

altair^{T.M.}
Word
Processing
Package

A Sophisticated Word Processing
System for the Altair 8800 Computer



3330 Peachtree Road, Suite 343
Atlanta, Georgia 30326

See Your Local
Altair Software Dealer

Altair Word Processing Package

A Sophisticated Word Processing Package
for the Altair 8800 Computer

Description of the Altair Word Processing Package

Background	1
Altair Word Processing Package General Description	2
The Text Editor	2
The Text Formatter	5
Applications	8
Conclusions	8
Examples of Word Processing Documents	9
Summary of Text Editor Commands	11
Summary of Formatter Commands	12

The Altair Word Processing Package is a set of proprietary software available to the end user under the terms of a limited use license agreement for a one-time license fee. The software and its documentation are the property of the Altair Software Distribution Company. Unauthorized reproduction or use of the software and its documentation is a violation of state and federal law.

Copyright 1977 Altair Software Distribution Company



ALTAIR SOFTWARE DISTRIBUTION COMPANY
3330 Peachtree Road, Suite 343
Atlanta, Georgia 30326 404-231-2308

Altair* Word Processing Package

A Sophisticated Word Processing Package for the Altair 8800 Computer.

BACKGROUND

Word processing systems come in all shapes, sizes, capabilities, and prices. They range in design from the automatic typewriter with self-contained memory to text editing and formatting systems implemented on large scale computer systems. Whatever the configuration, word processing systems have generally the basic function of the storage and repetitive typing of business documents. From this foundation, word processing systems begin to differ in how they implement this basic function and in the additional capabilities they present. Even the most simple systems provide for some editing of text, usually as it is being entered and at some later time. A feature greatly sought by word processing users is the ability to make more extensive editing changes in a document by moving around portions of text, inserting new text, deleting old text, having access to text which is stored in other documents, and in being able to search a body of text for a particular character set.

The "parent" of modern word processing machines -- the stoic MTST (Magnetic Tape Selectric Typewriter), first marketed by IBM in the early 1960's, allowed for only a very limited use of these desirable features. Since the MTST machines stored their text on magnetic tape cartridges, the text could only be handled sequentially. The advent of the computer disk provided a random access storage device which was ideal for large volume text editing. The introduction of the flexible diskette (the "floppy disk") provided the means for less expensive, more cost-effective word processing systems. Several word processing systems were implemented using random disk or diskette storage on minicomputers beginning about 1974. These systems typically cost the end user \$25,000 and up. A few of these systems were referred to as "shared logic" since the minicomputer employed in the word processing function could be used for other purposes.

Today, word processing systems stand on the threshold of a major decrease in cost and an increase in capability. The use of the Altair computer as the heart of a shared logic word processing systems means that old cost barriers have been broken while performance substantially increased. When MITS introduced the Altair computer in December 1974, the microcomputer became accessible to the general public. Commercial, professional, and small business users were quick to recognize the potential presented by a powerful, but low cost, general purpose computer. The needs of Altair computer users have been met by an increasing number of software packages for various business and scientific

*ALTAIR is a trademark of MITS, Inc.

applications. The Altair Word Processing Package represents a significant step in bringing highly sophisticated applications software to the Altair Computer user.

ALTAIR WORD PROCESSING PACKAGE--GENERAL DESCRIPTION

The Altair Word Processing Package is a flexible and powerful text editing and formatting system which operates on an Altair 8800 computer equipped with 48K of memory, a printer, a video terminal, and at least one floppy disk drive. The package utilizes Altair Extended Disk BASIC, and several machine language routines, and can therefore be used for a wide variety of other purposes, including the operation of the Altair Business System accounting system packages. The Altair Word Processing Package can operate in a variety of hardware configurations to allow maximum flexibility for the user. The typical configuration would include the Altair Q-70 Printer which produces typewriter quality output. Although the package will operate with a single floppy disk drive, most users find it more effective to use a dual drive installation. Virtually any video terminal which produces standard ASCII characters and is compatible with the Altair computer can be utilized.

The Altair Word Processing Package is an extremely flexible and powerful system. At first glance, the list of commands for the editor or the formatter may appear somewhat imposing; but, the package is self-instructing and even an inexperienced user can make effective use of the package within a few hours. The remainder of this brochure will illustrate a number of features of the Altair Word Processing Package; however, space does not permit a complete explanation of all of the powerful commands. Altair Software dealers throughout the United States can provide more complete information and a demonstration of the package.

THE TEXT EDITOR

The Word Processing Package can be described as two modules -- the text EDITOR and the text FORMATTER. The editor allows the user to create, modify, and store the textual document. A set of EDITOR COMMANDS allows the user to control the operation of the system as text is being inputted or modified at a later time. Likewise, a set of FORMATTER COMMANDS directs the way in which the formatter prints the document. Unlike many word processing systems which have invisible "control codes", the formatter commands are visible to the user as a set of special characters and statements. The user can see the embedded commands in the raw document -- not merely their result. This manner of operation gives the user much more control over the document to be printed.

The editor handles text as lines of up to 120 characters. Each line of text has a line number in the range .001 to 999.999, although the maximum number of lines permitted in a single document is 1000. The size of a single document can, therefore, be up to 120,000 characters (about 40-45 single-spaced, letter-sized pages). Two "full" documents of 1000 lines each can be stored on

a single floppy diskette or many shorter documents. Several documents may be handled together to form a "book" of virtually unlimited length. The storage of documents on the inexpensive diskettes allows the storage of a large number of documents in machine readable forms for very low cost. The Word Processing Package uses fractional line numbers, such as 1.005 to allow a large amount of text to be inserted at a particular point without altering subsequent line numbers.

Editing may be performed on a line of text as it is being inputted or at some later time with IN-LINE EDITING done by standard cursor control keys, if the video terminal is so equipped, and with "control" keys. The user types-in and edits lines of text where the video terminal is "positioned" just as a standard typewriter types text on the line so positioned. Additional in-line editing may be performed using the ALTER command discussed below.

The Altair Word Processing package provides a variety of editing commands which facilitate the creation and maintenance of text. The editor operates in two modes: (1) the text mode - inputting of lines of text, and (2) the edit mode - the inputting of an editor command. The editor flip-flops between the two modes when a null line -- a line with nothing on it -- is typed. When the editor is expecting a line of text the next sequential line number is presented by the system. When the editor is expecting editor command a "?" is presented by the system. Most of the editor commands may be abbreviated to the first character for convenience.

The user initiates the editing of a document by identifying the document name (and the disk drive number if more than one disk drive is employed) with an EDIT file name command. If a document by that name already exists on the diskette mounted in the specified drive, that existing document file is "opened" for editing. If not, the system opens a new document file under that name. The system assumes that line numbering will begin with 1 in a new document, or 1 greater than the last line in an existing document, and that each line number will be incremented by 1. The user may reset both the beginning line number and the increment at any time with the GOTO (line number) I (increment number) command. For example, G 10 I .5 would cause the line numbering to be 10.0, 10.5, 11.0, 11.5, and etc.

If a command is typed incorrectly, the system will respond with a brief prompt of the correct form of the command. At any time, a user can type the HELP command (abbreviated "H") to receive more detailed information about the error or to learn about other features of the system. In fact, the help messages are complete enough to serve as a self-instruct secession for learning to use the editor.

When a document is open for editing a number of commands are available to find and manipulate the text. The FIND command allows all or part of the document to be searched for a particular

a single floppy diskette or many shorter documents. Several documents may be handled together to form a "book" of virtually unlimited length. The storage of documents on the inexpensive diskettes allows the storage of a large number of documents in machine readable forms for very low cost. The Word Processing Package uses fractional line numbers, such as 1.005 to allow a large amount of text to be inserted at a particular point without altering subsequent line numbers.

Editing may be performed on a line of text as it is being inputted or at some later time with IN-LINE EDITING done by standard cursor control keys, if the video terminal is so equipped, and with "control" keys. The user types-in and edits lines of text where the video terminal is "positioned" just as a standard typewriter types text on the line so positioned. Additional in-line editing may be performed using the ALTER command discussed below.

The Altair Word Processing package provides a variety of editing commands which facilitate the creation and maintenance of text. The editor operates in two modes: (1) the text mode - inputting of lines of text, and (2) the edit mode - the inputting of an editor command. The editor flip-flops between the two modes when a null line -- a line with nothing on it -- is typed. When the editor is expecting a line of text the next sequential line number is presented by the system. When the editor is expecting editor command a "?" is presented by the system. Most of the editor commands may be abbreviated to the first character for convenience.

The user initiates the editing of a document by identifying the document name (and the disk drive number if more than one disk drive is employed) with an EDIT file name command. If a document by that name already exists on the diskette mounted in the specified drive, that existing document file is "opened" for editing. If not, the system opens a new document file under that name. The system assumes that line numbering will begin with 1 in a new document, or 1 greater than the last line in an existing document, and that each line number will be incremented by 1. The user may reset both the beginning line number and the increment at any time with the GOTO (line number) I (increment number) command. For example, G 10 I .5 would cause the line numbering to be 10.0, 10.5, 11.0, 11.5, and etc.

If a command is typed incorrectly, the system will respond with a brief prompt of the correct form of the command. At any time, a user can type the HELP command (abbreviated "H") to receive more detailed information about the error or to learn about other features of the system. In fact, the help messages are complete enough to serve as a self-instruct secession for learning to use the editor.

When a document is open for editing a number of commands are available to find and manipulate the text. The FIND command allows all or part of the document to be searched for a particular

set of characters or words (called a string). The system can be instructed to find and print all occurrences of the string or only the first occurrence. The string can be defined, by using single or double quotes, to find exactly that character sequence either isolated from other characters by delimiters, such as spaces (the double quote) or wherever found in text, regardless of what precedes or follows (the single quote). For example, FIND ALL "jack" 1-100 would find all occurrences of the word jack in lines 1 through 100 only if jack were preceded and followed by a delimiter such as a space or a period. Therefore, this command would not find the word jack in jacket or in blackjack. However, a single quote can be used to find occurrences regardless of what preceded or followed. For example, FIND "jack' would find jacket but not blackjack while FIND 'jack' would find jack in both jacket and blackjack. Likewise, FIND 'jack" would find blackjack but not jacket.

Two other commands which are frequently used are the COPY command and the MOVE command. Both commands copy a specified set of lines to another location with a specified line number increment; however, the MOVE command deletes the old lines. The LIST command displays (on the video terminal or printer) all or part of the document exactly as the document is stored in the system -- complete with line numbers and embedded format commands. The LIST command should be compared to the PRINT command which caused the document to be passed through the formatter where the embedded formatter commands are acted upon; LIST is used to review and edit a document.

The power of the Word Processing Package is greatly expanded through the use of STAR FILES. A star file is simply a document which is called upon as a reference or source document for the document currently being edited. Text in the star file can be listed (with the LIST* command), searched (with the FIND* command), and copied to the file being edited (with the COPY* command). However, the system will not permit any text in the star file to be modified so that a master document can be kept secure. For example, a MOVE* command would be invalid since MOVE implies a COPY and a DELETE and lines could not be deleted in a star file. Of course, if a user has edit access to the document which is identified as the star file, the particular document could be opened as an edit file and any desired modifications made. The examples which follow this narrative illustrate the use of the star file.

Any number of documents can be opened as star files, one at a time, during the editing of another document, as long as both files are on diskettes which are currently loaded in disk drives. (Hence, the desirability of a two-disk drive system).

The editor has a number of other useful commands to facilitate text editing. For example, lines may be inserted, deleted, or renumbered at any point. The ALTER command allows a line to be specified in which changes are to be made. The existing line is first displayed and the user may then use the cursor control "arrows" on the video terminal to move the cursor along the line. New characters may simply be typed over the old or a text string may be

inserted by typing CONTROL/I (striking the "I" key while depressing the "CONTROL" key). Whatever is typed following CONTROL/I is inserted in the line until an ESCAPE key is struck. Further, single control keys allow the user to delete unwanted characters, words, and remainders of lines. The DELETE command allows the deletion of single lines or large blocks of text.

The command statement REPLACE ALL "stock#" WITH "Harris Supply Inventory Control Number" would make the indicated replacement throughout the document wherever "stock#" occurred. It should be noted that editor commands may be abbreviated to simplify the typing for the operator. Usually, only the first letter or two of a command need be typed. The last example could be input as RA "stock#" W "Harris Supply Inventory Control Number". Another useful command is the SUBSTITUTE command which allows text strings to be represented with ten substitute codes (the % sign followed by a digit 0 to 9). For example, the user entering Substitute "Georgia Institute of Technology" For %0 and S "Massachusetts Institute of Technology" F %1 would need to type only:

- ?EDIT "Namelist"
1. Dr. John Smith
 2. %0
 3. [blank]
 4. Dr. Tom Jones
 5. %1

to obtain:

Dr. John Smith
Georgia Institute of Technology

Dr. Tom Jones
Massachusetts Institute of Technology

A number of useful "housekeeping" functions may be performed using the Word Processing editor. For example, document files on a diskette may be copied to other diskettes, renamed, or deleted.

THE TEXT FORMATTER

While the text editor offers the user a number of very useful editing features, the other side of the Word Processing Package -- the FORMATTER -- provides even more versatility. The formatter is invoked when a PRINT command is issued in the editor. The formatter processes any specified parts of the document and types out the document under the control of the embedded formatter commands. While these commands were passive in the editor and could be viewed and modified, they are acted upon in the formatter, thereby producing the printed document according to the formats specified. But, the formatter commands are really much more than just a way of specifying page margins and other simple formats. The formatter commands are really a powerful, high level language by which a user can "program" the output of a document.

The text resides in a document on the diskette in more or less free-form, generally without regard to margins, line lengths, pages, or the like. When the text is processed by the formatter the margins are set, text is "filled" by flowing from one line to

the next, pages are automatically spaced and numbered, specified headings are typed on each page, and lines are justified to the right margin. Of course, any of these features can be turned on or off at any point in the document. For example, text in the document can be printed "as is" in the raw document without filling or using specified margins. Likewise, a new page can be "forced" to start whenever a user desires.

The formatter commands are simple special text statements which are input and modified in the editor like any other text. Each formatter command begins with a "@" character; several of the formatter commands are also followed by the same character. It is not necessary to begin a new line each time a formatter command is used; they may appear anywhere in the text. Further, commands may be "nested" so that several particular formats will apply to a text string (such as boldface and underlining).

When a user first initializes the Word Processing Package, or at any subsequent time, a set of standard default parameters are defined for page size, margins, and other formatting information. A user then does not need to define these values in a document unless the particular document is different than the standard. For example, a law firm may define their standard document to be on 8.5" x 11" paper, single-spaced, with the horizontal margins 1.5" on the left and 1" on the right, and the vertical margins 1.5" at the top and 1" at the bottom. Documents to be printed with these dimensions would not need any specification within the document -- the default parameters would control. However, if a document were to be printed on 14 inch paper, the user could specify on the first line of the document (here shown after a remark line):

1. @REM Jones v. Smith, Brief on 12b(7) motion
2. @DIM 8.5, 14

If the law firm discovers, at the last minute, that the document should have been printed on 13 inch paper, the user simply uses the editor to change line 2 to read

2. @DIM 8.5, 13

and reprints the document.

The formatter contains all the usual formatting commands such as underlining, centering, indentation, headings at a specified line, pagination at a specified place, right justification, paragraph and page definition, skipping lines, multiple spacing, and tabulations. In addition, a number of special formatting commands are available. The tab right justifies the intervening text, such as a decimal number, to the next tab stop; this feature is useful for accounting or statistical work. Blocks of text can be set in all capitals, in bold, or both. A specified number of lines can be "floated" such that if they will not fit on the existing page, space will be reserved on the next; a feature that is useful for

attaching photos or drawings to the document. The indentations may be on the left, or the right, or both.

The real power of the Word Processing Package, however, comes not from these generally standard formatting commands, but from the file and input controls contained in the formatter. The first group of these formatter commands allows user interaction with the document while it is being printed. The @Display command allows a specified string of text to be displayed on the video terminal but not printed in the document. User defined "prompts" displayed on the terminal can be very helpful. The @Query command acts in a manner similar to @Display except that the printer halts and waits for the user's response. Whatever is typed on the video terminal by the user is treated as text material by the formatter and printed in the document or acted upon if a command. This powerful feature allows a document to query the operator to have inserted at that point text, such as names and addresses, or other commands, such as the name of a file to be opened.

The second major group of special formatter commands involve the use of an additional document file in printing. Just as a user may have star files opened to provide a master set of text for a document being edited, the document being printed may access any number of other documents, one at a time, for text and other commands. By using these features, a user may build "procedure" or "control command" files. Each additional file may contain large or small portions of text to be inserted. An additional document which is going to be referenced in printing may contain @RECORD statements. This allows the additional document being referenced to be treated as a set of individual records. The primary document may then access the next record in the additional file with the @GET command. Control in the primary or additional files may be shifted to another line by the @GOTO command. In this manner, a repetitive letter in the primary file can access a set of records (such as names and addresses) in one or more additional files. The primary file may refer to and use specific fields within a defined record in an additional file. The @IF command allows one text string to be compared with another to switch control if the second string contains, excludes, equals, or doesn't equal the first. By using several @Query commands, a user can build a mailing list which can then be run against a repetitive letter which is testing for the occurrence of keys in the mailing list records to determine whether to print the name.

The third major set of special formatter commands is the set of text variables. These are a set of variables which represent other strings of text. The system recognizes ten variables of the form @V0, @V1, ..., @V9. The text strings which each of these variables represents may be specified within the document by the @SET command or may be specified in the editor PRINT command which caused the document to be printed. The text string represented by a variable may simply be text or it may contain formatter commands, including other variables. When the formatter encounters the @Vx in the document, the substitution is made as if the string were part of the document. The system recognizes two other variables: @Name

is the name of the operator who initialized the system that day and @DATE is the current date. The use of variables provides a power method of having relevant information inserted at the appropriate place in the document without actually altering the raw document. For example, a form letter may contain the variable @DATE. Each time this letter printed, it is unnecessary to alter the current date; the @DATE variable obtains the date from the system. Likewise, a form letter used by many members of a firm may close:

55. Sincerely
56.
57.
58.
59. @V6

When the PRINT command was issued in the editor, variable V6 can be defined as "John Jones, Sales Manager"; the document itself is not altered.

APPLICATIONS

The possible uses of the Altair Word Processing Package are quite broad. Many law firms, customer-oriented retailers, engineering firms, and government offices already make extensive use of word processing systems. These firms will find that the Altair Word Processing Package is more flexible than their existing systems. The ability to expand the use of the system is great since it is based on the state of the art, low cost Altair 8800 computer. Further, the system which operates the Altair Computer System is a general purpose computer which can be used for a variety of other functions.

The Altair Word Processing System opens new vistas for many organizations. For example, the similarity between the embedded formatter commands and phototypesetting commands is evident. The text editor can be used to input and maintain text to be typeset. The "output" of the document would then be passed to a phototypesetting machine, either directly or through paper tape. (The Word Processing System can be interfaced to a particular phototypesetter at additional cost).

Special routines allow the Word Processing Package to accept as input virtually any file of text, including BASIC programs in ASCII format. This powerful feature allows a user to input and edit, and if he chooses, print text which has been generated by another computer or some other source.

CONCLUSION

The Altair Word Processing Package contains a number of features not found in other word processing systems, whether stand alone machines or part of a computer system. The special formatter commands take the package from the arena of a simple

office text editing system into the realm of a sophisticated, high level computer language for controlling text. Special routines allow the Word Processing Package to accept as input virtually any file of text. The features contained in the system are far beyond what is required for a simple office system, yet the power and flexibility are provided to serve a variety of users very well. These features, combined with the fact that the package operates on the low cost, general purpose Altair 8800 computer, make the Altair Word Processing Package a cost effective investment.

EXAMPLES OF WORD PROCESSING DOCUMENTS

April 15, 1977

Mr. Robert T. Jones, Esq.
17 Fairway
Augusta, Georgia 30447

Dear Bob:

Thank you for the opportunity to demonstrate the Altair Word Processing package. This text is stored in a file called EXAM1 while the inserts (the names and addresses) are stored in a document file called NAME4.

This letter is an example of using text from two different files to produce a single document on output.

Please let me know if I can provide any additional information. With every good wish, I am

Sincerely,

Ronald D. Roberts, President

Enclosure

RDR/JDL

Figure 1: This document is an example of a letter produced from two Word Processing documents, shown in figures 2 and 3.

```

1.000 @PEM This is an example of a Word Processing document.
1.500 @open "NAME4"
2.000 @Page
3.000 @skip5
4.000 @t @t @t @t @date
5.000 @skip4
5.500 @fill on
6.000 @get 30
6.100 @f2
6.200 @f3
6.300 @f4
6.400 @f5
7.000 @skip
8.000 Dear @f1:
9.000
10.000 @P Thank you for the opportunity to demonstrate the Altair Word
11.000 Processing package. This text is stored in a file called
12.000 EXAM1 while the inserts (the names and addresses) are stored
13.000 in a document file called NAME4.
14.000 @p This letter is an example of using text from two different
15.000 files to produce a single document on output.
16.000 @p Please let me know if I can provide any additional information
17.000 . With every good wish, I am
18.000 @skip2
18.500 @fill off
19.000 @t @t @t @t Sincerely,
20.000 @skip4
24.000 @t @t @t @t Ronald D. Roberts, President
25.000
25.100 Enclosure
25.200
26.000 RDR/@NAME
27.000 @goto2
30.000 @end

```

Figure 2: This document is an example of a Word Processing document which uses a background file called "NAME4", shown in Figure 3, to obtain the printed document, the letter shown in Figure 1.

```

1.000 @rem This is the name file called NAME4
2.000 @rec
3.000 Mr. Smith
4.000 Mr. John E. Smith
5.000 3632 Lincoln Lane
6.000 Washington, D. C. 20003
7.000 @rec
8.000 Bob
9.000 Mr. Robert T. Jones, Esq.
10.000 17 Fairway
11.000 Augusta, Georgia 30447
12.000 @eof

```

Figure 3: This document is an example of the background file from which the name and address records were "pulled" for the typing of the letter in Figure 1.

SUMMARY OF TEXT EDITOR COMMANDS

The following list provides a partial summary of the commands which are available in the text editor. Each command may be abbreviated by the use of the characters shown as capitals, the use of the full command is optional. Items shown in angle brackets < > are to be inserted by the user. Items shown in square brackets [] are optional. The * means that the command applies also to a star file. A "range" is a set of line numbers such as 1.01-25., while "range sequence" may be a set of several ranges, such as 1.01-25, 19.21.307-24,56. The symbol "ln" means a particular line, such as 21.307, while "inc" means line number increment, such as 1 or .001. A "string" is a set of characters designated with single or double quotes.

Alter<ln> - allows a line to be edited.

Copy [*] <range>TC<ln>[I inc] - transfers text from one set of lines to another or from a star file without disturbing the original set.

Delete [range sequence] - deletes a set of lines from file edit document.

Edit <file> [disk] - opens an edit file on the specified disk.

END - Ends the word processing session and closes all files.

Find [*] [All] <string><range sequence> - searches for occurrences of a particular string.

Goto [ln] [I inc] - directs the editor to go to a particular line, resets the line increment.

Help [command] - provides more extensive information about a particular command. Also operates a set of useful utilities.

List [*] [Number] [Printer] <range sequence> - causes the document to be listed without executing the embedded formatter commands.

Move <range>To <ln> [I inc] - transfers a set of lines while deleting the original set.

Number <ln> [I inc] - rennumbers the document with a specified line increment.

Print [*] <range>[options] - causes the document to be passed through the formatter.

Replace [All] <string1>With <string2> <range<sequence> - searches and replaces specified strings.

Star <file>[disk] - opens a star file on the specified date.

Substitute <string>For %<digit> - allows a symbol %x to stand for a set of text commands.

Tab <number>[, <number>]19 - sets tabs on the video terminal.

SUMMARY OF FORMATTER COMMANDS

The following list provides a summary of the commands which are available in the formatter. Each command may be represented by its first three characters in upper case, lower case, or mixed case; any other characters are optional. All commands (including the simple @ command) must be followed by a non-letter or by the end of a line. A blank is the usual character which separates a command from any following text. Commands may be "nested".

<text> stands for any series of characters (including other complete commands) which are to be treated as an argument to the command.

<text> may be terminated on the right hand side by a single @ command.

<numb> stands for a number.

@ASIs - Print the following lines as they are typed; i.e., each command is interpreted but no word movement from one line to the next takes place.

@BOLD <text>@ - Each character within text is overprinted with itself. This gives a pseudo-boldface type.

@CAPitalize <text>@ - All lower case letters within text are converted to upper case.

@CENTER <text>@ - The characters within <text> are centered between the current line position and the right hand indentation margin. If the text will not fit then no centering is done.

@DATE - The date that was typed in when the editor was initialized replaces this command.

@DIMension <numb>, <numb> - The physical page size (width, height) is specified in inches, otherwise system standard is used.

@DISplay <text>@ - The characters within <text> are displayed on the video terminal; they are not inserted in text. <text> cannot exceed one line.

@END - All files are closed and the printing is terminated. The editor is restarted.

@EOF - When encountered in the edit file, this command is treated like @END. When an @GET command encounters an @EOF in the star file, the edit file line number specified in @GET is branched to.

@F <numb> - The value of the <numb>th field after the current record (specified by the @REC command) replaces this symbol.

@FILl - Justify line by filling out the line as close as possible to the right indent margin; word movement from one line to the next is done to accomplish this.

@FLOat <numb> - If there is enough room on the current page for <numb> inches of space, then the next line to be printed will be after this space. Otherwise printing begins after <numb> inches on the next page. [The maximum space that will be skipped is the physical space between the current top and bottom margin.]

@GET <numb> - The next record in the star file (specified by @REC) is obtained. If an @EOF is encountered or an end-of-file is reached, then execution continues with line number <numb>. [If no line number <numb> is in the edit file then an error halt is made.]

@GOTO <numb> - Execution continues with line number numb from the edit file. If <numb> is not found then an error halt occurs.

@HEAD <numb>text @ - On line <numb> of every page is printed a heading which consists of text. If <numb> is within the top and bottom margin then no heading is produced.

@HM <numb><numb> - The horizontal margins (left, right) are specified in inches, otherwise system standard is used.

@IF <text₁>@ relop <text₂>@GOTO<numb> - The relative operator is applied to <text₁> and <text₂> and the GOTO is executed if and only if the relationship is true. The relations relop are: @CON - contains; @EXC - excludes; @EQU - equals; @NEQ - not equals; (e.g. @IF ABC @CON B@GOTO 20 will branch to line 20 of the edit file.)

@IN <numb> - Set the left and right indentation margins numb spaces from the left and right physical margins.

@INL <numb> or **@INR** <numb> - Like @IN except only the left or right margin is indented.

@JUST - Justify the line on the left and right side with respect to the left and right indentation margins. This command is like @FILL except the line is internally padded with spaces to make the line right justified.

@NAME - The operator's name that was typed in when the editor was initialized replaces this command.

@NUMBER <numb><text>@ - On line <numb> of every page is printed a page number preceded by all text characters up to @. If <numb> is the same as the heading number then the characters and number are right justified on the heading line, otherwise the characters and number are centered on the page. If <numb> is within the top and bottom margins, no page numbering is done.

@OPEN <numb> file @ - On drive <numb> the file specified by the file is opened as the star file. @GET is the only command that references this file, except for the @F<numb> command. Any fielded line (line referenced by @F <numb>) may also contain commands that affect the edit file.

- @PAGE** - Skip to the top of the next page. If already at the top of the page, then don't skip.
- @P** - Start a new paragraph - skip the number of lines specified by **@SPACE** and tab.
- @QUERy** **<text>** - The characters within **<text>** are displayed on the CRT; they are not included in the printed text. The terminal then waits for input from the user. The user's response is then entered into the text at this point, replacing the **@QUERy** command. Commands may be entered.
- @RECOrd** - A new record is started in the current star file.
- @REMArk** **<text>** - Remark - **<text>** is ignored in the file.
- @SET** **@V<digit>** **=<text>** - The variable **@V<digit>** is set to the value of the **<text>**. **<text>** may contain other format or edit commands.
- @SKIP** **<numb>** - **<numb>** lines are skipped: Unlike **@FLOAT** the lines "wrap over" onto the next page if necessary, also **SKIP** specifies number of lines, **FLOAT** specifies the number of inches.
- @SPACE** **<numb>** - Set spacing to **<numb>** line per written line, i.e., **@SPACE2** sets double spacing. Spacing begins on the line after the current line.
- @T** - Tabulate to the next tab stop.
- @TR** - Tabulate to the next tab stop, right justify intervening text before this stop.
- @TSET** **<numb>**, **<numb>** - Set up to 20 tab stops, each number must be greater than the previous one.
- @UNDer** **<text>** - The **<text>** is underlined.
- @V /digit /** - The value of the variable **@V<digit>** which was specified in the **PRINT** command from the editor, or in the **@SET** command, replaces the corresponding **@V<digit>** in the text. Thus names, dates, etc. can be variables.
- @VM** **<numb>**, **<numb>** - The vertical margins (top, bottom) are specified in inches, otherwise system standard is used.
- @** - This command "closes" or "delimits" arguments to certain format commands. This command must be followed by a blank or be the last symbol on a line to assure that any characters after the **@** did not construe another command.

mits

**2450 Alamo SE
Albuquerque, NM 87106**