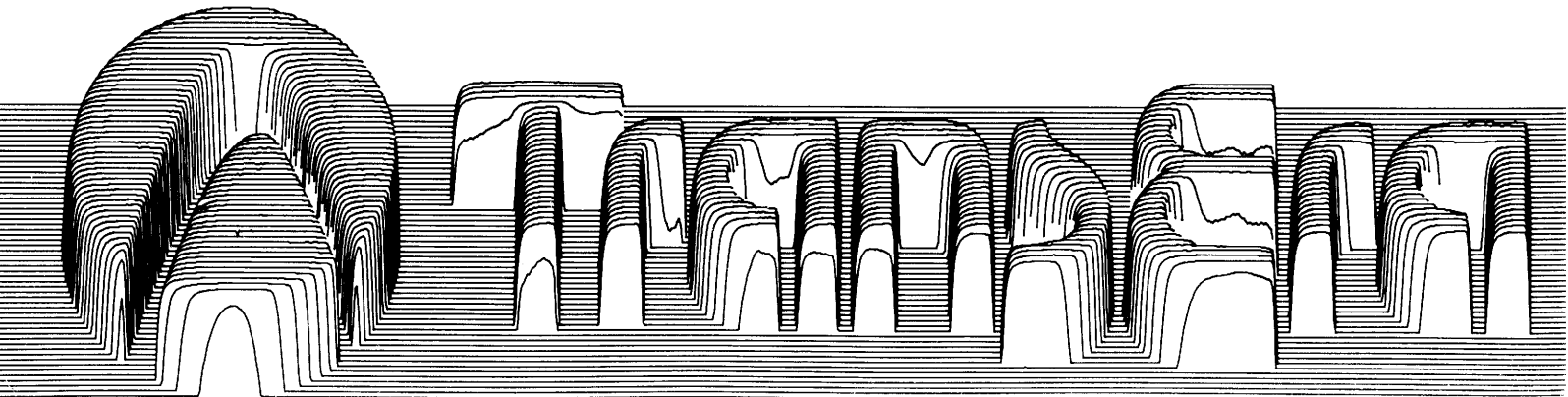


Editor/Word Processor
Operator's Manual



Editor/Word Processor Operator's Manual

TABLE OF CONTENTS

SECTION 1	General Description	
	Introduction	1-1
	ROM Pack Installation Instructions	1-1
SECTION 2	Operating Instructions	
	Introduction	2-1
	Getting Started	2-1
	Reentering the Program	2-1
	Entering Commands	2-1
	Entering Parameters	2-2
	File Types	2-3
	End of File Detection	2-3
	Handling Error Messages	2-4
SECTION 3	Editing Commands	
	Introduction	3-1
	DELETE	3-1
	INSERT	3-2
	RUBOUT CHARACTER	3-2
	NEIGHBORHOOD	3-2
	ALTER	3-3
	COPY	3-3
	MOVE	3-3
	FIND	3-4
	SEARCH	3-4
	UP	3-4
	DOWN	3-4
	SUBSTITUTE	3-5
	SPLIT	3-5
	MERGE	3-6
	UPPERCASE	3-6
	LOWERCASE	3-6
	CHANGECASE	3-7
	MARGIN	3-7
	ADD	3-7
	TRIM	3-8
	CENTER	3-8
	INDENT	3-9
	REPEAT	3-9

SECTION 4	Environmental Commands	
	Introduction	4-1
	SET	4-2
	WIDTH	4-2
	NUMBERS	4-2
	NONNUMBERS	4-3
	LASTLINE	4-3
	RETURN	4-3
SECTION 5	I/O Commands	
	Introduction	5-1
	OLD	5-2
	APPEND	5-2
	LIST	5-3
	SAVE	5-3
	OPEN	5-4
	CREATE	5-4
	CLOSE	5-4
	INPUT	5-5
	SKIP	5-5
SECTION 6	Sample Editing Session	6-1
SECTION 7	BASIC Program Description	
	Text Buffer Format	7-1
	Program Organization	7-2
	Variable Allocation	7-4

GENERAL DESCRIPTION

INTRODUCTION

The EDT-100 Editing Package is a versatile, high performance, ROM Pack driven software package for editing text and programs. The ROM Pack enhances speed and conserves memory space, and BASIC gives the program a friendly user interface.

Because the text buffer is a string variable, the user can return to BASIC without losing the text buffer. The user then has complete access to the data for custom applications. The ROM Pack performs most of the data manipulation in the text buffer, enhancing speed and conserving memory space. The user interface is in BASIC, so the package can read and write files on tape, disk, or the auxiliary memory.

Useful features include automatic line numbering, frequently used commands accessible through the User Definable Keys, and a versatile user interface. Editing commands include Delete, Alter, Copy, Move, Search, Substitute, Uppercase, Lowercase, ChangeCase, Margin, Add, Trim, Center, and Indent. I/O commands include Old, Append, List, Save, Open, Create, Close, Input, and Skip.

INSTALLATION INSTRUCTIONS

The power to the 4050 should be turned off before the ROM Pack is installed. After the power is shut off, the ROM Pack may be inserted into a slot in the firmware backpack or into a slot of a ROM Expander Unit. Press down gently until the edge card connector is seated in the receptacle connector.

OPERATING INSTRUCTIONS

INTRODUCTION

The BASIC program makes this a versatile, easy to use editing package. It accepts a variety of commands, some of which are accessible through the user definable keys. The commands include the necessary editing functions and some powerful formatting commands as well.

GETTING STARTED

To use the editing package, the ROM Pack must be installed according to the instructions given in the preceding section. Then the BASIC program must be OLDed in. The distribution tape contains a copy of the program on file 1, so the user need only press the **AUTOLOAD** key and let the computer do the rest. If desired, the program can be put on disk and OLDed from disk. When the program is RUN, the variables used by the program are defined and the text buffer is dimensioned. A notice appears on the screen telling the user that the program has finished initializing itself. This notice also displays the size of the text buffer. The program is now ready to accept commands.

REENTERING THE PROGRAM

If BASIC generates an error message and stops program execution or if you leave the program by using the RETURN command or the BREAK key, press **User Definable Key 1** to restart the program with no loss of data.

ENTERING COMMANDS

All commands can be typed on the keyboard. Some commands can also be executed by pressing a User Definable Key. When the program is prompting for a command (blinking question mark) the User Definable Keys are active and pressing one is generally equivalent to typing the command keyword. If you have typed part of a command line when you press the key, the program will interpret what you typed as parameters to the

command. This feature allows you to type a line number and press **User Definable Key 12** to enter the ALTER command. The repeat command will not work with this method of entering a command, because the command line string variable only contains the part explicitly typed. Most command keywords may be abbreviated to the first one to four characters depending on the command. The allowed abbreviations for each command are contained in the explanation of the commands.

ENTERING PARAMETERS

Parameters are entered on the command line after the command keyword and are delimited by spaces. Most parameters fall into the following classes: Range Specifiers, Line Specifiers, Literal Strings, Width Specifiers, and File Specifiers.

Range specifiers may be one number or two numbers separated by a comma. If one number is entered, the range consists of one line with the number given. If two numbers are entered, the range consists of the lines between those with the numbers given, inclusive. If this parameter is left blank, most commands default to the current line. The List and Save commands default to the entire text buffer if the range parameter is omitted. Example: **LIST 2,10** lists lines 2 through 10 to the screen.

Line specifiers are the number of the desired line. Most commands allow this parameter to be omitted, in which case the current line is used. Commands which add to the text buffer do so before the line specified. Example: **INSERT 5** starts inserting before line 5.

A literal string is a sequence of characters surrounded by quotation marks. As in BASIC, to enter a string containing quotation marks, the quotation marks which are to be part of the string must be doubled. This is necessary only on command lines and is not needed when directly entering text into the text buffer. The following command line searches for a quotation mark with commas on either side: **SEARCH ",","**.

Width specifiers are the number of characters in the desired line length, etc. If this parameter is omitted, the default width as set by the WIDTH command is used. Example: **WIDTH 60** sets the default width to 60 characters.

A file specifier identifies an I/O device, a disk file, or an Auxiliary Memory file. A number preceded by an at sign (@) selects a GPIB device, a tape file, or a ROM Pack interface, a number preceded by an exclamation mark selects an Auxiliary Memory file, and a literal string selects a disk file by name. If this parameter is left blank, a previously opened file is used if one is open or a default device is selected on commands that have a default device number.

A number preceded by an at sign refers to a device number on the GPIB or in a ROM Pack slot. The colon after the device number is optional, but if a number is entered after the colon, a FIND command is issued to this file number. For example, **SAVE @33:1** locates and opens file number one on the internal magnetic tape drive and saves the text buffer on this file. A number preceded by an exclamation mark refers to a

file in the TransEra Auxiliary Memory. For example, **SAVE !34** saves the text buffer in Auxiliary Memory file number 34. A literal string is the name of a disk file.

FILE TYPES

The I/O commands in the editing package use sequential files in the Auxiliary memory, sequential ASCII files on disk, and ASCII files on tape. Data transfers to and from GPIB devices and ROM Pack devices are done with PRINT and INPUT statements.

END OF FILE DETECTION

The program can access several devices other than the internal magnetic tape. The tape, disk, and TransEra Auxiliary Memory have built in end of file detection, but other devices such as ROM Pack interfaces and GPIB devices do not have automatic detection. For this reason, the APPEND, INPUT, and SKIP commands honor the line "...END OF FILE..." as the last line of the text file on those devices. This line is not inserted by the LIST and SAVE commands and must be manually entered into the text buffer by the user if this feature is to be used.

HANDLING ERROR MESSAGES

The program does as much checking for conditions that would cause an error message as practical. However, many such conditions are impractical or impossible to check. If any of these conditions occurs, BASIC will issue an error message and abort the program. These conditions include an OLD or SAVE to a nonexistent device and creating a line in the text buffer longer than 72 characters.

When this happens, restarting the program with User Definable Key 1 will allow the user to continue editing in many cases. In some cases, this is not possible. The most likely cause of this is that the text buffer contains lines too long for any commands. In this case, it may be necessary to split or delete the line and reenter it.

EDITING COMMANDS

INTRODUCTION

The editing commands include Delete, Insert, Rubout Character, Neighborhood, Alter, Copy, Move, Find, Search, Up, Down, Substitute, Split, Merge, Uppercase, Lowercase, ChangeCase, Margin, Add, Trim, Center, Indent, and Repeat. These commands allow the user to enter text into the buffer, manipulate the text already in the buffer, and format the text in the buffer for a professional looking final copy. This section of the manual describes each of these commands in detail.

DELETE (DE) range

range - range of lines to be deleted

The DELETE command removes one or several lines from the text buffer. If the range parameter is omitted, the current line is deleted. The keyword DELETE may be abbreviated to "DE". The abbreviation "D" specifies the DOWN command.

EXAMPLE: DE 2,10 (Deletes lines 2 through 10)

INSERT (I) line

line - line to insert text before

The INSERT command allows the user to manually add text to the text buffer. The added text is inserted before the specified line and the line numbers of all text after that added are increased by the number of lines inserted. The program displays the number of the new line to be added and waits for input. To leave insert mode, press any of the User Definable Keys. The Key's function will not be performed, but the program will leave insert mode. The INSERT command can be issued by pressing User Definable Key 2. The keyword INSERT may be abbreviated to "I".

EXAMPLE: I 5 (Inserts text before line 5)

RUBOUT (R)

In some unusual editing applications, it is desirable to put the rubout character (ASCII value 127) into the text. The RUBOUT command performs this function by inserting a line containing the rubout character before the current line. The MERGE and SPLIT commands may be used to position the rubout character in the middle of an existing line if needed. The RUBOUT command can be issued by pressing User Definable Key 7. The keyword RUBOUT may be abbreviated to "R".

EXAMPLE: R (Puts a line before the current line)

NEIGHBORHOOD (N)

The NEIGHBORHOOD command erases the screen and prints the lines surrounding the current line. The number of lines to print before and after the current line is set by the SET command. The NEIGHBORHOOD command can be issued by pressing User Definable Key 3. The keyword NEIGHBORHOOD may be abbreviated to "N".

EXAMPLE: N (Pages the screen and prints the neighborhood)

ALTER (A) line

line - line number to alter

The ALTER command allows the user to modify existing lines of text. This is similar to the insert mode, but the alter mode does not add new lines, it only modifies existing ones. To leave the alter mode, press any User Definable Key. The key's function will not be performed, but the program will leave the alter mode. The ALTER command can be issued by pressing User Definable Key 12. The keyword ALTER may be abbreviated to "A".

EXAMPLE: A 43 (Alters text starting with line 43)

COPY (C) range line

range - range of lines to be copied

line - line to receive copy of range

The COPY command creates another copy of the specified range of lines and inserts the new copy before the specified line. The existing range of lines are not affected by this command. The target line may be before or after the range to be copied, but it may not be inside of the range to be copied. The keyword COPY may be abbreviated to "C".

EXAMPLE: C 2,10 20 (Copies lines 2 through 10 before line 20)

MOVE (M) range line

range - range of lines to be moved

line - line to receive the moved range

The MOVE command inserts a copy of the range before the target line and deletes the original range. The target line may be before or after the range to be moved, but it may not be inside of the range to be moved. The keyword MOVE may be abbreviated to "M".

EXAMPLE: M 2,10 20 (Moves lines 2 through 10 before line 20)

FIND (F) line

line - line number to locate

The FIND command locates the specified line and makes it the new current line. The keyword FIND may be abbreviated to "F".

EXAMPLE: F 43 (Makes line 43 the current line)

SEARCH (S) "literal"

"literal" - string to search for

The SEARCH command attempts to locate a specified sequence of characters in the text buffer starting with the current position in the buffer and looking toward the end of the buffer. The new current line is set to the beginning of the line in which the string is found. The keyword SEARCH may be abbreviated to "S".

EXAMPLE: S "BASIC" (Makes the next line containing the string BASIC the current line)

UP (U)

The UP command moves the current line one closer to the beginning of the text buffer. It also displays the new current line. The UP command can be issued by pressing User Definable Key 16. The keyword may be abbreviated to "U".

EXAMPLE: U (Makes the next line the current line)

DOWN (D)

The DOWN command moves the current line one closer to the end of the text buffer. It also displays the new current line. The DOWN command can be issued by pressing User Definable Key 6. The keyword DOWN may be abbreviated to "D".

EXAMPLE: D (Makes the previous line the current line)

SUBSTITUTE (SU) "literal" "literal" range

- "literal"** - string to locate and remove
- "literal"** - string to replace first string with
- range** - range of lines to perform the operation on

The **SUBSTITUTE** command searches the specified range for the first literal string and replaces each occurrence with the second literal string. The default range is the current line. The keyword **SUBSTITUTE** may be abbreviated to **"SU"**. The abbreviation **"S"** specifies the **SEARCH** command.

EXAMPLE: SU "Editer" "Editor" 0,9999 (Replaces each occurrence of the string "Editer" to the string "Editor")

SPLIT (SP) line "literal"

- line** - line to split
- "literal"** - string to make the first part of the new line

The **SPLIT** command makes two lines out of one original line. The command searches for the first occurrence of the literal string in the specified line and inserts a carriage return before the literal string, making a new line which starts with the search string. Lines after the split line are automatically increased in number. The keyword **SPLIT** may be abbreviated to **"SP"**. The abbreviation **"S"** specifies the **SEARCH** command.

EXAMPLE: SP 33 "BASIC" (Splits line 33 making a new line 34 which starts with the string "BASIC")

MERGE (ME) line

line - line to merge

The MERGE command merges two lines into one line. The two lines to be merged are the specified line and the line immediately after it. Thus MERGE 3 combines lines three and four into one new line three. Lines after the merged lines are automatically decreased in number. The keyword MERGE may be abbreviated to "ME". The abbreviation "M" specifies the MOVE command.

EXAMPLE: ME 33 (Merges lines 33 and 34 into a new line 33)

UPPERCASE (UPP) range

range - range of lines to be uppercased

The UPPERCASE command changes all lowercase alphabetic characters to their uppercase equivalents. Control characters, numeric characters, and punctuation symbols are not affected by this command. Uppercase letters remain uppercase. The keyword UPPERCASE may be abbreviated to "UPP". The abbreviations "U" and "UP" specify the UP command.

EXAMPLE: UPP 2,10 (Makes lines 2 through 10 all uppercase)

LOWERCASE (L) range

range - range of lines to be lowercased

The LOWERCASE command changes all uppercase alphabetic characters to their lowercase equivalents. Control characters, numeric characters, and punctuation symbols are not affected by this command. Lowercase letters remain lowercase. The keyword LOWERCASE may be abbreviated to "L".

EXAMPLE: L 2,10 (Makes lines 2 through 10 all lowercase)

CHANGECASE (CH) range

range - range of lines to be changed

The CHANGECASE command changes all uppercase letters to their lowercase equivalents and all lowercase letters to their uppercase equivalents. Control characters, numeric characters, and punctuation symbols are not affected by this command. The keyword CHANGECASE may be abbreviated to "CH". The abbreviation "C" specifies the COPY command.

EXAMPLE: CH 2,10 (Changes the case of each character in lines 2 through 10)

MARGIN (MA) range width

range - range of lines to be margined

width - width of page

The MARGIN command adjusts the text within the specified range of lines to even up the right margin. This command honors blank lines as paragraph boundaries and puts as many words in each line within a paragraph as possible without exceeding the margin width. Short lines are made longer and long lines are made shorter. The keyword MARGIN may be abbreviated to "MA". The abbreviation "M" specifies the MOVE command.

EXAMPLE: MA 2,10 30 (Margins lines 2 through 10 to 30 characters)

ADD (AD) range "literal"

range - range of lines to add literal to

"literal" - string to add to each line

The ADD command concatenates the literal string to the end of each line in the specified range. Care should be taken to avoid making lines more than 72 characters long. An error message 21 will be issued if this is attempted. The keyword ADD may be abbreviated to "AD". The abbreviation "A" specifies the ALTER command.

EXAMPLE: AD 2,10 "....." (Adds the string "....." to lines 2 through 10)

TRIM (T) range width

- range** - range of lines to be trimmed
- width** - number of characters to trim each line to

When making tables of contents, it is often desirable to add a certain character to each of several lines and then make each line the same length. The TRIM command performs this function by deleting characters from the end of each line of the specified range until the length of the line is at most the specified width. If the width specifier is omitted, the default width set by the WIDTH command is used. If no lines in the range are longer than the width, the text buffer is unchanged. The TRIM command can be issued by pressing User Definable Key 4. The keyword TRIM may be abbreviated to "T".

EXAMPLE: T 3,5 55 (Trims lines 3 through 5 to 55 characters)

CENTER (CE) range width

- range** - range of lines to be centered
- width** - width to center to

The CENTER command is useful for centering titles, tables, etc. Spaces are added to the beginning of each line of the specified range until the line is centered with respect to the specified width. If the width specifier is omitted, the default width set by the WIDTH command is used. The CENTER command trims spaces from the ends of each line before centering, so the lines will be properly centered even if they have leading or trailing spaces. The keyword CENTER may be abbreviated to "CE". The abbreviation "C" specifies the COPY command.

EXAMPLE: CE 2 60 (Centers line 2 to conform to a right margin of 60 characters)

INDENT (IND) range width

- range** - range of lines to be indented
width - number of leading spaces to add to each line

The INDENT command trims leading spaces from each line in the range and adds leading spaces equaling the specified width in number. This leaves each line in the range with exactly the specified number of leading spaces. Both parameters must be included in the command line. If the width is zero, each line will be flush with the left margin. The keyword INDENT may be abbreviated to "IND". The abbreviations "I" and "IN" specify the INSERT command.

EXAMPLE: IND 45,50 3 (Indents lines 45 through 50 in 3 spaces)

REPEAT (REP)

The REPEAT command repeats the last command issued. It can be accessed through User Definable Key 11 and is useful if the same command is to be issued several times. The keyword REPEAT may be abbreviated to "REP". The abbreviation "R" specifies the RUBOUT command, and the abbreviation "RE" specifies the RETURN command.

EXAMPLE: REP (Repeats the last command)

ENVIRONMENTAL COMMANDS

INTRODUCTION

The environmental commands include Set, Width, Numbers, Nonumbers, Lastline, and Return. These commands set and display environmental information which adapts the program to a specific editing application and allows the user to customize certain aspects of the program to fit his needs. This section of the manual describes each of these commands in detail.

SET (SET) back forward

back - number of lines before the current line
forward - number of lines after the current line

The SET command determines the number of lines before and after the current line that will be displayed by the NEIGHBORHOOD command. If both parameters are zero, only the current line will be displayed. The parameters default to 5 and 10. The keyword SET may not be abbreviated. The abbreviations "S" and "SE" specify the SEARCH command.

EXAMPLE: SET 3 5 (Sets the neighborhood command up to display the 3 lines before and the 5 lines after the current line)

WIDTH (W) width

width - new default width

The WIDTH command sets the default width used by the MARGIN, TRIM, CENTER, and INDENT commands if the width specifier in these commands is omitted. This value is limited to values from 0 to 72. The keyword WIDTH may be abbreviated to "W".

EXAMPLE: W 65 (Sets the default width to 65 characters)

NUMBERS (NU) "literal"

"literal" - string to print after line number

The NUMBERS command enables the printing of line numbers for most commands which display or output lines from the text buffer. These include NEIGHBORHOOD, ALTER, INSERT, SAVE, and LIST. The literal string is printed after the line number and before the text of the line. The NUMBERS command can be issued by pressing User Definable Key 8. The keyword NUMBERS may be abbreviated to "NU". The abbreviation "N" specifies the NEIGHBORHOOD command.

EXAMPLE: NU ":" (Selects printing of line numbers followed by a colon)

NONUMBERS (NO)

The NONUMBERS command is the opposite of the NUMBERS command in that it disables the printing of line numbers. The NONUMBERS command can be issued by pressing User Definable Key 9. The keyword NONUMBERS may be abbreviated to "NO". The abbreviation "N" specifies the NEIGHBORHOOD command.

EXAMPLE: NO (Disables printing of line numbers)

LASTLINE (LA)

The LASTLINE command displays the current line number, the number of lines in the text buffer, the number of characters in the text buffer including carriage returns, and the number of free characters in the text buffer. The LASTLINE command can be issued by pressing User Definable Key 10. The keyword LASTLINE may be abbreviated to "LA". The abbreviation "L" specifies the LOWERCASE command.

EXAMPLE: LA (Displays the text buffer length and the current line number)

RETURN (RE)

The RETURN command terminates execution of the BASIC editing program and returns the user to BASIC. The text buffer is not affected by this command and no files are closed. The RETURN command can be issued by pressing User Definable Key 5. The keyword RETURN may be abbreviated to "RE". The abbreviation "R" specifies the RUBOUT command.

EXAMPLE: RE (Returns to BASIC)

I/O COMMANDS

INTRODUCTION

The I/O commands include OLD, APPEND, LIST, SAVE, OPEN, CREATE, CLOSE, INPUT, and SKIP. These commands allow exchange of data between the text buffer and peripheral devices. Many of these commands have the same keyword as a BASIC command, but they are different in several important aspects. Except for CLOSE, all of these commands leave the file open. This allows multiple SAVES and LISTs to the same file or device without having to reopen it each time. This section of the manual describes these commands in detail.

OLD (O) file

file - file specifier to bring in to the text buffer

The OLD command is similar to the BASIC OLD statement, but there are differences. The current contents of the text buffer are deleted and the file is brought in. The OLD command uses a ROM routine to perform the I/O into the text buffer, so it is much faster than APPEND. If the file specifier is omitted, a previously opened file is used. If no file is open, an error message is issued. The keyword OLD may be abbreviated to "O".

EXAMPLE: O @33:1 (FINDs and OLDs tape file 1)

APPEND (AP) file line

file - file to be brought in

line - line number to insert file before

The APPEND command inserts the contents of a file before the specified line in the text buffer. It is slower than the OLD command. If the file specifier is omitted, a previously opened file is used. If no file is open, an error message is issued. The keyword APPEND may be abbreviated to "AP". The abbreviation "A" specifies the ALTER command.

EXAMPLE: AP "DATA" 45 (Appends disk file "DATA" before line 45)

LIST (LI) file range

file - file to list to
range - range of lines to list

The LIST command outputs the specified range of lines to the specified file. If the range specifier is omitted, the entire file is LISTed. If the file specifier is omitted, the listing appears on the screen. The LIST command outputs control characters as the corresponding uppercase character, a backspace character, and an underline character. The keyword LIST may be abbreviated to "LI". The abbreviation "L" specifies the LOWERCASE command.

EXAMPLE: LI @41 (Lists the entire text buffer to the printer interface in ROM Pack slot 41)

SAVE (SA) file range

file - file to save to
range - range of lines to save

The SAVE command outputs the specified range of lines to the specified file. If the range specifier is omitted, the entire file is SAVEed. If the file specifier is omitted, the output is to the internal magnetic tape drive. The SAVE command outputs control characters exactly as they occur in the text buffer. The keyword SAVE may be abbreviated to "SA". The abbreviation "S" specifies the SEARCH command.

EXAMPLE: SA @16 30,500 (Saves lines 30 through 500 on GPIB device 16)

OPEN (OP) file

file - file to open

The OPEN command opens a file for reading so the OLD and APPEND commands need not specify a filename or device number. Files to be output to, must be opened with the LIST or SAVE commands. The OPEN command is present in the package primarily as a counterpart to the BASIC FIND statement. OLD and APPEND can open a file without doing an OPEN command by specifying a file device or filename in the command line. OPEN will find and open a tape file if one is already marked, but it will not MARK a tape file. To MARK tape files, return to BASIC and use the BASIC commands to MARK the files. The keyword OPEN may be abbreviated to "OP". The abbreviation "O" specifies the OLD command.

EXAMPLE: OP @33:10 (Finds and opens tape file 10 for input)

CREATE (CR) "name"

"name" - filename to create on disk

The CREATE command is for creating disk files to save or list to. The keyword CREATE may be abbreviated to "CR". The abbreviation "C" specifies the COPY command.

EXAMPLE: CR "NEWDATA" (Creates disk file "NEWDATA")

CLOSE (CL)

The CLOSE command closes open files. It should be executed before removing the tape or floppy disks or powering the system down. Since the SAVE and LIST commands can access the same file more than once, it is suggested that the CLOSE command be issued after saving or listing a file so the file is not added to inadvertently. The keyword CLOSE may be abbreviated to "CL". The abbreviation "C" specifies the COPY command.

EXAMPLE: CL (Closes all I/O files)

INPUT (INP) file

file - file to be displayed

The INPUT command reads one line from the specified file and displays it on the screen. This is useful when deciding which tape or disk file to append or old. It can also be used to skip past unwanted information in a file so a small part of the file may be OLDed or APPENDED. The INPUT command can be issued by pressing User Definable Key 20. The keyword INPUT may be abbreviated to "INP". The abbreviations "I" and "IN" specify the INSERT command.

EXAMPLE: INP @33 (Inputs and displays one line from the previously opened tape file)

SKIP (SK) file number

file - file to skip input from
number - number of lines to skip

The SKIP command reads the specified number of lines from the specified file. This is useful when OLDing or APPENDING a small portion of a file. This command does not affect the text buffer, and nothing is displayed. The keyword SKIP may be abbreviated to "SK". The abbreviation "S" specifies the SEARCH command.

EXAMPLE: SK "DATA" 30 (Opens disk file "DATA" if not already open and skips past 30 lines)

SAMPLE EDITING SESSION

The following is a sequence of commands executed in a typical editing session.

O @33:1	FINDs and OLDs tape file 1
AP "DAT" 30	APPENDs disk file "DAT" before line 30
W 60	Sets the default width to 60
CE 1,5	Centers lines 1 through 5 to 60 characters
MA 10,15 50	Margins lines 10 through 15 to 50 characters
IND 10,16 5	Indents lines 10 through 16 by 5 spaces
SA @33:2	FINDs and SAVES on tape file 2
CL	Closes I/O files
SA "NEWDAT"	SAVES on disk file "NEWDAT"
CL	Closes I/O files

BASIC PROGRAM DESCRIPTION

TEXT BUFFER FORMAT

The internal format of the text buffer is as follows. The entire text buffer is contained in one large string variable. Separate lines within the file are delimited by carriage returns. This allows the file string to be printed to the 4050 graphics screen, magnetic tape drive, disk, or other I/O device.

There is no overhead within the file string for line numbers or line lengths. The number of a line is found by scanning through the string and counting the carriage returns. The length of a line is found by scanning for the next carriage return after the start of a line. Thus, adding text automatically increases the line numbers of all lines following the added material.

PROGRAM ORGANIZATION

The program is organized into functional blocks. Most of these blocks begin with a REM statement for readability.

The following is a list of major program sections and the line numbers they start at.

USER DEFINABLE KEYS,	4
SYSTEM INITIALIZATION,	100
WARM START,	800
COMMAND FETCH AND DISPATCH,	1000
SYNTAX ERROR,	1700
CLEAN UP AFTER COMMAND,	1800
UNIMPLEMENTED EDITOR COMMAND,	1900

The following is a list of commands and the starting line number in the program where they may be found.

COMMAND	LINE NUMBER
DELETE	2000
INSERT	3000
RUBOUT	4000
NEIGHBORHOOD	5000
ALTER	6000
COPY	7000
MOVE	8000
FIND	9000
SEARCH	10000
UP	11000
DOWN	12000
SUBSTITUTE	13000
UPPERCASE	14000
LOWERCASE	15000
CHANGECASE	16000
MARGIN	17000
ADD	18000
TRIM	19000
CENTER	21000

SET	24000
NUMBERS	25000
NONNUMBERS	26000
LASTLINE	27000
RETURN	28000
OLD	30000
APPEND	31000
LIST	32000
SAVE	33000
OPEN	34000
CLOSE	35000
INPUT	36000
SKIP	37000
REPEAT	38000
CREATE	39000
SPLIT	40000
MERGE	41000
WIDTH	42000
INDENT	43000

The following is a list of subroutines used by the program to parse the command line. The target variables are enclosed in parentheses.

PARSE A LITERAL (L)\$	50000
PARSE A LINE (L1, L2)	51000
PARSE A RANGE (R1 - R4)	52000
PARSE A FILE AND OPEN IT (F1)	54000
TRIM LEADING BLANKS OFF W\$	55000
PARSE A WIDTH SPECIFIER (W2)	56000

VARIABLE ALLOCATION

The following is a list of the variables used by the program. Unless otherwise specified, numeric variables are scalars, and the string variables are dimensioned to 72 characters.

A\$ - FILE STRING

C\$ - CURRENT COMMAND STRING

C0 - CODE FOR CURRENT COMMAND, EQUAL TO STARTING LINE / 1000

C1 - CURRENT CURSOR LINE NUMBER

C2 - CURRENT CURSOR POSITION

E1 - PENDING ERROR CODE, SUCCESS CODE, 0=OKAY

F\$ - DISK FILENAME IF F1=1

F1 - CODE FOR I/O - 0=NO OPEN FILE
 1=DISK FILE OPEN
 +OTHER=DEVICE FILE OPEN
 -OTHER=MEM FILE OPEN

K1 - CODE FOR PENDING UDK (NUMBER OF KEY OR 0)

L\$ - LITERAL STRING TO SEARCH FOR

L1 - LINE NUMBER OF SELECTED LINE

L2 - POSITION OF SELECTED LINE

M\$ - ONE CHARACTER STRING CONTAINING A CARRIAGE RETURN

N\$ - STRING TO INSERT FOR NUMBERING ("" MEANS NO NUMBERS) (10)

N1 - POSITIVE OFFSET TO START OF NEIGHBORHOOD

N2 - POSITIVE OFFSET TO END OF NEIGHBORHOOD

O\$ - OLD COMMAND STRING

R1 - STARTING LINE NUMBER OF RANGE

R2 - STARTING POSITION OF RANGE

R3 - ENDING LINE NUMBER OF RANGE

R4 - ENDING POSITION OF RANGE

S\$ - LITERAL STRING TO SUBSTITUTE IN

W\$ - WORKING COPY OF CURRENT COMMAND STRING

W1 - DEFAULT WIDTH

W2 - CURRENT WIDTH

X\$ - 80-CHARACTER SCRATCH STRING

X - SCRATCH

X0 - SCRATCH

X1 - SCRATCH

X2 - SCRATCH

X3 - SCRATCH

X4 - SCRATCH

X5 - SCRATCH

X6 - SCRATCH

X7 - SCRATCH

X8 - SCRATCH

X9 - SCRATCH

Y\$ - 72-CHARACTER SCRATCH STRING

Z\$ - 150-CHARACTER SCRATCH STRING