provides a command which does most of the work for you; the REPLACE command.

The format of the REPLACE command is very similar to the SEARCH command—an R or REPLACE followed by a blank followed by the word you want to replace. VUE then displays a question mark to ask you what you want the word replaced with. Let's try our example of replacing CRT with Terminal.

AlphaVue 2.4 Status: space fold srchfold text

Editing LETTER.TXT 3064 bytes free margin 0 Page 2

>REPLACE CRT (RET) ?Terminal (RET)

VUE will then begin searching for the word CRT, and if it finds it, it displays the word just as the search command does. However, you will not be in Screen mode. VUE will only accept one of five characters here as valid input -- a Y, Q,N, Control-C, or ESC. If you hit Y, VUE replaces the string. In this case, the word CRT would be replaced with Terminal. If you hit N, VUE does not replace the string, but leaves it alone. In either case, VUE then searches for the string CRT again, and the whole process is repeated.

If you type in Q, ESC or Control-C, VUE will terminate the REPLACE command, and you are returned to Command mode. You are also returned to Command mode if VUE cannot find any more occurrences of the word in the file.

1.15.7 Moving Blocks of Text

Often, when editing a document or piece of text, it is necessary to move a large block of text from one location to another. To do this, first move the cursor (while in Screen mode) to the beginning of the block to be moved, and press Control-P. The line the cursor is on will now appear in reduced intensity on the screen. Now move the cursor to the end of the block to be moved, and press Control-P again. All the text in the block will now appear in reduced in reduced intensity.

The cursor should now be moved to the desired location of the block of text. The cursor cannot be inside the marked block during this operation. In otherwords, you cannot move a block to be inside of itself. Press ESCAPE to get into Command mode, then enter the command MOVE followed by a carriage return. VUE will move the block of text to the new location and return you to Screen mode. The block is removed from its old position.

If desired, you can copy the block instead of moving it. In this case, the block is copied to the new location, but the old copy is not removed. To do this, mark the block as for a move, but enter the command COPY instead of MOVE. Also, the marked block may simply be removed from the text with the command DELETE. The text found after the deleted block is brought together with any text that preceded the deleted block.

When you are through with the marked block, it may be "unmarked" with the CLEAR command. This restores all the text to full intensity.

Some terminals, like ADM3As, do not have the reduced intensity feature. See Chapter 3, "Command Mode," for methods for moving blocks of text with these terminals.

. .

CHAPTER 2

SCREEN MODE

VUE has 29 commands that you can use while in Screen mode; you'll use these commands to rapidly position the cursor to any point in your text, and to add and delete characters and words.

If any command is unclear, experiment. The result of any screen command is immediately visible, so there is never any question as to what is happening to your text. What you see is what you get!

(NOTE: Some terminal drivers may redefine the control characters used to activate VUE commands. In particular, some Hazeltine terminals use different control characters than the ones described in this manual. Consult the sources of the terminal driver you are using if there is any question.)

2.1 LINE EDITING COMMANDS

These comands are used to edit within a line of text. A line of text is all of the characters between two carriage return symbols. VUE can accept a line of up to 510 characters.

When you add or erase characters, this action takes place at the current cursor position on the text line. Move the cursor, using the commands described in this and the next sections, until it is at the point in the text that you wish to change. At that point you can change the characters there by simply typing your new text; the new characters will overwrite the old characters on the line. To insert characters into a text line, use the Insert command or enter character insert mode (see Section 2.5.1, "Character Insert Mode"). To delete characters, type a RUBOUT, use the SCRUNCH command, or use the various word, line, and block-of-text delete commands described later in the manual.

2.1.1 Forward: Control-L

To move the cursor from left to right along a line of text, type a Control-L. Each Control-L moves the cursor one character position to the right. When you reach the end of the text line, the cursor stops. If the new position of the cursor would be off the screen (because the text line is wider than the screen line), VUE shifts the line left so that the cursor remains on the screen.

2.1.2 Backspace: Control-H

To move the cursor from right to left along a line of text, type a Control-H. Each Control-H moves the cursor left one character position. When the cursor reaches the beginning of the line, the cursor stops. If the position of the cursor would be off the screen, VUE shifts the line right so that the cursor remains on the screen.

2.1.3 Delete: RUBOUT

When you hit the RUBOUT key, the cursor backspaces from right to left, replacing characters with blanks. If you are in character insert mode, however, a RUBOUT acts as a BACKSPACE command (see above) and a SCRUNCH command (see Section 2.2.5, "Scrunch: Control-D").

2.1.4 Insert: Control-F

To insert a blank into a text line, type a Control-F. Unless you are in character insert mode, you will want to type a Control-F for every new character that you insert in a line; otherwise VUE will overwrite the old characters with the new ones.

2.1.5 Scrunch: Control-D

To delete the character to the right of the cursor, type a Control-D; this removes the character, and shifts all of the characters from the right of the cursor to the end of the line left one character-position.

2.1.6 Go to Beginning of Line: Control-U

To move the cursor to the beginning of the current line, type a Control-U. (Side effect: if the left margin was not zero, it is set to zero until you leave the line.)

SCREEN MODE

2.1.7 Go to End of Line: Control-N

To move the cursor to the end of the current line, type a Control-N. If the line is longer than the screen width the left margin is shifted so that the end of the line is on the screen.

2.1.8 Delete to End of Line: Control-Y

To delete all of the characters at and to the right of the cursor until the end of the line, type a Control-Y.

2.1.9 Go to Next Word: Control-W

To move the cursor to the beginning of the next word, type a Control-W. As you type Control-Ws, the cursor continues to move to the right until it reaches the end of the line. The cursor stops at the end of the line unless you are editing a text file (i.e. Text Mode is on), in which case the cursor moves to the first word at the beginning of the next line.

2.1.10 Go to Previous Word: Control-A

To move the cursor to the beginning of the previous word on the line, type a Control-A. As you type Control-As, the cursor stops at the beginning of the line unless you are editing a text file (i.e. Text Mode is on), in which case the cursor moves to the end of the previous line.

2.1.11 Delete Word: Control-V

To delete all characters from the right of the cursor up to the beginning of the next word, type a Control-V. If the cursor is at the beginning of a word, VUE deletes that word and any blanks trailing that word. If you are in a field of blanks, VUE erases all of the blanks in the field to the right of the cursor. If you are in the middle of a word, VUE deletes the rest of the word and any trailing blanks.

A Control-V does not delete tabs. If you delete a word before a tab, VUE replaces the word with tabs so that any characters following the trailing tab do not move. VUE works this way so that you can change items in one column of text without disturbing the placement of other columns. For example, if you delete an operand in an assembly language program, the comment field will not move. To delete a tab, use the RUBOUT key or the Scrunch command.

(

2.1.12 Erase Line: Control-RUBOUT

To delete all the characters in the current line, type a Control-RUBOUT. This command is equivalent to a Control-U followed by a Control-Y. The line itself is not deleted as it is with the Control-Z command.

2.2 FILE EDITING COMMANDS

Use the following commands to move the cursor up and down through the file and to manipulate entire lines of text.

2.2.1 Down: Control-J (or Down-Arrow Key or Linefeed Key on some terminals)

To move the cursor down from one line to the next, type a Control-J. When you reach the end of the text, the cursor stops. As it moves down the screen, the cursor stays in the same column position on the screen unless it encounters a tab or a blank line (see Section 1.9, "Tabs"). If you try to move the cursor off the bottom of the screen, VUE moves the screen display (the "window") up a line so that the cursor stays on the screen.

2.2.2 Up: Control-K (or Up-Arrow Key on SOROC Terminals)

To move the cursor up to the previous line, type a Control-K. When you reach the beginning of the text, the cursor stops. As you type Control-Ks, if you try to move the cursor past the top of the screen, VUE moves the screen display down 12 lines, and the cursor is in the middle of the screen at the line that was previously off of the screen. As the cursor moves up the screen, it stays in the same column position unless it encounters a tab or a blank line (see Section 1.9, "Tabs").

Some terminals, like ADM2s, support reverse scrolling. On these terminals, VUE will scroll the display down one line when it reaches the top of the screen instead of 12 lines as described above.

2.2.3 Home: Control-^ (or HOME Key on SOROC and Lear Siegler Terminals)

To move the cursor to the beginning of the first line of text, type a Control—^ (or hit the HOME key). The first page of text is displayed, with the cursor in the upper left hand corner of the display.

SCREEN MODE

 $i = \frac{1}{2}$

2.2.4 Ending Page: Control-E

To move the cursor to the end of the last page of your text, type a Control-E. The cursor appears at the beginning of the last line of text. The lower 1/4 of the screen display consists of lines of asterisks, indicating that you are seeing the end of the text file.

2.2.5 Page Down: Control-T

To move the cursor one page down in your text, type a Control-T. The next page of text is displayed. If the cursor position before the Page Down command was less than one page from the end of the file, the screen will look the same as if you had used the Ending Page command.

2.2.6 Page Up: Control-R

To move the cursor one page back in your text, type a Control-R. The previous page of text is displayed. If you were within one page of the beginning of your text, the display looks the same as if you had entered a Home command.

2.2.7 Center Cursor: Control-S

The Center Cursor command tells VUE to redisplay the screen so that the line of text containing the current cursor position is in the middle of the screen.

2.2.8 Insert Line (Split Line): Control-B

To insert a blank line into your text file (or to split an existing line), type a Control-B. If the cursor is at the front of a line, a Control-B inserts a blank line (that is, a carriage return symbol alone on a line). If the cursor has text to the left of it on the line, a Control-B splits the line into two by inserting a carriage return at the cursor position.

2.2.9 Delete Line: Control-Z

To delete a line of text (or a blank line), type a Control-Z. VUE erases the line and moves the text below the cursor up one line on the screen.

2.2.10 Concatenate Lines: Control-0

To combine two lines of text, type a Control-O. VUE adds the line of text below the cursor onto the end of the line of text that contains the cursor. All lines below the cursor move up one line on the screen. If the merged line would be longer than 510 characters, VUE does not execute the Concatenate Line command; instead it rings the terminal bell.

2.2.11 RETURN

To move the cursor to the beginning of the next line, type a RETURN (sometimes labeled RET on your keyboard). If you are already at the end of your text, a RETURN creates a new line by entering a carriage return symbol. (If you are in entry mode you can also cause RETURN to begin a new line of text even if you are not at the end of text (see Section 2.5.4, "Entry Mode"). This is very useful when you are entering new text into the middle of a file.)

2.2.12 Next Match: Control-X

To find the next occurrence of a string you entered to the Search command (see Section 3.4.1, "Search"), type a Control-X. The display on the screen is of the page in the text file in which the string occurs. The cursor is at the beginning of the found string. To look for the next occurrence of the string in the file, type another Control-X. If VUE cannot find another occurrence of the string, it returns to Command mode.

2.3 CONTROL COMMANDS

Control commands do not change the text or move the cursor, but they do select various editing modes and perform special control functions.

2.3.1 ESCAPE

The ESCAPE key (sometimes labeled ESC or ALT MODE on your keyboard), enables you to switch between Screen and Command modes. You may enter Command mode at any time while editing by typing an ESCAPE. You may return to Screen mode by typing an ESCAPE after the Command mode prompt.

2.3.2 Break: Control-C

It is possible to enter many more commands to VUE than it can process immediately. For example, you might hold down the CONTROL and the Z keys, and so enter 10 or 20 Delete Line commands. VUE saves these commands that you enter until it can process them. To clear the VUE storage area of unprocessed commands, type a Control-C. This is a useful thing to be able to do when you've typed ahead a great many commands, and you want to interrupt VUE from processing them all.

2.3.3 Toggle Entry Mode: Control-\

Type a Control-\ (Control-Backslash) to leave or enter entry mode (see Section 2.5.4, "Entry Mode"). When you are in entry mode, you see an uppercase I in the upper right hand corner of the screen display. VUE reminds you that you have entered or left entry mode by sounding the terminal bell when you type a Control-\.

2.3.4 Toggle Character Insert Mode: Control-Q

To leave or enter character insert mode, type a Control-Q (see Section 2.5.1, "Character Insert Mode"). To remind you that you are in Character Insert Mode VUE displays an uppercase Q in the upper right hand corner of the screen.

2.3.5 Setting Block Markers: Control-P

AlphaVue supports a variety of operations on blocks of text. Among other operations, you can move copy and delete a block of text. In order to operate on a block of text, the block must be defined in some manner. To define a block, the command Control-P is used.

To define a block of text, move the cursor to one end of the desired block. Press Control-P. The line the cursor is on is redisplayed in reduced intensity. Now move the cursor to the other end of the desired block and press Control-P again. The entire block of text is now redisplayed in reduced intensity.

Until the markers are cleared in Command mode (See Section 3.5.4, "Clear"), all the text in the marked block will appear in lower intensity. You may edit text in the block, and any new text inserted into the block also appears in low intensity.

The size of the block may be varied by moving the cursor and pressing Control-P again. If the cursor is above the top of the current block, the block is extended upwards. If the cursor is below the top of the current block, the cursor position replaces the old bottom of the block.

Some terminals do not support reduced intensity display. The Control-P command will work on these terminals; however, it is not very convenient because the block is not visible. For such terminals, an alternate method of defining blocks is available, using delimiting symbols in the text. The delimiters are:

[* - start block of text

*] - end block of text

Example:

```
text in a file
[*this is some text
*]
hi there
```

The example delimits the string "this is some text."

For more information on blocks, see Section 3.5, "BLOCK OPERATIONS."

2.3.6 Absolute Character Insert: Control-G

Normally, VUE uses control characters to control its operation in Screen mode, and control characters are not allowed in the file. However, some applications may require control characters in a file. If you wish to edit a file with control characters in it, your initialization file should contain the statement CONTROL=TRUE. (For further information, see Section 4.4, "INITIALIZATION FILE.") Otherwise, VUE removes all control characters as it reads the file into memory, and the Control-G command is disabled. (NOTE: VUE automatically assumes that CONTROL=TRUE for files with a .LST extension)

If CONTROL=TRUE, the Control-G command allows you to enter control characters in the file. The key typed in after the Control-G is inserted into the file at the current cursor position, even if it is a control character. (However, carriage returns, linefeeds, and RUBOUTS may not be inserted into the file.)

Once a control character has been inserted into the file, it is displayed as an up arrow (^) followed by a letter representing the control character. For instance, if the character O4 hex appears in the file, it is displayed as "^D_" As the cursor moves along a line with control characters in it, it appears to jump across the control characters, just as it jumps across tab fields.

2.4 SCREEN-EDITING MODES

Screen mode encompasses several character-entry modes that affect the way VUE handles your text. These modes are character insert mode, entry mode, text mode, wrap mode, and indent mode: Character insert mode allows you to easily insert text into the middle of a line; text mode is useful when editing text files; wrap mode automatically adds carriage returns at the end of the line; entry mode is designed to make easier the task of entering formatted text (e.g., assembly language programs); and indent mode is useful when entering structured-language programs, such as Pascal or C programs. Any combination of modes may be used at one time.

2.4.1 Character Insert Mode

VUE usually overwrites existing text with new characters; that is, VUE normally replaces the old characters at your current cursor position with the new characters that you are entering. To insert characters you use the Insert command (Control-F) to enter blank spaces which are in turn replaced by your new characters; or use the Insert line command (Control-B) to create a blank line on which to place your new text. This is somewhat inconvenient when you want to add a few words in the middle of a text line.

Character insert mode allows you to enter characters WITHOUT overwriting existing characters. That means that as you enter characters, the existing characters on the line are shifted over to the right to make room for the new characters. VUE usually responds to the RUBOUT key by backspacing and writing a blank over the character you want to delete; in character insert mode, a RUBOUT performs the same function as a BACKSPACE and a SCRUNCH command. As you delete characters, the line of text shifts to the left to close up the gap.

Remember that Control-Q toggles the Character-Insert mode

2.4.2 Text Mode

Text Mode changes the definitions of the Go to Next Word (Control-W) and Go to Previous Word (Control-A) commands. Normally when using these commands, the cursor stops when it reaches either end of a line. If Text Mode is set, the cursor will continue to the next or last line of text. Text Mode also disables all the features of Entry Mode except for the auto line insert after carriage return. Text Mode is turned on and off in either Command mode or the initialization file. (NOTE: VUE automatically turns on Text Mode when you are editing files with a .TXT extension.)

2.4.3 Wrap Mode

Wrap Mode allows you to enter text without having to worry about entering carriage returns. When the cursor reaches the end of the screen, VUE automatically creates a new line, and moves the last word typed into the new line. You may enter text continually without worrying about entering carriage returns.

Note that Wrap Mode only works when you first enter text. In other words, inserting a character in the middle of a line will not cause the last word on that line to wrap around. Wrap Mode is turned on and off either in Command mode or the initialization file.

2.4.4 Entry Mode

When editing text files (with Text Mode on), you will find entry mode useful for inserting lines of text into the middle of your file. Usually (if you are not at the end of your text) a RETURN tells VUE to simply move the cursor to the beginning of the next line. When you are in Entry Mode, a RETURN tells VUE to create a new line below the current line of text. This allows you to enter text without using the Insert Line command (Control-B).

When editing files with Text Mode off, Entry Mode enables a variety of functions. Some of these functions are set up by the VUE initialization file, INI.VUE (see Section 4.4, "INITIALIZATION FILE"). When you are in Entry Mode, you see an uppercase I at the right hand corner of your screen display. To enter or leave entry mode, type a Control-\ (a Control-backslash).

Below we discuss the features of Entry Mode that apply to non-text files.

2.4.4.1 Fields - An important concept in entry mode is the field. VUE recognizes four fields: the label field, the opcode field, the operand field, and the comment field. These correspond to the four fields used by the assembler. Normally, these fields are set up as follows:

0 8 16 40 ; comment field columns 40-510 operand field columns 16-39

opcode field columns 8-15

label field columns 0-7

You may change the location of the comment column (normally 40) by changing the VUE initialization file. You can also change the comment character (normally a semicolon) to any character by changing the INI.VUE file.

Assuming the comment column is 40, the actual comment text may begin at either column 41 or 42. Some programmers like to have a space between the

comment character (semicolon) and the actual comment. In this case, the SPACE flag should be turned on, causing the actual comment text to begin at column 42.

If a line begins with the comment character, VUE considers the entire line to be the comment field.

2.4.4.2 Next Field - The following discussion assumes that you are in Entry Mode and not in Text Mode.

You may define a next-field character in your INI.VUE file; the default next-field character is a space. When you type the next-field character, VUE moves the cursor to the next field (see above) to the right of the cursor.

If the cursor is in the middle of a line, and you type the next-field character, the cursor will advance to the beginning of the next field. For instance, if the cursor is at column 9 (in the opcode field) it moves to column 16, the beginning of the next field. The next time you type the next-field character, the cursor advances to column 41 (the position after the start of the comment field). VUE assumes that you do not want to edit the comment character; if you do, just backspace and do so. If the SPACE flag is on, VUE moves the cursor to column 42, putting a space after the comment character.

If you try to move the cursor to an empty field, VUE creates the field for you. In the case of our last example (with the cursor at column 9) VUE adds a tab character to the line; this moves us up to column 16, and adds the operand field. If we hit the next-field key again, VUE adds enough tabs to the line to get us to the comment column (in this case 3 tabs, to get us to column 40). VUE then automatically adds the comment character to the line and moves the cursor past it. If the SPACE feature is enabled, VUE also automatically adds a space after the comment character.

Once the cursor is in the comment field, entering another next-field character has no effect on the text file except for inserting that character into the file if it is a printable character.

NOTE: When editing a text file (extension .TXT) VUE sees the entire file as a comment field; the next-field character has no effect.

2.4.4.3 Folding - Another feature active in Entry Mode is character folding. This means that VUE changes all characters you enter from lower to upper case unless they are in a comment field. This makes it easy to have upper case program statements and lower case comments. Simply leave the keyboard in lower case, and VUE will do the rest. This feature is disabled when editing text files (extension .TXT); you may also disable it in the INI.VUE file.

SCREEN MODE

(

2.4.4.4 Line Insert — The final feature available in Entry Mode is line insert after carriage return. When in Entry Mode, a line is inserted after every carriage return entered. This is handy when entering blocks of text in the middle of the file. You may disable this feature in the INI.VUE file.

2.4.4.5 Entry Mode with BASIC files (Auto Line Numbering) - When editing BASIC files (extension of .BAS), the operation of Entry Mode is slightly modified. The second operand field disappears, the comment character is set to the exclamation mark (!), the next field character becomes a tab, and the comment field normally begins at column 56. The most important feature, however, is auto line numbering.

When Entry Mode is on and a new line is started, entering a TAB produces a line number followed by a tab. The text of the line may then be typed in normally. If desired, pressing TAB again moves the cursor out to the comment field and adds the comment character, just as with assembly language programs. As with assembly language programs, the comment field is fully configurable.

To calculate the line number, VUE looks up the line number on the preceding line. It then adds the current auto-increment value (normally 5) and enters the number on the new line.

2.4.5 Indent Mode

Indent Mode is designed for use with structured languages like Pascal. It is enabled by the Indent command in the initialization file and Command mode. Its purpose is to make easier the task of editing programs with indented blocks of text.

When Indent Mode is on, the action of RETURN and the Beginning of Line (Control-U) command is modified. Both commands now move the cursor to the first text character instead of column one. In other words, any leading blanks or tabs are ignored and the cursor moves past them.

If a RETURN results in the creation of a new line (i.e., you are already at the end of file or in Entry Mode with the insert option on), then the new line is automatically indented with the optimum number of tabs and blanks so that the indentation of the new line matches the line above it. If this is not the amount of indentation required, you may then modify the indentation of the line with the space, RUBOUT, and/or TAB keys.

CHAPTER 3

COMMAND MODE

Forty-five commands are available in Command mode. These perform disk I/O, string searching, text block handling, formatting, and parameter setting functions.

3.1 ENTERING COMMAND MODE

To enter Command mode from Screen mode, type an ESCAPE. The screen clears, and several lines of text appear at the top of your screen. These lines of text give you the following information:

Flag Status	- see Section 3.13, "PARAMETER SETTING COMMANDS"
File Name	- name of file you are editing
Free memory	- amount of available space in memory
Margin	- current left margin
Page	- current page number

If the VUE intialization file tells VUE to print the Menu (see Section 4.4.2, "Help"), a summary of the VUE commands is displayed every time you enter Command mode. (You may enable or disable the Menu display by changing INI.VUE.) VUE then prompts you with a >. You may now enter commands to VUE. End all commands with a RETURN. You must begin all commands in the first character position after the prompt symbol.

You may return to Screen mode at any time by typing an ESCAPE.

NOTE: In the discussions below, the term "FILENAME" refers to the name of the file you are editing.

3.2 COMMAND FORMAT

All VUE commands are in the following format:

command blank(s) {command parameter}

At least one blank must separate the command from any parameters if they are present. If a numerical argument is missing, VUE assumes that it is zero. NOTE: You may abbreviate some of the Command mode commands (see Section 3.14, "ABBREVIATIONS").

If VUE does not understand your command, it prints:

What?

on the line below the command line; it does not erase your command. You may re-edit the command for another attempt.

3.3 COMMANDS THAT ACCESS THE DISK

VUE has eight commands that perform disk access. Four of these commands access two files on the disk: thé original source file, and the output file produced by the editing session. The other four commands may access other files as well. During the editing session, lines of text may be read into memory from the source file, or written to the output file.



VUE creates the output file at the beginning of the editing session. It is given the same name as the source file but an extension of .TMP. When the editing session is complete, the output file is renamed to match the extension of the original source file; the source file's extension is then changed to .BAK (to indicate a backup file). This way, the source file is not modified unless the editing session is successful, and you always have a copy of the original source file after each editing session. (NOTE: Because of this backup facility, you cannot edit files with extension .BAK or .TMP)

Normally, when VUE is first executed it reads in as much of the source file as it possibly can. If the file does not fit into memory, about 500 free bytes are left. If desired, you can disable this initial load with the "/noyank" option when you invoke VUE (see Chapter 4, "EXECUTING VUE").

Sometimes it is necessary to edit files that are larger than the memory available. Usually in these cases, it is best to try to break the file up into smaller pieces. But if this is not possible, the file may be edited by
reading part of the file into memory with the Yank command, editing it, and then writing this to the output file with the Unyank command. This procedure is repeated until the entire file has been edited.

3.3.1 Finish

The Finish command tells VUE that you are done with your editing session. VUE writes out to disk any text still in memory, and makes sure that any leftover text in the source file is copied to the output file. VUE deletes any file on the disk with the name FILENAME.BAK (i.e., a backup file on the disk that has the name of your source file). VUE then renames the source file to FILENAME.BAK, and renames the temporary output file to FILENAME.EXT, where "FILENAME" is the name of the file being edited, and ".EXT" is the extension of that source file.

3.3.2 Save

The Save command allows you to update your text to the disk without leaving VUE. It is equivalent to leaving VUE via the Finish command and then re-editing the file. The Save command only works on files which completely fit in memory. After the Save operation is completed, the internal operation of VUE is unaffected. In other words, the cursor location and flag settings are not modified.

3.3.3 Go

Go is an Edit-and-go command. First the text is output as in the Finish command. Then, depending on the extension of the file, the appropriate processor is invoked. For instance, if an assembly language file is being edited (extension .MAC) the file is automatically assembled. The following is a list of the extensions supported and the command strings invoked. If a command string has been defined in the INI.VUE file, it will be used if the extension of the file being edited is the same as the default extension.

-MAC	>	MACRO filename
.BAS	>	COMPIL filename
_TXT	>	TXTFMT filename
- CMD	>	FILENAME.CMD
.LSP	>	LISP Y ;loads in extended library (DSKIN "filname")
PDL	>	PDLFMT filename
.INI	>	LOG DSK0:1,4 ;this reboots the system MONTST SYSTEM,FILENAME.INI

-

AlphaVue 2.4 Status: space fold srchfold text

Editing OEFIL_BAS	<u>12245 bytes free</u>	margin O	Page 20
<u>>g</u>			
.COMPIL OEFIL			

3.3.4 Quit

This command returns you to AMOS command level without updating your file. It is useful if you've made irretrievable editing mistakes during the current editing session. The temporary output file is erased, and the original source and backup files are not disturbed.

3.3.5 Yank {n} or Yank {filename}

This command has two forms. The first form accepts a numeric parameter. In that case, n lines of text from the source file are read and appended to the end of the text already in memory. If the number of lines is not specified, as many lines will be appended as will fit into memory.

Example:

>yank 50 RET

 read the next 50 lines from the input file on disk and append them to the text in memory. The second form of the command accepts a filename rather than a number. In this case, the specified file is read in and inserted into the text at the current cursor location. If the extension of the file is not specified, it defaults to the extension of the file being edited. This command is useful for combining files, and for "boilerplate" type applications, where a number of pre-prepared blocks of text are merged together to form a custom document.

Example:

>yank info.txt (RET)

- read in file INFO.TXT and insert it after the cursor.

3.3.6 Unyank {n} or Unyank {filename}

The Unyank command also has two forms. If a number is specified, this command writes out the first n lines of text in memory to the output file and deletes them from memory. If the number of lines is not specified, all the text in memory is written out to the disk.

Example:

>unyank 200 @ET)

~ write the first 200 lines of text in memory to the output file and delete them from memory.

If a filename is specified as the parameter, VUE writes a block of text out to the specified file. The filename that is specified as the parameter must not begin with a number. If the extension of the file is not specified, VUE will use the same extension as the file currently being edited. If the file already exists, it is deleted and any previous contents are lost. The block to be written out is specified by standard block delimiters (see Section 3.5, "BLOCK OPERATIONS").

>unyank sub1_mac (RET)

- write out the marked block to the file SUB1.MAC.

If the proper drivers are loaded into memory before executing VUE, any device may be accessed. For instance, you may directly output a marked block to a printer named LISTER with the command:

>unyank trm:lister (RET)

COMMAND MODE

3.3.7 Dir

The Dir command is used to find out what files are on the disk. It is similar to the DIR command at AMOS command level. If no filename is specified, all files with the same extension as the file being edited on the current account are listed. Wildcard filenames are supported; however, wildcards are not allowed in the drive or PPN specifications, and only one file specification is allowed — multiple filenames separated by commas are

Example: assume editing PROG.BAS

≥dir *.* (RET)

- list all files in current account

>dir RET

- list all .BAS files in the current account

>dir dsk0:*.ini[1,4] (RET)

- list all .INI files in System Account

3.3.8 Erase

This command is used to erase files on the disk. It is usually used to erase temporary files created with the Unyank command, or to free up space if you are using a floppy disk system. If the extension is omitted, VUE assumes the extension of the file you are editing. The erase command does not support wildcards.

≥erase sub1_mac (RET)

- erase the file SUB1_MAC from the disk.

3.4 STRING SEARCH AND STRING REPLACE COMMANDS

VUE has five commands for locating and replacing strings. The string to be located or replaced is specified following the command as with any command that accepts a parameter. If a replacement string is required, VUE prompts you for the second string with a question mark. Always end your search and replacement strings with a RETURN.

There are two possible modes for the string commands. These are determined by the SEARCHFOLD flag. If this flag is true, the search ignores case

(upper or lower case) when searching for a string. In other words, if you tell VUE to search for the string "text," the strings "text," "Text," and "TEXT" all match. If the SEARCHFOLD flag is false, this feature is disabled, and the only string which matches "Text" is "Text."

The SEARCHFOLD flag is also checked when a string is replaced. If the flag is false, the replacement string is put into the file with no modification. If the SEARCHFOLD flag is true, then before the old string is replaced, it is checked to see if it starts with an upper case character. If it does, the new string will also have its first character capitalized. If all the characters in the old string were capitalized, all the characters in the new string will be capitalized.

Example: replace "chain" with "link"

	SEARCHFOLD true	SEARCHFOLD false
"chain">	"link"	"link"

NOTE 1: The search and replace strings may not exceed 40 characters in length, and may not contain embedded carriage returns. However, a blank in the search string will also match a tab or carriage return. Therefore, search will find multiple word strings which are on more than one line.

NOTE 2: The search and replace commands are affected by the operation of the SBLK flag. If this flag is set, and if a block is marked in the text, only strings that are within the marked block will be affected by the search and replace operations. This is especially useful with the Global command.

3.4.1 Search

Type Search, the string of characters you are looking for, and a RETURN. Remember to separate the Search command and the search string with a space. VUE searches in the text for the string you've specified; if it finds it, VUE displays the page on which the string was found. If it didn't find it, VUE displays the message: string not found in file. The search always starts at the beginning of the file. This command has two side effects:

- 1. It returns to Screen mode.
- The specified string will be used for subsequent Next Match operations (see Section 2.3.12 "Next Match: Control-X").

Example:

>search LABEL: (RET)
- searches for the string "LABEL:"

3.4.2 Next

The Next command is similar to the Search command. However, instead of starting the search at the beginning of the file, the Next command starts at the current cursor position.

Example:

<u>>next LABEL: (RET)</u>

 searches for the string "LABEL:", starting at the current position.

3.4.3 Whole

The Whole command is also similar to the Search command. As with the search command, it begins searching for the specified string at the beginning of memory. However, if it does not find the string, it checks to see if any more text can be brought in with the Yank command. If so, it automatically writes all the text in memory to the output file, and then yanks in as much additional text as will fit. It then resumes the search operation. This sequence continues until Whole either finds the string or runs out of text to read in.

Note that this command does not affect the operation of the Next Match (Control-X) command in Screen mode. In particular, the Next Match command never automatically yanks in more text from disk, but only searches in the current memory buffer.

If the Whole command does not find the string, you are left with the end of the file in memory. If you wish to return to editing the beginning of the file, you must leave VUE with the Finish command and re-edit the file. 3.4.4 Replace

This command replaces strings in the text. Before replacing the string, it displays it and asks whether or not this string should be replaced.

Type the Replace command, a space and a string of characters to search for; then type a RETURN. VUE prompts you with a question mark for the characters you wish to replace the first string with, if it is found. Type the replacement string and a RETURN. If the string to be replaced does not exist in the file, VUE returns you to Command mode; otherwise, it displays the first occurrence of the search string. VUE now waits for you to tell it what to do; you must type one of these four options:

Y : yes, replace the string and look for next occurrence

N : advance to next occurrence without replacing this string

Control-C or

ESC or

Q : return to Command mode without replacing this string

C : center this string on the screen, then wait for a Y, N or Q

The cursor advances through the file until there are no more occurrences of the search string, or until you enter a Q, ESC, or Control-C.

Example:

>replace chain (RET) ?link (RET)

> will search through file for string "chain," and replace it with "link" if you tell it to do so.

When entering the replacement string, either an ESC or Control-C cancels the Replace command and returns you to Command mode.

3.4.5 Global Replace

This command is identical to the Replace command; however, it does not ask you whether or not to replace each string. It replaces every occurrence of the search string in the file. After VUE is done, it lists the number of strings that it replaced. Use this command with great care; you can easily mangle your file if there are occurrences of the search string that you don't know about. For instance, if you issue a Global replace for "is" to "was," VUE also changes the word "this" to "thwas."

>global chain (RET) ?link (RET)

 replaces every occurrence of the string "chain" with the string "link."

The Global command is especially useful in conjuction with the Block Operations. If the SBLK flag is true, Global will only replace text within a marked block. This makes it easy to replace all occurrences of a string within a limited portion of your text.

3.4.6 Wildcard

VUE allows you to search for strings containing ambiguous characters, i.e., wildcards. A wildcard is a special symbol that may match several characters. The wildcard facility is somewhat similar to that used by some of the AMOS level commands. Normally, a question mark corresponds to any character, and an asterisk corresponds to any number of characters. For example, the search string "(??)" matches the strings:

(RO)

(AB)

(IN)

while the search string "(*)" matches those strings as well as the strings:

(index) (B) (days of future passed)

Leading and trailing asterisks are not allowed in a search string.

If you need to search for a string that contains a question mark or asterisk, you may change the wildcard characters. To do this, use the WILDCARD command. Type WILDCARD, followed by a blank, followed by the two new wildcard characters. If no characters are entered, the wildcard feature is turned off.

≥wildcard %# (RET)

- set the single wildcard character to % and the multiple character wildcard to #

>wildcard RET

- turn off wildcard ability.

>Wildcard ?* RET

- this is the normal wildcard mode.

Although wildcards may be used in replace and global operations, they are not recommended, since wildcards are not allowed in the replacement string.

3.5 BLOCK OPERATIONS

VUE provides several commands for manipulating blocks of text. The techniques for delimiting blocks are covered in Section 2.4.5, "Setting Block Markers".

When a block command is executed, the file must contain at least one valid pair of markers, and the cursor MUST NOT be in between these markers. If one of these conditions is not met, the message "marker error" is displayed and the file is not modified.

If you are using the [*,*] delimiting symbols, and more than one pair of markers exist in the text, only the first pair is recognized by the block commands.

In addition to the commands described in this section, the Unyank and Search commands can also use marked blocks.

To perform a block operation, enter Screen mode. Mark the block of text, following the instructions in Section 2.4.5. If you are performing a COPY or MOVE block, move the cursor to the desired location. Then enter Command mode and type in the block command, followed by RETURN.

3.5.1 Copy Block

This command copies a block of text. Type COPY followed by a RETURN. The block is copied in front of the current line (the line the cursor was on before Command mode was entered). The copy command may be executed multiple times without leaving command mode; the block will be recopied each time. If you copy 5 times you will end up with six copies of the block (the original and the five duplicates). When you are done copying, be sure to clear the reduced intensity markers or remove the [*,*] delimiters.

3.5.2 Delete Block

This command deletes a block of text. Type DELETE followed by a RETURN. The block to be deleted is specified as in the paragraph above. The reduced intensity markers or the [*,*] delimiters are cleared or deleted along with the block. The user is returned to Screen mode after VUE deletes the block of text.

3.5.3 Move Block

This command moves a block of text. Type MOVE followed by a RETURN. The block to be moved is specified as in the example above. VUE copies the block to the current cursor position, and then deletes from the file the marked block along with the delimiters. This command is equivalent to a Copy command followed by a Delete command. VUE returns you to Screen mode after it moves the block.

3.5.4 Clear

The Clear command deletes any block markers that have been set with Control-P. If no markers have been set, this command is ignored. (It does not clear any [*,*] delimiting symbols.)

3.6 FORMATTING COMMANDS

VUE provides two commands to assist in formatting text. These commands are not meant to replace TXTFMT; rather, they are designed for use in simple applications where the power of TXTFMT is not needed.

3.6.1 Center

This command centers the text on the current line (the line the cursor was on before Command mode was entered). Leading blanks are ignored, trailing blanks are not. VUE returns you to Screen mode. 3.6.2 Format

This command formats the paragraph starting at the current cursor position. The text is formatted so that no line is wider than the screen. If the text is in a comment, the comment characters are not formatted, but the text is. (In other words, this command works properly on comments.) VUE formats from the current cursor position to the end of the paragraph.

If not formatting a comment, a paragraph is terminated by -

- 1. A blank in column one
- 2. A slash (/) in column one

If formatting a comment, a paragraph is terminated by -

- 1. A blank in column two
- 2. A character other than the current comment character in column one

To determine if it is formatting a comment, the format command checks the character in column one of the first line formatted. If it is equal to the current comment character, then it assumes that a comment is being formatted.

After VUE formats the text, it returns you to Screen mode.

3.6.3 Width

The Width command sets the width used by the Center and Format commands. Normally the width used is the width of the terminal display. One numeric argument is specified, which is the new width. If no argument is given, the formatting width is reset to the width of the terminal display. Example:

>width 50 RET

set a narrow formatting width

3.7 ENVIRONMENT COMMANDS

VUE provides two commands for saving and restoring the current editing environment. Often when editing a file, it would be convenient to "mark one's place" at the current point, and move somewhere else for a quick look or modification. Then, when this is done, it should be possible to return to the original editing point and continue from there. VUE provides this capability with the Push and Pop commands.

3.7.1 Push

The Push command "pushes" the current cursor location onto an internal stack. This stack may be up to ten levels deep. Push then returns you to

3.7.2 Pop

This command "pops" a cursor location from an internal stack. It restores the location marked earlier by a Push command. VUE then returns you to Screen mode at the restored position. (That is, you see the page displayed that you "pushed" earlier with the Push command.) If the stack is empty, this command has no effect and you remain in Command mode.

3.8 MEMORY SPLIT COMMANDS

Often when editing a long file, VUE takes a long time to perform operations which update the file. In order to speed up this process, VUE allows you to split the memory image of the file in two parts. After the file is split, only the first part of it can be edited. However, operation is much faster. This is especially useful when adding new text near the beginning of an existing document.

3.8.1 Split

The Split command splits the current memory image at the cursor location. When you return to Screen mode, the line the cursor was on appears to be the last line of the file. While the file is split, the word "split" appears on the Status Line and lines of &'s appear as the end of file marker instead of the usual asterisks.

3.8.2 Unsplit

The Unsplit command performs the exact inverse of the Split command. It reattaches the text that was detached with the Split command. If no text has been split off, the Unsplit command has no effect.

The Unsplit command is executed automatically before any disk transfer operations are attempted. Specifically: the Finish, Go, Save, Yank, and Unyank commands all perform an automatic Unsplit before they begin.

3.9 EDIT

The Edit command exits Command mode and returns you to Screen mode. VUE displays the page you were editing when you entered Command mode. This command is equivalent to typing ESC while in Command mode.

3.10 PAGE N

The Page command allows you to move the cursor to a specific screen page. VUE returns you to Screen mode, and moves the cursor to the page of the number you've specified. If there is no sign in front of the number, the cursor is moved to the absolute page specified by the number. If there is a sign in front of the number, the number is taken as an offset from the current page.

Examples:

>page 5 (RET) - moves the cursor to page 5 >page +3 (RET)

- moves the cursor forward 3 pages

Note that the definition of a page is one screenful (usually 24 lines) of text. This has no relation to any page-numbering sequence which may be used in the final document.

3.11 MARGIN N

This command sets the left margin to n characters. When you resume screen editing, everything you type will be displayed starting at column n. Useful for editing files with many lines wider than the screen display.

WARNING: VUE will not work with lines longer than 510 characters!!

Example:

>margin 50 (RET) - sets left margin at 50

Note that this command simply sets the VUE display margin. It does not change the actual text margin, only VUE's display of the text.

3.12 BLANKS N

This command is used to generate a large number of blank lines. One argument is supplied, which is taken as the number of blank lines. The lines are added at the current cursor position.

Example:

>blanks 20 RET

- insert 20 blank lines at current cursor location

3.13 PARAMETER SETTING COMMANDS

You may modify a number of flags and constants with commands in Command mode. Some of these may also be modified by the INI.VUE file. The parameter setting commands follow one of the formats below:

Command boolean	The argument of this command may be "true," "t," "false," or "f" (e.g., Insert true). A true value sets the flag; a false value disables it.
_	

- Command string The argument of this command must be a character (e.g., Comment !).
- Command n The argument of this command is a numerical value (e.g., Column 25).

3.13.1 Text boolean

This command turns Text Mode on/off. For a complete description of Text Mode, see Section 2.5.2, "Text Mode."

3.13.2 Wrap boolean

This command turns Wrap Mode on/off. When Wrap Mode is on, VUE automatically supplies carriage returns during text entry. For a complete description of Wrap Mode, see Section 2.5.3, "Wrap Mode."

>wrap true RET

- turn on Wrap Mode

3.13.3 Searchfold boolean

This command turns the SEARCHFOLD flag on and off. For a detailed explanation of what this flag does, see Section 3.4, "STRING SEARCH AND STRING REPLACE COMMANDS." When this flag is on, the word "srchfold" appears on the VUE Status Line when in Command mode.

3.13.4 Sblk boolean

This command enables/disables searches only in marked blocks. If Sblk is set to true, search operations only locate strings within marked blocks (if such blocks exist).

3.13.5 Indent boolean

This command enables/disables Indent Mode. For a detailed description of Indent Mode, see Section 2.5.5, "Indent Mode." When Indent Mode is on, the word "indent" appears on the VUE Status Line.

3.13.6 Help? boolean

This command enables/disables the printing of the menu when you are in Command mode; it has no effect if the menu was not enabled in the INI.VUE file.

3.13.7 Comment character

This command sets the current comment character used by Entry Mode and the Format command.

>comment ! RET

 this sets the comment character as the exclamation point (as in BASIC)

3.13.8 Column n

This command sets the current comment column used by Entry Mode; we recommend that this number be a multiple of eight.

Example:

≥column 56 RET

- this command sets the comment column to 56.

3.13.9 Field n

This command defines the next-field character used by Entry Mode. The number n is the decimal value of the ASCII code of the character to be used. For example, space is 32 and tab is 9.

Example:

>field 9 RET

- this sets the next-field character to tab.

3.13.10 Insert boolean

This statement enables/disables the insertion of a new line after every RETURN (see Section 2.5.4, "Entry Mode"). When this flag is on, the word "insert" appears on the VUE Status Line when in Command mode.

≥insert true (RET)

- enables line insertion

3.13.11 Fold boolean

This command enables/disables the comment folding feature in entry mode. When this flag is on, the word "fold" appears on the VUE Status Line when in Command mode.

3.13.12 Space boolean

This command enables/disables the generation of a space after every comment character in Entry Mode. When this feature is on, the word "space" appears on the VUE Status Line when in Command mode.

3.13.13 Delta n

This command sets a new autoincrement line number value when editing BASIC files in Entry Mode. If no value is supplied, the command is ignored. Otherwise, the autoincrement value is set to the new number.

Example:

>delta 50 RET

- set autoincrement to 50

3.13.14 Control boolean

This command enables/disables control characters. If this command is set to true, control characters may be entered into the text with the Control-G screen editing command.

4

3.13.15 Smart boolean

This command enables/disables the intelligent terminal features of VUE. If you use a terminal with insert and delete line functions, turning on this flag will improve some of VUE's screen handling abilities. If your terminal does not have these features, enabling this flag will cause problems. VUE now supports a new terminal driver format that makes the Smart command unnecessary. However, old terminal drivers may still need to use the Smart command.

3.13.16 Abbreviations

You may abbreviate several of the command mode commands to their first letter:

f - finish
g - go
q - quit
s - search
r - replace
y - yank
u - unyank
m - margin
p - page
e - edit
d - dir

CHAPTER 4

EXECUTING VUE

This section tells you how to call VUE from the AMOS command level. After you invoke VUE, the first thing it does is to print a sign-on message. It then searches for an initialization file (INI.VUE); VUE processes the file if it exists. Next VUE looks for the file specified on the command line. If it finds it, VUE reads the file into your memory partition. As the file is read in, VUE prints a period for every 64 lines that it reads in from the file. While the file is being read in, you may interrupt VUE and return to AMOS command level by pressing Control-C. When VUE finishes reading in the text, it displays the first page of text (if the INI.VUE is set up so that you enter in Screen mode), and you can begin to edit your text.

If the desired file is not found, VUE asks if you want to create the file. If you enter Y and a RETURN, VUE creates the file and you can begin to edit. If you answer anything else, VUE returns you to AMOS command level.

4.1 INVOKING VUE

The format for invoking VUE is as follows:

.VUE {/options} FILENAME{.EXT} (RET)

Parameters in curly brackets are optional.

4.1.1 Options

The only option presently supported is the /noyank option. This option disables the reading in of the input file when VUE is invoked. VUE then automatically comes up in Command mode so that you may yank in a specific number of lines.

.vue /noyank letter.txt (RET)

 this invokes VUE but does not cause any text to be read in.

4.1.2 Filename

Filename specifies the name of the file you are editing. If not present, a FILE SPECIFICATION ERROR results.

4.1.3 Extension

Extension specifies the extension of the file being edited. If the filetype is missing, VUE uses the default extension of .MAC. (You may change the default extension in the VUE initialization file.)

4.2 ERROR RECOVERY

If an error occurs while writing out the output file (for instance, the disk is write protected), VUE gives you a chance to recover. It prompts you with an error message describing the error and indicates that a copy of your text is stored in a memory module IMAGE.VUE. This module may be saved using the SAVE command from AMOS command level, and then re-edited by VUE. NOTE: The module does not contain any linefeeds, and will look odd if you try to type or print it. Re-editing it with VUE restores the linefeeds.

4.3 INITIALIZATION FILE (INI.VUE)

When VUE is invoked, it looks for a file named INI.VUE. If found, this file is read in and processed. The file consists of a series of assignment statements which modify VUE parameters. This allows you to "personalize" VUE. VUE searches for INI.VUE in three PPNs. Assuming that your ppn is EP,PN], the search order is: 1: MEM: (searches in user memory partition)

2: [P,PN] (searches in your account)

3: [P,0] (searches on project library account)

4: [7,0] (searches on system library account, DSKO:)

The format of the assignments is:

parameter=argument

The assignments may be in upper or lower case. In each case, the parameter is assigned the value of the argument. The assignments may be in any order, and any parameter not assigned a value will retain its default value. Each assignment must start on the first character of a line. Several parameters require boolean arguments. A boolean argument is either "true" or "false," or any truncation of these words. For example, "t," "tr," and "true" all represent the boolean value TRUE.

If desired, a :T may be placed into the INI.VUE file. This will cause all lines following to be echoed to the console as they are processed, just as with .CMD and .DO files. Any line beginning with a semicolon (;) is ignored.

4.3.1 Default=extension

This assigns the default extension. The extension must consist of three valid RAD50 characters (uppercase only). The default extension if none is assigned in the INI file is .MAC.

Example:

default=TXT

This statement makes the default extension .TXT.

4.3.2 Help=boolean

This statement enables the display of a help menu in Command mode if HELP is assigned a value of true. If enabled, the available editing space is reduced, because the menu takes up some of your editing room. When this statement is processed, VUE searches for a file named MENU.VUE in the same way that it searched for INI.VUE. (However, VUE does not search for MEM:MENU.VUE.) If found, VUE reads the menu into memory. Each time you enter Command mode, the menu is displayed. Normally, the menu contains a complete reference of VUE commands, but the user may change the file if he would like to modify or replace the menu. The value of HELP defaults to false if the HELP statement does not appear in the .INI file.

help=true

This statement enables the display of the command menu.

4.3.3 Start=location

This statement determines the cursor starting location of VUE after the file is read in. Normally, if this statement is not in the INI file, VUE enters Screen mode at the beginning of the file as if the HOME key had been hit. This statement allows VUE to start up in three locations:

if location=home VUE starts up normally at the home position in Screen mode.

- if location≕end VUE starts up in Screen mode at the end of file.
- if location=command VUE starts up in Command mode.

Example:

Start=command

This statement causes VUE to enter Command mode when started.

4.3.4 Go=command string\$

This statement allows the user to specify the command string to be executed when the GO command is invoked (see Section 3.3.2, "Go"). The command string may be any sequence of characters including RETURNS and linefeeds. The string is terminated by a dollar sign (\$). The percent sign (%) is treated as a special character. It is replaced by the name of the file being edited. The new command string will only be used if the file you are editing has the current default extension.

Example:

Go=COMPIL %\$

This statement causes the file being edited to be compiled by the BASIC compiler if the GO command is used and if the extension of the file being edited is the same as the default extension.

EXECUTING VUE

NOTE: The GO command string cannot be longer than 100 characters, but can be more than one line. For example:

go=TXTFMT HEADER,% RENAME/D %=HEADER.LST\$

4.3.5 Text=boolean

This statement enables/disables Text Mode. Normally, VUE begins with Text Mode off. However, when editing a file with an extension of .TXT, Text Mode defaults to true.

4.3.6 Wrap=boolean

This statement enables/disables Wrap Mode. If this statement does not appear in the INI.VUE file, Wrap Mode is initially off.

4.3.7 Comment=character

This statement sets the comment character recognized by VUE in Entry Mode (see Section 2.5.4, "Entry Mode"). The default character is the semicolon (;), unless a .BAS file is being edited, in which case it is the exclamation point (!).

Example:

comment=!

This sets the comment character to the exclamation point, which is used by BASIC.

4.3.8 Column=comment column

This statement sets the normal starting column for comments. This value is used in Entry Mode (see Section 2.5.4, "Entry Mode"). The default value is 40. (The default is 56 if you are editing a BASIC file.) The column is specified as a decimal number. For best results, multiples of eight are recommended.

column=64

Sets the comment column to 64.

4.3.9 Space=boolean

This statement generates a space after the comment character in Entry Mode (see Section 2.5.4, "Entry Mode"). The default value is true.

Example:

space=false

This disables the generation of a space after the comment character.

4.3.10 Insert=boolean

This statement tells VUE to insert a new line every time a carriage return is hit in Entry Mode (see Section 2.5.4, "Entry Mode"). The default value is true.

Example:

insert=true

This enables the line insert feature.

4.3.11 Fold=boolean

This statement tells VUE to fold all characters not in comment fields to upper case while in Entry Mode (see Section 2.5.4, "Entry Mode"). The default value is true.

Example:

fold=false

This disables the folding option.

4.3.12 Field=character

This statement sets the next-field character used in Entry Mode (see Section 2.5.4, "Entry Mode"). The character is specified as a decimal number which corresponds to the ASCII value of the character. Normally, this character defaults to a space (decimal 32).

Example:

Field=9

This statement sets the next-field character to the tab key (decimal 9).

4.3.13 Searchfold=boolean

This statement enables/disables the search option of ignoring upper/lower case when searching (see Section 3.13.3, "Searchfold"). The default value is true.

Example:

Searchfold=true

This forces the search command to ignore case.

4.3.14 Indent=boolean

This statement enables/disables Indent Mode (see Section 2.5.5, "Indent Mode"). The default value is false.

Example:

Indent=true

This enables Indent Mode when VUE is started.

4.3.15 Width={number}

This statement sets the text formatting width. Normally, this width defaults to the terminal display width.

4.3.16 Wildcard={wildcard characters}

This statement sets the wildcard characters used in the search commands. If this statement does not appear in the INI file, the default of:

wildcard=?*

is assumed. The statement:

wildcard=

turns off the wildcard feature.

4.3.17 Entry=boolean

This statement allows you to automatically turn on Entry Mode. Normally, Entry Mode is toggled on/off with the Control-\ key in Screen mode. The Entry statement in INI.VUE automatically turns on Entry Mode if set to true.

4.3.18 Sblk=boolean

If set to true, this statement forces searches to only find strings in marked blocks, if they exist. It corresponds to the Sblk command in Command mode.

4.3.19 Delta={number}

This statement allows you to set the autoincrement used for line numbering in BASIC programs. If not set, it defaults to 5.

4.3.20 Control=boolean

If set to true, this statement allows control characters to exist in the file. If a file is to be read in containing control characters, this flag must be set in the INI.VUE file. (NOTE: When editing .LST files, VUE defaults to Control=true automatically.)

4.3.21 Smart=boolean

This statement enables/disables the intelligent terminal screen handling routines. If a terminal has hardware insert and delete line functions (such as an ADM1, ADM2, or Hazeltine 1500) this flag should be set true to enable improved screen handling. The default value is false. If a release 4.4 or later terminal driver is available, this flag is unnecessary.

Smart=true

This enables the intelligent terminal handling.

4.3.22 Dumb=boolean

This statement enables/disables dumb terminal handling. When this flag is enabled, VUE assumes that the terminal does not provide the erase to end of line and erase to end of screen functions, and simulates them for you. This is useful for terminals like the ADM3A. If a release 4.4 or later terminal driver is available, this flag is unnecessary.

4.4 SAMPLE INI.VUE FILE

The following is a sample INI.VUE file that might be used by a BASIC programmer getting acquainted with VUE:

default=BAS help=true go=COMPIL % RUN %\$ delta=10 comment=! field=9

The first line of the file sets the default extension to .BAS, since our hypothetical programmer will be mostly editing BASIC programs. To edit a file named LEDGER.BAS, simply type:

.VUE LEDGER (RET)

VUE automatically fills in the extension of .BAS from the INI.VUE file.

The second line of the file turns on the menu display. Since our programmer is just getting familiar with VUE, the menu will help him to remember the commands.

The third and fourth lines set the GO command string. When VUE is exited with the GO command, the file being edited is automatically compiled and run, if it is a BASIC program.

The fifth line sets the autoincrement line numbering value to 10. This is used when entering new lines of a BASIC program.

The sixth line sets the comment character to an exclamation mark. This is the comment character used by BASIC.

The last line sets the next-field character to the tab key. Normally, the next-field character is space, but space is commonly used in BASIC programs. The TAB is a key not often used in BASIC programs.

Note that the last two lines actually have no useful effect, since VUE now defaults to these settings when editing BASIC files.

CHAPTER 5

HARDWARE AND SOFTWARE REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

The hardware required to run VUE 2.4 is:

- 1. An AM-100 or AM-100/T CPU with at least 48 kilobytes of memory.
- 2. A high speed CRT terminal (recommended 9600 baud or greater) with the following functions -
 - 1. XY cursor positioning 2. Clear screen

these functions are desirable but not necessary -

- 3. Erase-to-end-of-line
- 4. Erase-to-end-of-screen
- 5. Reduced intensity (TCRT functions 11 and 12)

HARDWARE AND SOFTWARE REQUIREMENTS

The software required to run VUE 2.4 is:

1. AMOS version 4.4 or later

2. A terminal driver that supports the following functions:

1. XY cursor positioning

2. Clear screen (TCRT function 1)

3. Erase-to-end-of-line

4. Erase-to-end-of-screen

If necessary, the last two functions may be simulated for dumb terminals which do not have them (Lear Siegler ADM-3A). This may be done in one of two ways. The command DUMB=TRUE may be used in the initialization file to cause VUE to simulate these features. Alternately, VUE now supports a new TDV format which contains a bit indicating that these features do not exist.

Terminals that fully meet these requirements with standard Alpha Micro software are:

Soroc IQ120 Lear Siegler ADM1, ADM2, ADM3A Hazeltine 1500 series Beehive B150

VUE is fully rementrant and may be loaded into system memory if you plan to do a lot of editing.

APPENDIX A

VUE 2.4 RELEASE NOTES

The 2.4 version of AlphaVue contains a significant number of new features. For users already familiar with VUE, this document provides a summary of the changes.

(NOTE: AlphaVue 2.4 is approximately 5000 bytes longer than earlier releases. If you are running VUE in system memory, you may have to allocate more system memory space.)

A.1 NEW SCREEN MODE COMMANDS

Several new control characters have been added. These are:

Control-N: Go to End of Line - This key moves the cursor to the end of the current line in one operation. (See Section 2.2.7.)

Control-RUBOUT: Erase Line - This key erases all the characters on the current line, but does not delete the line. It is equivalent to a Control-U followed by a Control-Y. (See Section 2.2.12.)

Control-P: Set Block Marker - AlphaVue 2.4 has a new way of marking blocks which may be used instead of block delimiters in the text. Control-P is used to mark the ends of the block. While block markers are active, the text in the marked block is displayed in reduced intensity. The old delimiter method is retained for use on terminals that do not support reduced intensity, such as ADM-3As. (See Section 2.4.5, Section 3.5.)

Control-G: Absolute Character Insert - It is now possible to insert most control characters into a file using VUE. (See Section 2.4.6.) 2.7.6

In addition to these control characters, several new modes have been added. When Wrap Mode is active, it is not necessary to enter carriage returns. VUE automatically wraps the the last word entered to the next line when the end of the screen is reached. (See Section 2.5.3.) In addition, Entry Mode has been upgraded for BASIC programmers. It will now perform automatic line numbering if needed. (See Section 2.5.4.5.)

VUE 2.4 RELEASE NOTES

A.2 NEW COMMAND MODE FEATURES

There are now over 40 commands available in VUE Command mode. Some of the more interesting new commands are outlined below. The input processing has been upgraded so that it is no longer possible to edit the prompt character.

A.2.1 Disk Access Commands

Save: This command saves the current memory image as with the Finish command, but does not leave VUE. (NOTE: Save only works on files which fit in memory.) This is useful for periodically backing up your work without having to leave VUE. (See Section 3.3.2.)

Yank and Unyank: These commands have been upgraded to allow you to merge and split files into the current file. This allows "boilerplate" type operations, such as putting together a custom real estate document from a number of standard paragraphs. In addition, the original forms of Yank and Unyank no longer leave blank lines hanging around when editing long files. To specify a yank or unyank from a separate file, follow the command with the name of the desired file, instead of a number of lines. (See Section 3.3.4, Section 3.3.5.)

<u>Dir</u> and <u>Erase</u>: These commands are useful with the new Yank and Unyank commands. They allow you to look at the disk and to erase temporary files. (See Section 3.3.6, Section 3.3.7.)

A.2.2 Searches

Searches now allow for ambiguous search strings (wildcards). New search commands allow searching from the current cursor location and searching files larger than memory. In addition, a number of inconvenient bugs in these commands have been fixed.

<u>Next:</u> This command is similar to the Search command. However, it starts searching at the current cursor location. (See Section 3.4.2.)

Whole: This command searches entire files, even if the file is bigger than memory. (See Section 3.4.3.)

<u>Replace and Global:</u> It is now possible to escape while entering the replacement string by typing Control-C or ESC. In addition, both Control-C and ESC are now accepted to terminate a replace operation, as well as Q. (See Section 3.4.4, Section 3.4.5.)

A.2.3 Block Commands

The block commands have been upgraded with an easier method of block definition. In addition, a block of text may be written to disk with the Unyank command and searches and replacments may be confined to marked blocks. (See Section 2.4.5, Section 3.3.5.)

A.2.4 Formatting Commands

Format and Center operations may now be set to any line width.

<u>Width:</u> This command sets the formatting width. It is followed by a number, which is used as the new width. If no number is specified, the width is reset to the terminal width. (See Section 3.6.3.)

A.2.5 Memory Split Commands

Often when editing a long file, VUE takes a long time to perform operations which update the file. In order to speed up this process, VUE allows you to split the memory image of the file in half. After the file is split, only the first part of it can be edited. However, operation is much faster. This is especially useful when adding new text near the beginning of an existing document.

<u>Split:</u> This command splits the current memory image at the cursor location. (See Section 3.8.1.)

Unsplit: This command is the exact inverse of the Split command. It reattaches the text that was detached with the Split command. If no text had been split off, the Unsplit command has no effect. The unsplit command is executed automatically before any disk transfer operations, such as Finish, Go, Save, Yank, or Unyank. (See Section 3.8.2.)

A.2.6 New Menu Features

AlphaVue now supports a new menu format which allows multiple menus. A number of subjects may be stored in the menu, and called up as needed. To find out what subjects are available, type HELP or MENU. A list of subjects will be displayed. To call up a subject, type HELP or MENU followed by the subject name.

When VUE enters Command mode, a menu of screen editing commands is shown. The second subject menu is displayed after the first command is entered. A RETURN (blank line) may be used to display the second menu.

VUE 2.4 RELEASE NOTES

(.....

 $(\overline{\ }$

A.2.7 Parameter Commands

A number of new parameter commands have been added to control the new features:

Text and Wrap: These commands enable and disable Text and Wrap modes. Wrap Mode was described earlier. Text Mode makes VUE act like earlier versions did when editing .TXT files. (NOTE: Text Mode is automatically turned on when editing .TXT files.) (See Section 2.5.2, Section 2.5.3.)

Sblk: This command, if set true, forces searches to only locate strings in a marked block, if any. This is especially useful with the Global command. A block of text can be marked, and a Global replacement done which only affects the text in that marked block. (See Section 3.13.4.)

Delta: This command sets the auto-increment value used in Entry Mode with .BAS files. VUE comes up with this value set to 5, but it may be reset to any value except zero by this command. (See Section 2.5.4.5.)

A.3 NEW INITIALIZATION FEATURES

The VUE initialization processing has been completely revised internally. It is now possible to break out of VUE while it is initially reading in a file by pressing Control-C. VUE will now operate correctly with INI.VUE in memory, and it will find the menu on the system disk properly. If desired, a :T may be placed in the INI.VUE file so that the INI file is echoed as it is processed, as with CMD and DO files. A number of new initialization statements are available to control some of VUE's new features.

A.4 IMPROVED HARDWARE SUPPORT

VUE now supports a new terminal driver format which allows non-standard screen sizes and automatically determines what features a terminal has.

APPENDIX B

NEW MENU FORMAT

Version 2.4 and later of VUE support a new menu format, which allows for multiple menu screens. The old single menu format is still supported. The old format consists of straight text terminated by a dollar sign.

To generate or modify a menu in the new format, the macro assembler is used. A library file, MENDEF.MAC, is supplied to assist in this process. It defines several macros which are used to define menus, and sets up the header which VUE uses to differentiate the new menu format from the old format.

The actual menu definition consists of three sections. The first is a table of offsets to each menu. The table is terminated by a null word. The second section is a table of names for each menu. Each table entry is defined with the SCREEN macro. The SCREEN macro requires two arguments. The first is the name of the menu. The second is the number of the menu, as defined by the offset table. In other words, the first menu in the offset table is menu 1, the second is menu 2, etc. The name table is terminated with a null byte.

The third section of the menu definition is the actual text of each menu. The text for each menu is normally preceded by a label, which was used in the offset table. Each line of text is defined with the L macro. This accepts one line of ASCII text, and appends a carriage return/linefeed to it. Since most menus contain embedded blanks, the assembler will require angle brackets around the text. The last line of text in a menu is defined by the LL macro. This macro terminates the text with a null byte instead of with a CRLF.

The first menu defined in the offset table is always used by VUE as the screen menu; i.e., it is always displayed when VUE leaves Screen mode. The second menu defined is initially the Command mode menu.

The sources for the standard AlphaVUE menu are included to aid in understanding menu generation.
Index

Auto Line Numbering 2-11 Backspace command 2-2 BASIC files 2 - 11Beginning of Line command . . . 2-2 Blank Line Generation 3~16 Block delimiting symbols 3-11 Block Operations 1-13, 2-7, 3-11 2 - 6Carriage return 2 - 6Center command 3-12 Center Cursor command 2-5 Character insert mode 2-7, 2-9 3-12 4-5 Command abbreviations 3-20 Command format 3-1 Command mode 1-9, 3-1 4-5 Concatenate Line command 2~6 Control-characters 1-1 Form feed 1-7 Control-characters in file ... 2 - 8Control commands 2 - 62 - 6ESCAPE 2-6 Toggle character insert mode . 2 - 7Toggle entry mode 2-7 Copy Block command 3 - 11Cursor 1-4 Cursor control commands 2~1 Backspace 2 - 2Beginning of Line 2 - 2Center Cursor 2-5 Down 2-4 End of Line 2-3 Ending Page 2-5 . . Forward 2-1

ALPHAVUE USER'S MANUAL - INDEX

Go to Next Word	• • • • • • 2-3	
Go to Previous Word	1 • • • • • 2-3	
Home		
Page Down	2-5	
Page Up	2-5	
Up	2-4	
,	••••••	
Delete Block command	•••••	
Delete Line command .		
Delete to End of Line	command 2-3	
Delete Word command	2-3	
Deleting characters	2 - 1 + 2	_7
Deleting lines		-5
Deleting words	•••••••	
Directory compand	• • • • • • 2-3	
Diek sass	• • • • • • 3-6	
Disk access commands	•••••• 3-2	
V1r	• • • • • • 3-6	
Erase		
Finished	• • • • • • 3-3	
Go • • • • • • • • •	3-3	
UnYank		
Yank	3-4	
Down command	2-4	
	•••••	
Edit command	3-15	
End command	2-5	
End of Line command		
Entry mode	••••••	
Fields	••••••	
Folding	•••••	
lino incont	•••••2-11	
	•••••2-11	
	••••• 2-10	
Den Rent commands	••••• 3-13	
	• • • • • • 3-14	
Push	• • • • • • 3-14	
Erase Command	• • • • • 3-6	
Erase Line command	•••••	
Erasing lines	• • • • • • 2-4	
Errors	4-2	
ESCAPE	2-6	
Extension	• • • • • • 4-2	
File editing commands .	•••••	
Filename	4-2	
Files	1-3	
Finish command		
Folding	2 2 11	
Form feeds		
Format command	• • • • • • • ⁻ (7 47	
e e e e e e e e e e	•••••• 5~15	

Page Index-2

e.

Ć

ALPHAVUE USER'S MANUAL - INDEX

.

Forn Ce Fo Wi Forv GLob	natt ente orma idth varc oal	r at c Re	g om pla	co mai ac:	nm nd	an • • •	ds • •	an		• • •	•	•	•		3-12 3-12 3-13 3-13 2-2 3-10
Go Go d Go t Go t	omm oN	an lex 're	d t l vid	lo lou:	rd s (co	om: rd	na c	nd om	mai	nd	•	•	•	4-4 3-3 2-3 2-3
Help Home	• co	• M M A	ano	ł	•	•	•	•	•	•	•	•	•	•	4-3 2-4
Inde INI. Inse Inse Inse Invo	nt VUE rt rti kin	mo con Lin ng g	de mma ne ch VUE	and co nai	d omr ac	nar te	nd ers		•	•				•	2-12 4-2 2-2 2-5 2-1 4-1
Line	le	ngt	:h	•	•	•	•	•		•	•		•	•	2-1
Marg Maxi Memo Sp Un Menu Merg Move	in mum ry lit spl ing Bl	con Li Spl it Fi	nma ine Lit	inc linc c c s	i on • •	ngt ma			•	• • • • • •		•	•	•	3-15 1-7 3-14 3-14 3-14 1-5 3-5 3-12
Next Next Next	co fi Ma	mma elc tch	and d c	- 017	ma	Ind	•	•	•	•	•	•	•	•	3-8 2-10 2-6
Optio	ons	•	•	•	•	•	•	-	•	•	•	•	•	•	4-1
Page Page Page	cor Dov Up	nma vn co	nd co	mm an	• an d	• d	•	•	•	•	•	•	•	•	3-15 2-5 2-5

Page Index-3

Parameter	` set	tin	g c	omr	nanc	ls			•	3-16	
Column					• •				-	3-18	
Comment		•		-				-	-	3-17	
Control				•					-	3-19	
Delta .							_	-	-	3-19	
Field .		•						-		3-18	
Fold .				-				-	•	3-10	
Help .				-	• •	-	•	•	•	3 17	
Indent				•	•••	•	•	•	•	3-17	
Insert				•		•	•	•	•	3-17	
Sblk -				•	• •	•	•	•	•	7-17	
Searchf	<u>ојч</u> .	• •	•	•	• •	•	•	-	•	2 17	
Smart	otu	• •	•	•	• •	•	•	-	•	3-17	
Space	• •	• •	•	•	• •		۰	•	•	3-19	
Toxt		• •	•	•	• •	•		•	•	3-19	
iext _	• •		•	•	• •	•	•	•	•	3-16	
wrap .	• •	• •	-	•		•	•	-	•	3-17	
Pop comma	nd 🚬	• •	•	-	• •	•		•		3-14	
Push comm	and		•				•		•	3-14	
Quit comm	and									,	
	anu	• •	•	•	• •		•	•	•	5-4	
Replace St	tring	1	m m s	and						70	
Replacing	chai	9 2 2 4 4	0.000		•	•	•	•	•	5-9	
RETHEN	Cirai	αςι	era	•	•••	•	•	•	•	5-9	~ ~
PHOANT	• •	• •	•	•		•	•	•	٠	1-2,	2-6
. 10000	• •	• •		•	• •	•	•	•	•	2-2,	2-9
Save comm	and										
Screen mod	anu Ao	• •	•	•	• •		٠.	•	•	5-5	
Screen-odd	18 . + 1	•••	• 	. •	• •	•	•	•	•	2-1	
Screen-eur	i t i ng	u mo	aes	5	• •	•		•	•	2-8	
Schunch co	ommar		•	• •	• •	•	•	•	•	2-2	
Search con	imano	•	•	•	• •	•	•	•	•	3-8	
Next mat	ch	• •	•		•		•	•	•	2-6	
Searching	tor	cha	rac	ter	`S	•	•	•	•	3-8	
Split comm	and		•	• •		•	•	•		3-14	
Splitting	file	s.	•			•		-	•	3-5	
Splitting	line	s.	•		•	•	•	•	-	2-5	
TAB .	_	_					÷			17	
Text files	• •	* •	•	• •	•	•	•	•	•	2-0	
Text Mode	•	••	•	• •	•	•	•	•	•	2-7 /_5	
TEAC HOUE	• •	• •	•	•••	•	•	-	•	•	470	
Unsplit co	mman	d.	•							3-14	
Unyank 🔒						-		-	-	3-5	
Up command			-	• •	•	•	•	•	•	2-4	

4

٧l	JΕ	С	on	n m	an	ıd	٤	in	е									4-1	
٧l	JE	i	ni	it	ia	ili	iza	at	io	n	fi	le						4-2.	4-9
	Co	sι	ur	nn				-										4-5	
	Co	сm	me	۶n	t												-	4-5	
	Co	on	tı	۰٥	ι	•	•			•	-		•					4-8	
	De	ef	aı	ı١	t		•	•					•	•				4-3	
	Fi	ie	ι	ł		•	•	•		-	•							4-6	
	Fc	sι	d		•		•	•										4-6	
	Go)	•	,	•	•		•	•	•			•	-	•		•	4-4	
	He	۱٤	р		•	•		•		-	•	•		•	•			4-3	
	In	۱d	er	۱t		•		•					•		-	•		4-7	
	In	۱S	er	t		•	•	-	-			•		•	•		•	4-6	
	Se	a	rc	h	fo	lc	ł	-	•	-	•	•			•		•	4-7	
	Sm	ıa	rt			•	-		•									4-8	
	Sp	a	се		-	•	•	•			•		-				•	4-6	
	St	а	rt			•	•			-								4-4	
	Te	x	t		•													4-5	
	Wr	a	р			•	•		•		•				•			4-5	
Wh	οl	e	С	o	mm	an	١d											3-8	
₩r	ap)	Мо	d	е						•		-					4-5	
												•							
Ya	nk		co	m	nai	nd				•.							•	3-4	