PCC PERTEC COMPUTER CORPORATION

ALTAIR™ WORD PROCESSING SYSTEM

October 1977

MITS NO. 102713

ALTAIR WORD PROCESSING SYSTEM

USER MANUAL

TABLE OF CONTENTS

SECTION 1 - HOW TO USE THIS MANUAL

This Manual has been prepared to assist you, the user of an Altair Word Processing System, in receiving the maximum benefit of the extensive capabilities of the package. This Manual has been organized to (1) show how simple using the system can be, (2) provide a quick guide to the user who wishes to obtain a brief explanation or make use of only the simpler commands, and (3) provide a comprehensive explanation of the concepts of the system for the user who wishes to obtain a thorough understanding.

The Altair Word Processing System is one of the most comprehensive systems on the market. Its commands can appear overwhelming if you try to learn them all at once; this is not necessary or even advisable. The system will allow you to use the editor and formatter to produce documents with only a few commands. Although this Manual is designed to be a comprehensive reference guide with only a short introduction, it will assist the beginning user in producing a document.

If you wish to:	In this Manual, see:
Have a quick introduction...	Section 5 - Start-Up Procedures which provide sufficient information to begin to use the editor, Section 6 - Summary of Text Editor Commands and Section 8 - Summary of Formatter Commands, which provide a quick reference to both sets of commands, and the examples in Section 14 - Applications and Examples. The Overview in Section 2 presents a fairly complete summary of the system.
Obtain a thorough understanding..................	Read the entire Manual with particular emphasis on Section 2 - An Overview and Section 4 - General Concepts. The detailed descriptions of the Text Editor and the Formatter, Section 7 and Section 9 provide several examples. The examples in Section 14 - Applications and Examples show the use of the system in a variety of situations.

If you wish to:	In this Manual, see:
Make a quick reference to a particular point.........	The Table of Contents
Interpret an error.........	Section 13 - Error Messages and Handling
Document procedures for your installation..........	Section 15 - User Installation Procedures
Interface the System with other Software.............	Section 12 - Systems Information and either the Altair BASIC Manual or the Altair DOS Manual
Be trained on the use of the system.................	Continue reading this section.

This User Manual can be used as a self-teaching training manual. It is suggested that the "pupil" glance through the Manual once quickly to see how the Manual is organized without trying to comprehend all the information. Then, a careful reading of Section 2 - An Overview will provide much information about how the system functions. It is suggested that the pupil then refer to the examples of several text documents and the formatter results in Section 14 - Applications and Examples, particularly the first two examples.

The pupil should then learn-by-doing by actually using the machine to walk-through the examples. Section 5 - Start-Up Procedures will allow the user to begin to make use of the system. The pupil may wish to use a document already on a diskette or use the SAMPLE text file provided on each MASTER WORD PROCESSING SYSTEM diskette produced.

In addition, your local Altair Computer Center can provide assistance in training and teaching aids.

SECTION 2 - AN OVERVIEW OF THE ALTAIR WORD PROCESSING SYSTEM

Background - Word processing systems come in all shapes, sizes, capabilities and prices. They range in design from typewriters with magnetic tape memory to automatic typewriters with self-contained memory to text editing and formatting systems implemented on large-scale computer systems. Whatever the configuration, word processing systems generally have the basic function of the storage,modification and typing of repetitive business documents. From this foundation, word processing systems begin to differ in how they implement these basic functions and in the additional capabilities they present. Even the simplest systems provide for some editing of text, usually as it is being entered and perhaps at some later time. Features greatly sought by word processing users include 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, being able to search a body of text for a particular character sequence, and having the embedded central characters easily accessable for corrections.

The "parent" of modern word processing machines - the stoic MT/ST (Magnetic Tape/Selectric Typewriter), first marketed by IBM in the 1960's, allowed for only a very limited use of these desirable features. Since the MT/ST machines stored their text on magnetic tape cartridges, the text could only be handled sequentially. The advent of the rigid or "hard" 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 rigid 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 have crossed the threshold of a major decrease in cost and increase in capability. The use of the Altair computer as the heart of a shared logic word processing system means that old cost barriers have been broken while performance has substantially increased. When MITS, Inc., introduced the Altair computer in December 1974, the small general-purpose computer 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 applications. The Altair Word Processing System represents a significant step in bringing highly sophisticated applications software to the Altair computer user.

General Description - The Altair Word Processing System is a flexible and powerful text editing and formatting system which operates on an Altair 8800 computer equipped with a printer, a video terminal and at least one floppy disk drive. The package utilizes several machine language routines, as well as Altair Disk BASIC, which allows the computer system to be used for a wide range of other general-purpose computing, including accounting systems. The Altair Word Processing System 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 System is an extremely flexible and powerful system . At first glance, the list of commands for the editor or 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 Text Editor - The Word Processing System can be described as two modules - the text EDITOR and the text FORMATTER. The editor allows the user to create, modify and store the textural document. A set of EDITOR COMMANDS allow 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 blind "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 source document - not merely their result. This manner of operation gives the user much more control over the document to be printed and simplifies changing or correcting formats.

The editor handles text as a set of lines each up to 120 characters in length. 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 500. The size of a single document can, therefore, be up to 60,000 characters (about 20-25 single-spaced, letter-sized pages). Four "full" documents of 500 lines each or many shorter documents can be stored on a single floppy diskette. Several documents may be linked together to form a "book" of virtually unlimited length. The storage of documents on the inexpensive diskette allows the storage of a large number of documents in machine readable forms for very low cost. The Altair Word Processing System 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.

The Altair Word Processing System provides a variety of editing commands which facilitate the creation and maintenance of text. The editor operates in two modes: (1) the TEXT MODE for inputting of lines of text, and (2) the EDIT MODE for the inputting of an editor command. The editor flip-flops between the two modes when a null line (a line with nothing but a carriage return) 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 commands, a "?" is presented by the system. Most of the editor commands may be abbreviated by 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 will create a new document file with that name if the user so requests. 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. As lines of text are created, the editor moves to the next sequential line number, called the current line. The user may reset both the current line number and the increment at any time with the GOTO <line number> I <increment number> command. For example, G10 I.5 would cause the line numbering to be 10.0, 10.5, 11.0, 11.5, and so forth.

If a command is typed incorrectly, the system will respond with a brief error message. At any time, a user can utilize the HELP commands to receive more detailed information about features of the system.

When a document is open for editing,a number of commands are available to search 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 though 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 editor 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 formatter commands. The LIST command should be compared to the PRINT command which causes 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 Altair Word Processing System is greatly expanded through the use of STAR FILES. A star file is simply one text document which is called upon as a reference or source document for another text 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 the star file. Of course, a user has edit access to the document which is identified as the star file and that particular document could be opened as an edit file and any desired modifications made. The examples which are shown in Section 14 - Applications and Examples 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 any of the 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 DELETE command allows the deletion of single lines or large blocks of text.

Another useful command is REPLACE. The command statement:

```
REPLACE ALL "stock#" WITH "Harris Supply Inventory Control Number"
```

would make the indicated replacement throughout the document whereever stock# occurred. It should be noted that editor commands may be abbreviated to simplify typing for the user. Except for the END command, only the first letter of a command need be typed. The last example could be input as:

```
RA "stock#"  W"Harris Supply Inventory Control Number"
```

Another powerful command is the SUBSTITUTE command which allows text strings to be represented with up to ten substitute codes (the % sign followed by a digit 0 to 9). For example, the user who had entered:

```
          ? SUB "Georgia Institute of of Technology" For %0

          ? SUB "Massachusetts Institute of Technology" F %1
```

would type only:

```
          ? EDIT "Namelist"
          1.000 Dr. John SMith
          2.000 %0
          3.000  blank line -[a space or two followed by a
                 carriage return]
          4.000 Dr. Tom Jones
          5.000 %1
          6.000  null line -[only a carriage return typed]
          ? LIST
```

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 Altair Word Processing System editor. For example, information about document files may be displayed and document files on a diskette may be copied to other diskettes, renamed, or deleted. In addition, the system can"read" files created by other computer programs and treat them as text document files.

The Text Formatter - While the text editor offers the user a number of very useful editing features, the other side of the Altair Word Processing System - 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 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 source text 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 source document without filling or using a justified right margin. Likewise, a new page can be "forced" to start whenever a user desires.

The formatter commands are special text statements which are input and modified in the editor like any other Each formatter command begins with an "@" 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 both boldface and underlining).

When a user first initializes the Altair Word Processing System,or at any subsequent time, a set of standard default parameters may be 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 inch 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 specifications within the document - the default parameters would control. However, if a document were to be printed on 8.5 x 14 inch paper, the user would simply specify in the document (shown here after a remark line):

```
1.000  @REM Jones v. Smith, Brief on 12b(6) motion@
2.000  @DIM8.5,14
```

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:

```
2.000  @DIM8.5,13
```

and reprints the document.

The formatter contains all the usual formatting commands such underlining, centering, indentation, headings at a specified line, pagination at a specified place, right justification, paragraph and page definition, skipping lines, multiple spacing, and tabulation. 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 affixing photos or drawings to a document. The indentations may be on the left, or the right, or both.

The real power of the Altair Word Processing System, 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 video 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 as a command. This powerful feature allows a document to query the operator to have text inserted at that point, 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 additional document files 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 files may be shifted to another line out of sequence 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.

The @LINK command allows another document file to be linked on the existing file being printed as if it were part of the existing file. This allows an unlimited number of documents even on different diskettes, to be linked together to produce a single document of sequential page numbers of virtually any length. (The only real limit on the length of a document is that page numbers are limited to four digits, therefore, a document longer than 9999 pages will not have correct page numbers.)

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 causes 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 is 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 powerful method of having relevant information inserted at the appropriate place in the document without actually altering the source text document. For example, a form letter may contain the variable @DATE. Each time this letter is printed, it is not necessary 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:

```
54.000 @INL30
55.000 Sincerely,
56.000
57.000
58.000
59.000 @V6
```

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

SECTION 3 - HARDWARE CONFIGURATION

The Altair Word Processing System is designed to be used on a disk-based, Altair 8800 series computer using the Altair Disk BASIC, Version 4.0 Software. The actual hardware configuration may vary from user to user, but certain minimum hardware components are necessary:

Central Processing Unit - Altair 8800 series computer utilizing Altair Disk BASIC software is required. The minimum internal memory requirement is 48K (49152 bytes).

Video Terminal - All typing entries into the system are made through a video terminal (CRT) using a typewriter-like keyboard with a television display. The video terminal is used for all editing functions. The standard configuration of the Altair Word Processing System uses a 24-line, upper/lower case ASCII terminal.

Printer - The printing of documents is accomplished using a high-quality, typewriter-like precision printer. The standard configuration for the Altair Word Processing System uses the Altair Q-70 Printer. Although a number of the formatting enhancements are lost by using other printers, many other upper/lower case printers can be used.

Disk Units - Altair flexible (floppy) disk units are used for master program storage and for the storage of document files. The floppy diskettes (which resemble 45 RPM phonograph records) can be moved in and out of the disk drive to provide unlimited document storage capability. Although a single disk drive is the minimum required to operate this system, substantially improved performance is obtained through the use of dual disks.

The Altair Work Processing System is designed to automatically interface with a number of different printers. When the system is initialized each day, the system checks to see whether a particular printer interface has been initialized; if not, the initialization for the existing interface is performed.

The flexibility of the hardware requirements allows the Altair Word Processing System to be used in a variety of business situations. Your local Altair computer center can provide additional information about the specific hardware which will best serve your needs.

THIS PAGE LEFT INTENTIONALLY BLANK

SECTION 4 - GENERAL CONCEPTS OF THE ALTAIR WORD PROCESSING SYSTEM

• The general concepts of the operation of the Altair Word Processing System are discussed in this section. This information is provided to give a user a conceptual framework for viewing what is happening in the system.

The Text Editor/Formatter - The Altair Word Processing System is organized in two modules, the TEXT EDITOR and the FORMATTER, to give the maximum amount of flexibility in preparing documents and having them typed on the printer. The basic concept of the system is that the user can"see" the entire contents of a file document, including line numbers and embedded format commands. In this way, the user has full control over the way the document is organized; there are no invisible control codes that must be accounted for even though they cannot be seen. When the user instructs the system to LIST all or part of a document file, the system displays on the screen (or types on the printer) exactly what is contained in the document file. Although this listing is unformatted (i.e. margins not set, etc.) the embedded format commands are exhibited and can be altered just as any other text by using the TEXT EDITOR. When the user instructs the system to PRINT all or part of a document, the system passes the document through the FORMATTER, and the document is typed on the printer, having executed all the embedded format commands.

Because the formatting is not done until a document is PRINTED, the document is not restricted to any particular margin settings, line spacing, tabulation, page sizes, or other formats. In fact, the length of a "line" in the document as it is stored on the diskette and initialized in the system can be up to 120 characters in length. Until the document is passed through the FORMATTER by the PRINT command, the page margins have no relation to the line lengths of the lines in the document. This feature allows much greater versatility in inserting or changing text. (see the raw source text documents and the "finished" formatted documents in Section 14 - Applications and Examples.)

The Current File and the Star File - the document file that is currently being edited is known as the current file. A file becomes the current file when it is named in the ?EDIT "name" command. For example, ?EDIT "Letter 3" makes the file "Letter 3" the current file. If "Letter 3" is not found on the diskette mounted on the specified disk drive, the operator is informed that the file named "Letter 3" does not exist. At this time, the operator is given the option of creating the file "Letter 3" and making this the current file. As the current file, a document can have all of the text editing functions performed upon it, such as entering text, renumbering lines, moving text, or deleting lines.

The Altair Word Processing System allows you to use another existing document file as a reference or a master file when using the editor. This reference file is known as the star file because it is referenced in the editor commands by the addition of "*" to the command. The star file is simply another text file in which you can search, list or copy to your current file. The star file is handled just like the current file except that the star file cannot be modified in any way; that is, you cannot renumber, delete, or in any way change the contents of the star file. (Of course, you can call up the file that was the star file as the current file and make any of these modifications.) In this way, the star file can serve as a master file or a reference file and not be subject to inadvertent changes.

A file becomes the star file by being "opened" with the ?OPEN "name" command. For example, ?OPEN "Letter 2" would close any previously opened star file and then designate "Letter 2" as the new star file. If "Letter 2" was not found on the diskette in the specified disk drive, the operator would be so informed and no file would be opened.

As the star file, a document may be used for reference only. Text may be copied from the star file to the current file, the star file may be searched, listed or printed; but no operation is permitted which will alter the star file. Only the current file may be modified by being edited. Therefore, the COPY* command is valid for the star file since it does not alter the original lines, but the MOVE* is not valid since it would have the effect of deleting the original lines.

Any number of star files may be used, one at a time, to provide text for the current file. The star file is used both by the editor and the formatter, although a couple of important differences exist in the usage. When used in the editor, the star files can be used as master documents, each providing portions of text to be copied to the current file, or being available to be searched or listed. The star file in the formatter is slightly different. Reference is made in the formatter to the star file, once it has been opened, by the use of the @GET and @FIELD commands in the current file. These commands cause the next record - a set of text lines called fields - to be brought into and processed by the formatter. The star file in the formatter can contain formatter commands which are acted upon.

The syntax (that is, the form) of the commands which access the star file is slightly different in the editor and the formatter. (See the description of the ?OPEN command in Section 7 and the @OPEN, @STAR and @LINK commands in Section 9.)

The text document files reside on diskettes as computer readable files. Each of these computer files carries with it the text of the document and a set of indexes which are used to quickly access lines within the document. These indexes allow new lines to be inserted, deleted or moved around without having to physically move the text, by simply changing the indexes. Each of the text document files has a name which can be up to eight alphanumeric characters in length (but may not begin with a "$" or a "#" character.) Each diskette keeps track of its individual file names and therefore, it is possible to have two text files with the same name, one each on different diskettes. Computer files used by the system also reside on the diskettes. These systems files have names which begin with a "$" or a "#" character. A listing of text document files on a diskette can be obtained by using the HELP FILES command, which lists the names of the text files. Since the Altair Word Processing System uses the operating system and file management of Altair BASIC, it is possible to create a disk file with Altair BASIC or Altair DOS programs which can be "read" by word processing. (See Section 12 - Systems Information).

Line Numbers and Ranges - One of the central concepts of the Altair Word Processing System is the concept of how the system uses line numbers. It is quite common for a document to have its lines numbered. For example, a draft of a contract may have a set of line numbers for each page on the left hand side of the page. What happens if you need to insert a new line between lines 2 and 3 -- does the new line become line 3, line 3 become 4, etc? In other words, the relation of the line numbers in the original set of lines is lost unless fractional line numbers are available. The Altair Word Processing System solves this problem and greatly increases flexibility by using decimal line numbers which can range from 0.001 to 999.999; which means in our example, we could have inserted line number 2.1 (or 2.05 or 2.003). It should be noted that although there are almost a million possible line numbers, the document may have a maximum of 500 lines. Leading zeros are suppressed so that line "001.100" is simply displayed as line "1.100".

New lines are entered into the document either by being typed or by being copied or moved into the file at a particular place. When lines of text are initially typed into a newly-established file, the system presents line 1 for the typing of text, followed by line 2, followed by line 3, etc. We say that these line numbers have an "increment of 1" because they are increasing by 1 each time. However, if we want to insert several lines of text between lines 2 and 3, it would be useful to have the increment set at some fraction, such as .01 so new lines of text would go in as 2.01, 2.02, 2.03, etc. The command GOTO provides the user with the opportunity to reset the current increment with the use of the "Inc" option.

The TEXT EDITOR commands are typed into the system in response to a "?" displayed on the screen -- the Edit Mode. Text itself is typed into the document in response to line numbers being dsiplayed on the screen, such as 1.05 - the Text Mode. The editor "flip-flops" between the edit mode and the text mode when a null line---a line with no characters--- is typed by depressing the carriage return (C/R) key. For example:

```
           1.05  @T @T  Sincerely, (C/R)
           1.06  (C/R)
          ?      (C/R)
           1.06  (C/R)
          ?      (C/R)
           1.06  (SPACE)  (C/R)  (The space character is a character)
           1.07
```

The carriage return with nothing else on the line causes the editor to flip-flop between the current line number (waiting for text) and the editor mode "?" (waiting for a command).

Searches and Text Matching - Some of the most useful functions of the Altair Word Processing System are derived from its ability to search a document for a particular set of characters or words (called a 'string') to find a match. The string which is matched can then be displayed on the screen in the context of its line (the FIND command) or replaced by another string (the REPLACE command).

The single quote (') and double quote(") marks have special meaning for commands that search for strings. The quote symbols are known as string delimiters. The single quote (') means the search string is located in the document just as it occurs, regardless of what precedes or follows. The double quote (") means the search string is located only when it is preceded or followed by a delimiter, such as a space or a period. A search string may use either single (') or double quote (") or both. 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.

The Way the Formatter Operates - The formatter is a special part of computer software within the Altair Word Processing System which takes a specified text document file and passes it to the printer. While the document is being processed by the formatter, and passed to the printer, the document formats are being controlled by formatter commands embedded within the text document. The formatter commands begin with the character '@' and each have a specific action which is taken by the printer. (The first example in Section 14 - Applications and Examples should be examined at this point for illustration.) When a person begins to type a letter, certain formats are usually assumed. The left and right margins are set, typing is begun at a certain distance from the top and concluded at a certain distance from the bottom. Likewise, the typist may indent, set tab stops, double-space, leave several lines for insertion of a drawing and place page numbers and a heading on the page. Each of these items relates to the format of the typed page and are but a few of the many things which can be controlled by formatter commands in the Altair Word Processing System. (Each of these formats need not be specified for each document. The default file, established by a user,is used to supply these formats when they are not included in a document. See the discussion of the default file below.) In addition,the System has extensive capabilities for more powerful instructions, such as pulling inserts from other documents and testing logical conditions. The Altair Word Processing System is a powerful set of commands for controlling text documents.

The embedded formatter commands take the form @X where 'X' is a particular command statement. The command may be followed by operands, that is something on which the command operates. These operands may be a set of characters or words, a document file name, or a set of numbers. In some cases, the end of the command operands must be designated by the use of the '@' symbol which is the '@' character followed by a space or by the end of the line.

The Default File - The Altair Word Processing System makes use of a default file to specify 'standard' formats for such things as margins, page size, spacing, tab settings, and indicating whether continuous form paper is being used. These default file values are established so that they need not be repeated in every document; only when the formats in the document to be printed differ from the default values do the format values need to be specified in each individual document. The values in the default file may be redefined at any time by using the HELP DEFAULT command in the editor. See Section 10 - Redefining the Default File for a User's Installation.

THIS PAGE LEFT INTENTIONALLY BLANK

SECTION 5 - START-UP PROCEDURES

The Altair Word Processing System is a set of programs which reside on the MASTER WORD PROCESSING DISKETTE. The main programs which control the word processing functions are reached through a program called "WP MENU" which allows the selection of the various functions. The WP MENU program is loaded through Altair Disk BASIC, Version 4.0. However, BASIC is not included on the MASTER WORD PROCESSING DISKETTE and must be loaded into the machine from another disk.

Bringing-up BASIC for word processing should be performed in the same manner as when bringing up BASIC for the other Altair software packages. Altair Disk BASIC is loaded from an appropriate disk and the inquiries answered as follows:

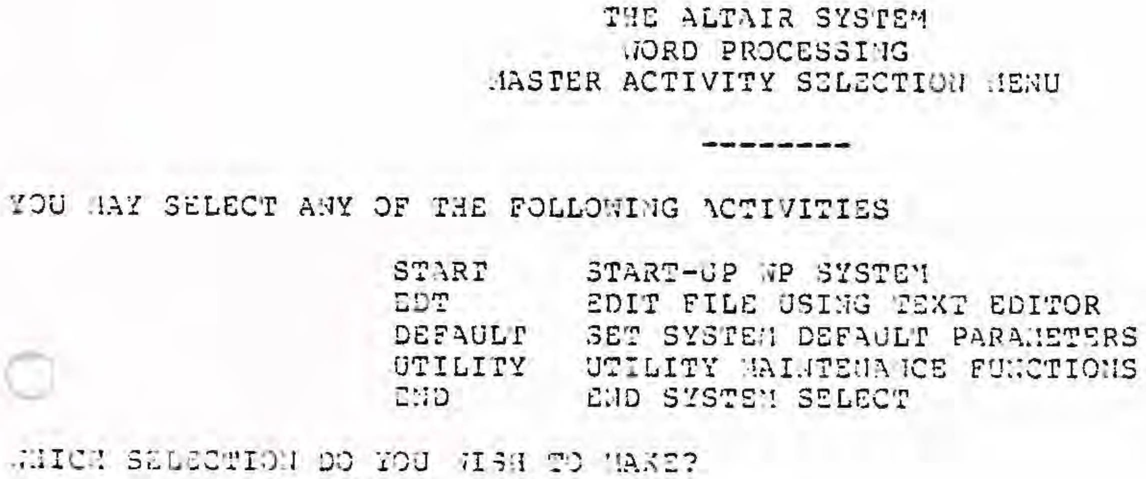
```
MEMORY SIZE?   46000
HIGHEST DISK NUMBER?  0  (or 1)
HOW MANY FILES? 5
HOW MANY RANDOM FILES?  5
```

The diskette containing BASIC is unloaded from the disk drive and the MASTER WORD PROCESSING DISKETTE is inserted in disk drive 0 and a MOUNT 0 command is issued. After mounting the MASTER WORD PROCESSING DISKETTE, type in the command:

```
RUN "WP MENU"
```

which will bring the menu program into operation. If the hardcopy printer drivers have not been previously loaded and initialized, the system will make note of this and will automatically add the proper drivers. It will take approximately twenty seconds for the printer driver to be loaded and initialized. (Note: the START program will allow the printer drivers to be reinitialized at any time.)

The MENU program will then display the following menu selection:



```
                              THE ALTAIR SYSTEM
                               WORD PROCESSING
                        MASTER ACTIVITY SELECTION MENU

                                  --------

YOU MAY SELECT ANY OF THE FOLLOWING ACTIVITIES

                START      START-UP WP SYSTEM
                EDT        EDIT FILE USING TEXT EDITOR
                DEFAULT    SET SYSTEM DEFAULT PARAMETERS
                UTILITY    UTILITY MAINTENANCE FUNCTIONS
                END        END SYSTEM SELECT

WHICH SELECTION DO YOU WISH TO MAKE?
```

The main word processing program is called when item "EDT" of the menu is selected. The system will display:

```
            The Altair Word Processing System
                     Copyright 1977
                          ASDC
```

the date to be used. The entry that is made here will correspond to the variable @DATE whenever @DATE is referenced in a document. The next prompt will be for the name of the user. The entry here will correspond to the @NAME variable used within the system. (See Section 9 for a discussion of the use of @DATE and @NAME in the text documents.)

The Altair Word Processing System is in the edit mode when it first "comes-up", indicated by a '?'. Typing HELP (or just H) will provide directory of other HELP commands which may be accessed by the user needing more information.

The MASTER WORD PROCESSING DISKETTE contains space for a few text document files. However, it is suggested that only frequently used, short files be placed on this diskette and most text document files be put on other diskettes. Before a diskette can be used for the first time, it must be initialized. A utility program is provided for this purpose. See Section 11(A) - Use of the Utilities - Diskette Initialization. A diskette which has been initialized for another Altair computer language, such as Altair BASIC or Altair DOS, may be used without being reinitialized. Likewise, a diskette which has been used in the Altair Accounting System may be used. DO NOT INITIALIZE ANY MASTER DISKETTES, SUCH AS THE MASTER WORD PROCESSING DISKETTE OR THE DISKETTE WHICH CONTAINS ALTAIR DISK BASIC, AS INITIALIZING DESTROYS THE PROGRAMS ON THE DISK.

SECTION 6 - SUMMARY OF TEXT EDITOR COMMANDS

The following list provides a summary of the commands which are available in the text editor. A detailed description of the command is provided in the next section. Each command may be abbreviated by the use of the characters shown in capitals, the use of the full command is optional. The text editor will actually make the commands capitals, even if typed in lower case. Items shown in angle brackets < > are to be inserted by the user. Items shown in square brackets [] are optional. The * means that the command may also apply to a star file. A "<range>" is a set of line numbers such as 1.01-25.0, while a "<range sequence>" may be a set of several ranges, such as 1.01- 5.0, 19, 21.307-24.0, 56. Omitting a range or range sequence entirely means the entire file. The symbol "<ln>" means a particular line number 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. [A command is an editor command in quotes or is an '@' in quotes.] A <file> is a file name optionally followed by a comma and the disk drive number [,<drive>]. A null line flip-flops the editor between the edit mode (?) and the text mode (line numbers). For examples, see Section 7.

Copy [*] <range> TO <ln> [I<inc>] - transfers a set of lines from one text (which can be a star file) to another portion of the text

Delete [<range sequence>] - deletes a set of lines from the edit document

Edit <file> [, <drive>] - opens a edit file on the specified drive

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, and optionally resets the line increment

Help [<command>] - Operates a set of useful utilities. (See the description of HELP DEFAULT in the next section.)

ist [*] [Number] [Pr] <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>] - renumbers the document with a specified line increment starting at ln

Open file [,<drive>] - opens a star file on the specified drive

Print [<file>][<parameter>9]<range> - causes all or part of a specified document to be passed through the formatter. The parameter setting @V digit = string, etc.

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

Substitute<string>for %<digit> - allows a symbol % to stand for a set of text commands where x is a digit 0 to 9. (NOTE: a space is required between the '%' and the 'digit' when using the SUBSTITUTE command; when the substitution is actually used, however, no space is allowed.

Tab<number>[,<number>19] - set editor tabs on the video terminal up to 20

SECTION 7 - THE TEXT EDITOR COMMANDS - DETAILED DESCRIPTION

General Use - The Altair Word Processing System text editor manipulates text through the use of a number of text editor commands. These commands are typed in response to a ? displayed on the screen. The user types an appropriate command to perform whatever function is desired. To switch between text input mode (line numbers displayed) to the command mode ("?" displayed), simply type a null line, that is strike nothing but a carriage return. The system displays all editor commands in upper case, even if entered in lower case.

Command Syntax Conventions - The commands available for the use in manipulating text material are shown on the following pages. After each command, the proper syntax (that is, the proper form) is listed. The syntax shows what must or can be specified with each particular command to achieve the desired results. Except for the END command, only the first letter of an editor command must be typed to execute the command, hence the minimum required letters are shown below as upper case. For example, the Goto command means that the editor is to go to a specified line number; G or GOTO will execute the command. The following conventions are used to indicate the syntax required:

Upper case letters indicate the minimum to execute the command.

<> Means that whatever is within these angle brackets is to be typed by the user. The angle brackets themselves are not typed.

[] Means that whatever is within these square brackets is optional and need be specified only where that option is desired. The square brackets themselves are not typed.

<file> Means that a file name is required (set within quotes) of up to 8 alphanumeric characters. The file name may not begin with $ or #, lower case differs from upper For example, 'Letter' is a valid file name, different from 'LETTER'.

<inc> Means an increment is set (such as .01, 1, or 10).

<ln> Means a line number is required (such as 81, 81.5, or 81.505)

<name> Means a name is typed within quotes

<num> Means a number is typed

[,<drive>]

Means a disk drive number (0,1, 2) may be specified. Omitting the disk drive number will cause only drive 0 to be checked or referenced. Therefore, if another drive is required, it must be specified.

<range>

Means that a set of lines may be typed in the form 1-30.5, which would specify all lines between 1.000 and 30.500. A single line number may be typed. The symbol # represents the last line in the document; therefore, 30.5-# means all lines between and including 30.500 and the last line of the document. Omitting the range where required means the entire file is to be included. The limits of the line numbers are 0.001 to 999.999, almost one million possible line numbers even though the maximum number of lines in any single text document is 500.

<range sequence> or
<range seq.>

Means a set of ranges in the form 1, 3, 5 - 9.5, 27-35, 39. This specifies that lines 1 and 3 (but nothing in between) are included, as are all lines between and including 5 through 9.500 and lines 27 through 35, and line 39. The same conventions apply to 'range sequence' as to 'range'.

<string>

Means a set of characters within single or double quotes such as 'hello there' or "hello there"; the single or double quotes have special meaning only in the search commands FIND and REPLACE.

The Text Editor Commands are:

Copy [*]<range>To<ln>[I<inc>]

The copy command copies a <range> of lines in the current file (use no*) or from the star file (use *) to the current file starting the new lines at line number <ln> with the current or reset increment <inc>. Unlike the MOVE command, the COPY command does not delete the original lines.

Caution must be used in selecting <ln> and <inc> because copying can overwrite existing lines.

Examples of the use of the COPY command are:

```
?C7 To 25.1
```

Copies line 7 to line 25.1 (if line 25.1 already existed, the old line will be overwritten).

```
?C*35-50 To 3.5 I .01
```

Lines 35 through 50 are taken from the star file and copied into the current edit file beginning at line 3.5 and incrementing 3.51, 3.52, etc. If the star file had contained only integer numbers (35, 36, 37,etc.) there would be no more than 16 lines in the star file in the range 35-50. In this case, the line numbers of the copied lines in the current file would be 3.50 through 3.65. However, if the star file really contained 200 lines in the form 35.00, 35.01, --- 49.08, 50, the resulting lines in the current file would be 3.50 through 4.49, an occurrence that might produce unexpected and undesired results.

Delete <range sequence>

The DELETE command deletes the lines specified in <range sequence> from the text document (file). Omitting the <range sequence> assumes the entire file, but because of the severity of this action, the system will ask you to double check this.

Examples of the DELETE command are:

```
?DELETE  75.03-75.999
```

Deletes lines 75.03-75.999 but would leave line 76.000 intact.

```
?DELETE 7,31.02-35.9,76-138.5
```

Deletes lines 7, and other lines indicated.

```
?D
```

Deletes the entire text document file. Because of the severity of deleting the file, the user will prompted to make certain that this action is desired.

```
?D* 17-21
```

Is an invalid command since deletions are attempted in the star file.

Edit <file> [,<disk>]

The EDIT command specifies which file is to be the file currently being edited. If the file already exists on the diskette mourted in the specified disk drive, that file will be opened. If not, a file by that name will optionally be created on that diskette. The file name may contain up to 8 alphanumeric characters but may not begin with $ or #.

Examples of the EDIT command are:

```
?EDIT "Letter 3" or
?E"Letter 3"
```

(NOTE: File "LETTER" is not the same as 'Letter'.) The file which is specified by EDIT becomes the current file. If you desire to edit another file, simply enter the command EDIT "name2". The first file is closed and the new file "name2" is opened as the current file.

```
?E'TEST-167',2
```

Opens for editing file 'TEST-167' on disk drive 2 (the third drive on a system)

```
?E'XYZ$7$'
```

Opens for editing file 'XYZ$7$' on disk drive 0.

END

The END command causes the current file being edited to close properly by storing the special indexes in the file. If the Word Processing System is halted while in operation, by turning off the power for example, the current file document may be improperly closed. It is important to use the END command to end the Word Processing use. It should be noted that an improper closure of a file is not catastrophic--the system automatically resurrects the improperly closed file when you try to EDIT or OPEN it. However, this process may take an extra minute or two.

Find [*] [All] <string><range sequence>

The FIND command searches the current file or the star file (by using *) to locate the first or all (by using A or All) occurrences of a specified <string> of characters in a given <range sequence> and list on the screen those lines where the occurrences exist. The single quote and double quote have special meaning when used with the character string. The single quote means that the character sequence will be found regardless of whether a delimiter exists. A double quote requires the presence of a delimiter for a match to occur and the string to be found. See discussion in Section 4 - General Concepts under "Search and Text Matching". Upper and lower case letters are not interchangeable in searches.

Examples of the use of the FIND command are:

```
?F "Jones"
```

Would find the first Jones in the entire file where delimited by spaces, period, etc.

```
?F*A'a'  1-100
```

Would find all occurrences of the letter 'a' in the lines 1 through 100 in the star file regardless of whether delimited.

```
?FIND A"jack'
```

Would find all occurrences of 'jack' in the file if delimited at the front. That is, this would find the words jack and jacket, but not blackjack or Jack.

Goto [<ln>] [I<inc>]

The GOTO command directs the editor to allow text to be typed beginning on a certain line (ln). If no line number is specified, the editor goes to the next line beyond the end of the document. The line numbering increment may be reset with the GOTO command. When a file is first opened for editing, the current line number is at 1 if it is a new file or at one past the end of the file if it is an already existing file; and the increment is set to 1 whether new or existing. The current line number appears when the editor is switched between the edit mode ('?') and text mode (current line number). The current line number is reset when GOTO's are used and when text is manipulated with COPY or MOVE commands.

Examples of the GOTO command are:

```
?G#
```

Goes to the end of file

```
?G1
```

Goes to first line of file

```
?G100 I.01
```

Goes to line 100 and also resets the increment to .01.

```
?GOTO17
```

Goes to line 17. The increment remains at whatever had been previously set

```
?GOTO20  I5
```

Goes to line 20 and sets the increment at 5.

List [*] [Numbered][Printer]<range sequence>

The LIST command directs the listing of the document exactly as it is contained in the file. This means that no formatting is performed and the embedded format commands are visible. The LIST command shows what is in the file while the PRINT command prints the final document in its formatted condition. The listed file may be listed with line numbers (N) and may be listed on the hard copy printer (P) rather that on the screen.

Examples of the use of the LIST command are:

```
?LIST*
```

Lists the star file on the screen without line numbers

```
?LNP
```

Lists the current file on the printer with line numbers

Move<range>To<ln>[I<inc>]

The MOVE command moves the lines of text specified by <range> by copying them starting at line <ln> (with the increment specified by <inc>while deleting the original <range> lines from the file. MOVE is compared to COPY which does not delete the old lines. Caution must be used not to overwrite lines. See discussion in Section 4 - General Concepts.

Examples of the use of the MOVE command are:

```
?M1-25 To 40,I.01
```

Moves the text originally at lines 1-25 to the lines beginning with line 40. The new lines would be numbered 40.00, 40.01, 40.02, etc. The original lines would be deleted.

```
?MOVE 1-25T75
```

Moves the text originally at lines 1 to 25 to the lines beginning at 75. The increment here would be whatever was previously set.

A WORD OF CAUTION: The lines 1 to 25 in the last example may contain 26 lines of text or may contain hundreds of lines in the form 1.001, 1.002, 1.003, etc. This may cause unwanted results to occur if caution is not used in the MOVE and the COPY commands.

Number<ln> [I<inc>]

The NUMBER command renumbers the lines of the entire document. The new line number begins at <ln> with an increment set by <inc>.

Examples of the use of the NUMBER command are:

```
?N10,I.001
```

Renumbers the text document file beginning at line 1 with an increment of .001. The lines would be numbered 1.000, 1.001, 1.002, etc.

Open <file> [,<drive>]

The OPEN command tells the system which file is to be the STAR FILE. Any existing document may be a star file except the file currently being edited. If a file of that name doesn't exist on the specified disk drive, an error message is printed. Once open, a star file may be searched with the FIND* command, listed with the LIST* command, or copied with the COPY* command. Any other command would be an attempt to alter the star file and would be invalid.

Examples of the use of the OPEN command are:

```
?OPEN"Text", 1

?OPEN"XYZ$7X", 0
```

Print [<file>] [,<drive>] [<parameter setting>9] <range>

The PRINT command directs the current or the specified file to be passed to the formatter. The formatter will perform the operations directed by the embedded formatter commands and print the "finished" document. The PRINT command is compared to the LIST command (which lists the raw source text as it is in the text document file, either on the screen or on the printer, without executing the embedded formatter commands). The PRINT command can specify that only a portion of the file be passed to the formatter by indicating line number 'range'. The PRINT command causes the formatter programs to be loaded from the MASTER WORD PROCESSING DISKETTE; therefore, the MASTER DISKETTE must be mounted in disk drive 0 when the PRINT command is issued. The PRINT command causes control of the system to be passed to the formatter while it is operating; control is returned to the editor when the formatter has completed its tasks. Therefore, the MASTER DISKETTE must must be in disk drive 0 when the formatter finishes. See Section 14 - Applications and Examples "Long Documents." The parameter settings provide for variables to be used in the formatter of the form @V digit, and can be set in the PRINT command.

Examples of the use of the PRINT command are:

```
?PRINT
```

Causes the entire current file to be printed.

```
?P @VO="John T. Jones, Jr."
```

Causes the entire current file to be passed to the formatter. The variable @V0 in the formatter is set to "John T. JOnes, Jr."

```
?PRINT "LETTER", 1 @V2="Bob" 15-70
```

Causes lines 15 through 70 of the text document file called "LETTER" on disk drive 1 to be passed to the formatter. (If the file is not found on the specified disk drive, an error message is printed.) The variable @V2 in the formatter is set to "Bob".

```
?P "LET01", 1 @V1="Bob"  @V2="James E. Jackson,
 President"  @V3="ded" 1-165.3
```

Causes the lines 1 through 165.3 of the text document file called "LET01" on disk drive 1 to be passed to the formatter. The formatter variables @V1, @V2, @V3 are set as indicated. Should any of these variables also be set in the text document, the setting made in the PRINT command will prevail. It is important to note that the disk drive number must be preceded with a comma, else the command will be interpreted incorrectly. Also, it should be noted that the 'range' specified in the PRINT command provides for only one set of line numbers (i.e. 100-150) and not for a range sequence (e.g. 100-150, 175-200, 237, 240-250).

Replace [All] <string1>With<string 2><range sequence>

The REPLACE command searches the lines of text specified by 'range sequence' and replaces the first occurrence (or all occurrences in a line if 'A' is typed) of 'string1' with 'string2' in each line of text specified. The REPLACE command uses the same conventions on text searching and matching for 'string2' as the FIND command: single quotes mean delimiters are not used while double quotes mean delimiters are required for a match to be made. See FIND command discussion above.

An example of the REPLACE command is:

```
          ?RA "Jones"W"Smith"
```

Would replace all Jones in the file with Smith.

The REPLACE command uses the ALL command option to signify whether the replacement is to take place on only the first match or in ALL matches in each line. (This use of ALL differs from the FIND command where the ALL signifies that all the lines in the specified range are to be searched rather than stopping after the first match is found). For example, in the text lines (underlining shown for reference):

```
          101.000  a big dog and a little dog
          102.000  the first dog that would
```

the REPLACE command:

```
          ?R'dog'W'cat'101-102
```

would produce:

```
          101.000  a big cat and a little dog
          102.000  the first cat that would
```

while if you had typed instead:

```
          ?RA'dog'W'cat'101-102
```

would produce:

```
          101.000  a big cat and a little cat
          102.000  the first cat that would
```

Substitute <string>For %<digit>

The SUBSTITUTE command allows a specified string of characters to be represented by a two digit code - a % character followed by a single digit (0-9). Whenever the code is encountered in the editor, it replaces the code for a fill string.

For example:

```
          ?S 'Atlanta, Georgia 30326" F% 4
```

Would allow the characters 'Atlanta, Georgia 30326' to be entered by typing only '4%'. whenever the % followed immediately by a digit is encountered, the system will try to make a substitution. If you wanted to type just a %4 for some reason (such as type this page), you would simply type %%4, the system enters %4 as text rather than as a substitute code.

The SUBSTITUTE command operates directly in the editor, it is not part of the formatter. When the % digit is encountered, the substitution is made immediately, the % is not passed to the text file. The SUBSTITUTE command is useful in typing text documents where repetitive items may be encountered.

For example, in typing a mailing list, it may be helpful to enter the substitution:

```
101.000  [null line]
      ?  S 'Denver, Colorado' F%1
      ?  S 'San Francisco, California' F%2
      ?  S 'California' F%3
      ?  S 'New York, N.Y.' F%4
      ?  S 'Atlanta, Georgia' F%5
```

then the typing of one of the % digit's specified would cause the substitution to be made:

```
      ?  [null line]
101.000  John Q. Jones, Esq.
102.000  One Embarcadero Place
103.000  [%2] [causes the editor to insert in text:]
         San Francisco, California [then you may
         type the zip code.]
```

The SUBSTITUTION command is not limited to text; formatter commands may be used as substitution. For example, if you were doing a number of titles which were in bold, centered, all upper case, and underlines, then the following:

```
? S'@BOL @CAP @CEN @UND 'F%1
? S'@ @ @ @ 'F%2
```

would allow immedate action of substitutions shown

```
101.000  %1 table of contents  %2
```

to produce:

```
101.000  @BOL @CAP @CEN @UND table of contents  @ @ @ @
```

Tab<number>[,<number> 19]

The TAB command allows you to set up to 20 tabs for entering text. The appropriate number of spaces are actually entered into the text at that point.

For example:

```
? T5, 10, 15, 20, 25
```

would cause the indicated number of spaces to be tabbed over when the tab key is struck.

The editor TAB command should be distinguished from the embedded tab @T which directs the formatter to print a tab on the printer upon output. The formatter tab @T tabs over the number of spaces set by the tab set command in the formatter TSET. The editor command ?TAB sets a tab stop for immediate movement when the 'tab key' on the keyboard is struck. Space characters are actually inserted in the text at that point.

For example:

```
101.000  [null line]
  ? TAB10,20
  ? [null line]
101.000 This is an example of how the Tab command
102.000 [tab key] operates
103.000 and how [tab] [tab]  two tabs
104.000 would look in a sentence.
105.000 [null line]
  ? LN101-104
101.000 This is an example of how the Tab command
102.000           operates
103.000 and how              two tabs
104.000 would look in a sentence.
  ?
```

On some keyboards, a TAB key may not be a separate key and the [Control]/T key may be used.

THIS PAGE LEFT INTENTIONALLY BLANK

SECTION 8 - SUMMARY OF FORMATTER COMMANDS

The following list provides a summary of the commands which are available in the formatter. A detailed description of the commands is provided in Section 9. Each command may be represented by its first three characters in upper case, lower case, or mixed case; any other characters are optional. (In the command descriptions below, the capital letters within a command must be stated, the lower case letters are optional and may be included for clairty purposes.) All commands (including the simple @ command) must be followed by a non-letter or by the end of a line. A blank (a space) is the usual character which separates a command from any following text. Some commands require that an operand, such as a number, be indicated without any space between the command and the number. Commands may be "nested"; that is, commands may contain other commands within them.

<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.
<text-120>	is like <text> except the length must be less than 120 characters, <text-31> likewise is limited to 31 characters
<numb>	stands for a number in the range 0 or 0.001 to 999.999
<integer>	is a number with no decimal point
<digit>	is an integer from 0 to 9

Items in brackets [] are optional and may be ignored.

Formatter Commands

@

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. Its use is defined by the commands indicated below as being closed with an @.

@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 psuedo-boldface type.

@CAPitalize <text>@

All lower case letters within text are converted to upper case.

@CENter <text-120> @

The characters within <text> are centered between the current left hand and right hand indentation margins. 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>[,<integer>]

The physical page size (width, height) is specified in inches, otherwise system standard is used. If the integer is not equal to 0, the separate rather than continuous forms are assumed.

@DISplay <text-120> @

The characters within <text> are displayed on the video terminal; they are not inserted in text.

@END

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

@EOF

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

@F<integer>

The value of the integer field (line) after the current record (specified by the @REC command) replaces this symbol.

@FIL1

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. No extra spaces are inserted to right justify the text.

@FLOat<numb>

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

@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> in the edit file. [If no line number <numb> is in the edit file, then an error results.]

@GOTo<numb>

Execution continues with line number <numb> for the edit file. If <numb> is not found, then an error occurs.

@HEAd<integer><text-120>@

A heading which consists of <text> is printed on line <integer> of every page.

@HM<numb>,<numb>

The horizontal margins (left, right) are specified in inches, otherwise system standard is used.

@IF<text1-120>@<relop><text2-120>@Goto<numb>

The relative operator <relop> is applied to <text1-120> and <text2-120> and the GOTO is executed if and only if the relationship is true. The relations are:

@CON - contains
@EXC - excludes
@EQU - equals
@NEQ - not equal

@IN<integer>

Set the left and right indentation margins <integer> spaces from the left and right physical margins.

@INL<integer> or @INR<integer>

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 lines right justify.

@LINk<drive><file>@

On <drive>, the file specified by the <file> is opened as the new edit file. The text which is to be formatted then comes from this new file beginning with the first line in the new file.

@NAMe

The person's name that was typed in when the editor was initialized replaces this command.

@NUMber<integer><text-120>@

On line <integer> 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 page number are right justified on the heading line, otherwise the characters and numbers are centered on the page. <text> must be less than or equal to four characters long.

@OPEn<disk><file>@

The specified file is opened on drive <disk> as the STAR file. @GET is the only command that references this file, except for the @F<integer> command. Any fielded line (line referenced by @F<integer> may also contain formatter commands.

@PAGe

Skip to the top of the next page. If already at the top of a page, then do not skip.

@P

Start a new paragraph - skip the number of lines specified by the current line spacing and indent.

@QUEry<text-120>@

The characters within <text-120> 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. The text (now containing the user's response) is formatted.

@RECord

A new record is started in the current star file.

@REMark <text-120>

Printing is supressed during processing of the @REM command; however, any commands embedded in <text-120> will be executed.

@SET @V<digit>=<text-31>@

The variable @V<digit> is set to the value of the <text-31>. <text-31> may not contain other @V values but may contain other format or edit commands producing text up to 31 characters long.

@SKIp<integer>

<integer> lines are skipped. This command will not skip past the top of the page.

@SPAce<integer>

Set spacing to <integer> lines per written line, i.e., @SPAce2 sets double spacing. Spacing begins on the line after the current line

@STAr<disk><file>@

Same as @OPEn

@SUBscript<text>@

The <text> is subscripted within the document.

@SUPerscript<text>@

The <text> is superscripted within the document.

@T

Tabulate to the next tab stop.

@TR

Tabulate to the next tab stop, right justify intervening text before this stop.

@TSEt<integer>,<integer>

Set up to 20 tab stops, each number must be greater than the previous one.

@UNDerline<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.

SECTION 9 - THE FORMATTER COMMANDS - DETAILED DESCRIPTION

This section describes in detail how the Formatter commands are used to control the typing of the final document. Section 8 provides a quick reference list of these commands, while this section provides a more detailed treatment of each command. Section 2 - An Overview of the Altair Word Processing System contains a description of the commands grouped by general function; the list of Formatter commands in this section is in alphabetical order.

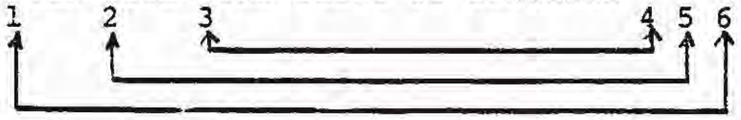
General Description _ The Formatter commands are a set of text characters that have special meaning when the text document is passed through the Formatter to be printed with the PRINT command from the text editor. The Formatter commands are commands which are embedded within the text of the document; that is, the Formatter commands are visible and can be edited in the text editor just like any other text statements. When the document is to be printed, the Formatter scans the text document to identify the Formatter commands and takes whatever operation is indicated by the command. For example, if the Formatter finds the text " @CEN Table of Contents@ " in the document, the commands @CEN and "@" will be interpreted to center the text string "Table of Contents".

The "@ " Command - Nesting of Commands - The Altair Word Processing System provides considerable flexiblity by allowing a Formatter command to be nested; that is, allowing several commands to be used together. This feature is achieved by using the "@ " character as an individual command, in addition to using it as the beginning of all other Formatter commands. Each Formatter command is actually made of two parts, the "@ " character and whatever characters immediately follow it. Therefore, "@ " is actually an "@" character followed by a space character (or by the end of a line). The set of characters immediately following the "@" character are scanned by the Formatter to be interpreted as Formatter commands. The Formatter scans until it finds a space character and interprets everything it has found to that point as part of the command. For a particular set of commands, everything to the right of the space character will be treated as the text on which the Formatter will operate, until the Formatter finds "@ " command (by itself), which is paired with the starting command. For example, in the text:

```
@CEN @BOL @UND Table of Contents@ @ @
```

three commands will operate on the set of characters "Table of Contents". The set of characters will be CENTERED, put in BOLD type and UNDERLINED. The three commmands are each followed by a space which signifies the end of the command. In addition, it should be observed that each initial command is paired to an ending "@ " command as follows:

```
@CEN @BOL @UND Table of Contents @ @ @
```



Omitting the closing "@ " command is one of the most common errors; it will usually result in the message "*** STRING OVERFLOW ***" being displayed. For example, in the text lines:

```
1.010 @BOL @CEN Table of Contents@
1.020 @SKIP3
1.030 SECTION NUMBER @TR PAGE
1.040 @SKIP2
1.050 Section 1 @TR 1
```

the "@CEN" command has an ending point, the "@ " command on the first line; however, the "@BOL" command finds no ending command in the text shown, and everything will be put in bold print.

Four Types of Formatter Commands

This example shown immediately above serves to illustrate an important aspect about Formatter commands: there are essentially four types of commands when considering what follows the command.

(1) A command that stands alone and has nothing following (such as the @ASIS command);

(2) A command that has a number as part of the command and the number is specified with the command without a space (such as the @SKIP5 command);

(3) A command which puts into effect an operation on the text which continues until it is turned-off with a paired "@ " command (such as the @BOLD command); and

(4) A command which operates like (3), but is limited to 120 or 31 characters on which it can operate until it finds the paired "@ " command (such as the @CENTER command).

In the example given above, the text "Table of Contents" is properly centered because the @CENTER command finds the paired "@ " at the end of the line, within the required 120 characters. However, @BOLD finds no ending "@ " command, but this does not create an error since you may have wanted everything in the table of contents typed in bold print. The command which requires the paired or ending "@ " to be within 120 characters are those which are generally typed within one line.

Notation Used in Describing Commands

The following symbols help describe how the Formatter commands are used. The symbols are used as a short-hand way of describing what to use with the Formatter commands to make them perform as you desire.

Symbol	Meaning
<>	The angle brackets mean that whatever is in the brackets is to be supplied by the user
[]	The square brackets mean that whatever is in the brackets is optional and is used only if the user desires
<number>	means that a number in the range 0 or .001 to 999.999 is required to be supplied by the user
<integer>	means that an integer (a number with no decimal point) between 0 and whatever limit is appropriate for the usage is supplied by the user
<digit>	means that an integer between 0 and 9 is supplied by the user
<file>	means that the name of a word processsing text documentfile, up to 8 characters in length, is supplied by the user
<drive>	means that a disk drive number, between 0 and 16, is supplied by the user. If no number is supplied, 0 is assumed
<text>	means that text is to be supplied by the user. This symbol indicates that this is a command of the third type listed above; it does not have a limitation on the amount of text which can follow until the text is ended with a "@ " command which text follows
<text-120>	means that text of no more than 120 characters (or whatever number of characters is specified) is to be supplied by the user. This text is ended with a "@ " command paired with the command which <text-120> follows.

Most of the Formatter commands use only the first three letters to specify the command.* (That is, "@CENTER", and @CEN" both would cause whatever followed the space, until the paired "@ " command was reached, to be centered.

* [There are a few exceptions to the general rule that Formatter commands are abbreviated to three characters. These exceptions are @F for the field command, @P for the paragraph command, @HM for the horizontal margin command, @IF for the "if" command, @IN for the indent both left and right command, @T and @TR for the tab and tab right commands, @V for the use of the variable, @VM for the vertical margin command and "@ " itself. When other characters are shown, such as "," or "=", they must be used as indicated.]

SECTION 10 - REDEFINING THE DEFAULT FILE FOR A USER'S INSTALLATION

The default file is a systems file used by the Altair Word Processing System which allows the user to define several characteristics of what the "standard" document will be for his installation. Each of these particular characteristics can be individually defined in a document, but if they are not defined in a particular document, the system uses the characteristics in the default file. This useful feature allows documents which are to follow the standard format defined for an installations to omit the formatting commands, but allows a different format to be easily specified.

The default file may be easily modified by selecting the Default option which is within the WP MENU (see Section 5 - Start-Up Procedures) or by executing the HELP DEFAULT command within the Text Editor. THE MASTER WORD PROCESSING DISKETTE is originally supplied with the "standard" characteristics as shown in the following default file display. The user may modify the default file according to these instructions. It is suggested that the user record the new default file specifications in Section 15 - User Installation Procedures of this Manual.

It should be noted that all of the characteristics set in the default file may be altered by Formatter commands in the text except for setting the characters (horizontal) per inch, lines (vertical) per inch, whether error messages are to be embedded in the output text (see Section 13 - Error Message and Handling), and the printer type which is set only once depending on the users particular printer.

The Set Default screen display is pictured on next page 10-2.

```
                         THE ALTAIR SYSTEM
                          WORD PROCESSING
                       SET DEFAULT PARAMETERS

IS THE DEFAULT FILE ON THE DISKETTE MOUNTED IN DRIVE 0 (Y OR N)?

*** PROGRAM TO EXAMINE/CHANGE THE DEFAULT PARAMETERS ***

*** NOTE: THE DEFAULT VALUE IS THE VALUE ASSUMED IF YOU
          HIT 'RETURN' WITH NO ENTRY FOR THAT ITEM ***

-----------------ITEM------------------   --CURRENT(DEFAULT)--   --YOUR ENTRY--
PRINTER TYPE (S; C; Q1; Q2; Q3)           Q3                     Q3
CHARACTERS PER INCH (HORIZONTAL)           10                     10
LINES PER INCH (VERTICAL)                  6                      6
PHYSICAL PAGE WIDTH (INCHES)               8.5                    8.5
PHYSICAL PAGE HEIGHT (INCHES)              11                     11
LEFT MARGIN WIDTH (INCHES)                 1.25                   1.25
RIGHT MARGIN WIDTH (INCHES)                1.25                   1.25
TOP MARGIN WIDTH (INCHES)                  1                      1
BOTTOM MARGIN WIDTH (INCHES)               1                      1
SPACING (1=SINGLE; 2=DOUBLE; 3=TRIPLE)     1                      1
JUSTIFICATION (A=ASIS; F=FILL; R=RIGHT)   A                      A
PARAGRAPH INDENTATION (CHARACTERS)         5                      5
CONTINUOUS OR BURST FORMS (C OR B)        C                      C
ERROR MESSAGES IN OUTPUT (Y OR N)         N                      N
NUMBER OF TAB SETTINGS (MAXIMUM 20)        5                      5
TAB SETTING #  1                           5                      5
TAB SETTING #  2                           10                     10
TAB SETTING #  3                           15                     15
TAB SETTING #  4                           20                     20
TAB SETTING #  5                           25                     25

END OF SET DEFAULT PARAMETERS
```

SECTION 11 - USE OF THE UTILITIES

The Altair Word Processing System contains a number of utility programs which are helpful in maintaining the various text document files on diskettes. The utilities are selected from the Utility Menu which is called when Item "Utility" is selected from the Word Processing Menu. (See Section 5). The utility programs are, for the most part, self-instructing and contain sufficient information to allow easy use. In addition, the master diskette on which Altair Disk BASIC is supplied contains a utility program (PIP) which is useful in maintenance.

Functions From the Utility Menu:

INIT

Before a new diskette may be read from or written onto by the BASIC operating system, it must be initialized. This process takes several minutes and also may be used to erase all data from old, reusable diskettes. NOTE: Caution must be taken to not accidently initialize a diskette with good program or data files.

CP1

Copy Random File (1-Disk System) This program allows for data file transfer in a single disk system from a source diskette to a destination diskette. The program transfers about 100 records at a time and provides the proper operator prompts necessary to perform the diskette swapping during the transfer.

CP2

Copy Random (and ISAM) File (2-Disk System) This program serves the same data transfer function as explained in the above COPY1 routine, except this program assumes that the user has a 2-disk system. In this case, the diskette swapping is not necessary.

CPS

Copy Sequential File. This program is used to transfer sequential files either in 1-disk or 2-disk systems. The size of the file to be copied must not be too great since the entire file is buffered in the computer's memory during the transfer.

FILEV

Diskette File Verification Routine. This program is used to check out all files on a diskette to see if all files are accessible, with no diskette surface errors.

The Utility Menu will appear as:

```
                          THE ALTAIR SYSTEM
                           WORD PROCESSING
                   UTILITY FUNCTION SELECTION MENU

                              --------

THE FOLLOWING UTILITY FUNCTIONS ARE AVAILABLE

                    INIT      DISKETTE INITIALIZATION
                    CP1       COPY RANDOM FILE (1 - DISK SYSTEMS)
                    CP2       COPY RANDOM FILE (2 - DISK SYSTEMS)
                    CPS       COPY SEQUENTIAL FILE
                    FILEV     TEST ALL FILES ON A DISKETTE
                    MENU      RETURN TO MAIN WP MENU

WHICH UTILITY WOULD YOU LIKE TO RUN?
```

A. DISKETTE INITIALIZATION

Every diskette to be utilized in the Altair Word Processing System (or in any of the other systems such as the Altair Accounting System) must be initialized before it may be used in the system. Although any high quality flexible diskette from a reputable source may be used to store document files, each diskette must be formatted and encoded ("initialized") with certain information before it may be used on the Altair computer. This process is accomplished with the initial utility program, or by using preinitialized Blank Diskettes, which may be obtained from your local Altair computer center. Any diskette initialized for the Altair computer center,such as a disk initialized for the Accounting System, may be used by the Altair Word Processing System. (It should be noted the although the files created by the Word Processing System have their own unique data structure, they can be used as data files under other programs. (See Section 12 - System Information.

CAUTION: THE MASTER COPY OF THE WORD PROCESSING SYSTEM DISKETTE HAS ALREADY BEEN INTIALIZED. DO NOT ATTEMPT TO INITIALIZE THIS DISKETTE, OR ANY OTHER DISKETTE WHICH CONTAINS USEFUL DATA OR PROGRAMS, OR ALL INFORMATION WILL BE DESTROYED. Should this unfortunate event occur, contact your local Altair computer center.

The Format Utility program is selected from the Utility menu, which was itself selected from the main WP MENU (see Section 5 - Start-Up Procedures). After INIT has been selected, a blank diskette (NOT one which contains any useful information) is loaded in the disk drive according to the instructions given on the screen by the system.

Each Utility function contains the necessary instructions and operator prompts for use.

B. FILE STATUS

Two types of file status information are available. First, the Word Processing System will provide a set of status information about word processing files by issuing, from the editor, the command ?HELP FILES, <disk drive number> which will provide a listing of all word processing files on the diskette loaded in that disk drive. In addition, comprehensive information about the current text file and the star file, if either is in use, is provided. The display produced by HELP FILES is:

(see next page)

```
? EDIT 'SAMPLE'
*** SAMPLE ON DRIVE 0 HAS BEEN OPENED FOR EDITING WITH 39 LINES
? HELP FILES

 CURRENT EDIT FILE IS [SAMPLE] FROM DRIVE 0 WITH 39 LINES
 NO STAR FILE IS OPENED

 FILES ON DRIVE 0 ARE:
SAMPLE      ADD-9/24      FORM-1       LET-9/24

?
```

C. BACK-UP PROCEDURES

Computers can greatly facilitate processing of information by business, government, science, or any facet of society. Their usefulness is a matter of record in the past thirty years. However, computers are merely machines and they do occasionally have problems. Every user has had at least one experience where a computer has "gone down" and some important information has been lost or made hard to retrieve. Altair computers, like all other computers, are not infallible and may occasionally have problems. The Altair Word Processing System has been designed in such a way that no hardware failure will be catastrophic.

As each line of text in a document is entered or modified, it is placed on the diskette for storage together with the line number for that line. Should the system fail during processing, by a power outtage or a hardware failure, the only information which may be lost would be that of a current line being entered or modified. The system has the capability to rebuild the special indexes used by word processing solely from information on the diskette.

The only problem then that can cause a user more than just minor inconvenience is one that causes the loss of information on a diskette. These problems usually result from improper handling of the fragile diskettes by touching the recording surface or exposing to heat. Upon rare occasion, a disk drive itself may cause some information on a diskette to be lost. The more usual occurrences will be for a user to inadvertently initialize a diskette containing useful data, thereby destroying the data. (See Diskette Initialization above.) The loss caused by these problems can be minimized by the implementation of a set of back-up procedures (these procedures should be entered in Section 16 - User Installation Procedures). The exact procedures needed would vary greatly between installations; however, the following general rules provide a guideline:

1. On all working diskettes used, make a back-up copy of the diskette each evening. The back-up copy made three days prior may be erased and the diskette re-entered into the blanks.

2. On all diskettes which contain master text of documents, make one back-up copy of the diskette at any time the text master is altered or added to. This extra master is kept at the installation.

3. On all master text diskettes, make a back-up copy of the diskette each Friday. Place this copy in a safe off-premises. The back-up copy of that master copy made four Fridays back may be retrieved to be erased.

4. A user may make a couple of copies of the MASTER WORD PROCESSING diskette for back-up purposes. Each Altair Disk BASIC diskette comes with a utility copy program called "PIP", with which the user (assuming he has two or more disk drives) may copy an entire diskette. (Note that any copying or use of the software which is not authorized by the Altair Software Distribution Company may be in violation of federal and state law.) Place one back-up in the off-premises safe for storage.

THIS PAGE LEFT INTENTIONALLY BLANK

SECTION 12 - SYSTEM INFORMATION -- INTERFACING OTHER SOFTWARE

(NOTE: The information in this Section is provided for the convenience of users who desire to interface other software with the Altair Word Processing System or to obtain a more technical understanding of the operation of the system. THIS SECTION IS NOT NECESSARY FOR THE USER unless the user desires this special information.)

The Altair Word Processing System uses the same disk file structure that is employed by disk files created the Altair Disk BASIC and Altair DOS computer languages. This fact provides one of the important advantages to the Altair Word Processing System compared with other word processing machines. The Altair Word Processing System can be easily interfaced with data files created by other computer software. This feature means that a user can have computer programs which perform specialized functions in his business, such as accounting, inventory record keeping, laboratory measurements, data collection, or engineering computation, and direct the output to be used in the Altair Word Processing System. In this way, a user can greatly facilitate routine report writing, bid submittals, real estate closings or other text-oriented functions. Programs operating on Altair computers can output data in random disk files; this data can be used as text by the Altair Word Processing System if the output files for those programs follow the simple guidelines of this section. Information on using Altair Disk BASIC and Altair DOS to create data files are contained in the separately published manuals for those systems. Local Altair computer centers can advise users on securing assistance in writing or interfacing other programs.

The Altair Word Processing System utilizes a file structure available in Altair BASIC or DOS known as random files. This means that a user has access to any particular record in a disk file by knowing merely the name of the file and the record number. The files are "random" because the records may be accessed in any order, they need not be located sequentially. The records used by Altair BASIC and DOS are 128 characters in length; that is, the records will store 128 alphanumeric characters. The files are identified on a particular diskette by an eight-character name. The Altair Word Processing System uses one computer file for each word processing document file and stores three types of information in each computer file:

(1) a set of identification information known as "header record" or a "label record" for the file;

(2) the text itself, stored as one line of text per one computer record; and

(3) the "index", a complex set of indexing information which allows the Altair Word Processing System to efficiently access the line of text.

A word processing text document file stored on a diskette for later use has each of these three items complete and ready to function when the file is called by an EDIT, OPEN, or STAR command. However, the word processing system will handle files which have completed only the first two items -- the header record and the text records. The Word Processing System has the capability to build (or re-build) an index from information contained in the record for each line, a feature which greatly facilitates interfacing other software programs to word processing since the indexing information is extremely complex. This rebuilding is being done whenever the message *** THE FILE [name] WAS NOT CLOSED PROPERLY. ONE MOMENT FOR CORRFCTION *** appears on the video screen

Any data files created by Altair BASIC or DOS as a random file can be used as input to word processing if the guidelines below are followed. The file may be given any name except a name beginning with a '$' character or a '#' character since these two characters identify files used internally in word processing. The first record (record number one) of the file initially contains the header record information as follows:

CHARACTER POSITION	LENGTH	DESCRIPTION
1 - 4	4	00000000(HEX)
5 - 10	6	#EDIT# (ASCII CHARACTER
11 - 16	6	00000200FFFF (HEX)

Character positions 17 through 128 are used by the indexing portions of the file and should be filled with binary zeros. The records from record number two forward are used to store the text, one record for each "line" of text as follows:

CHARACTER POSITION	LENGTH	DESCRIPTION
1	1	Length of text line in bytes*
2	1	FF (HEX)-designates a text line
3 - 4	2	0000(HEX)-reserved for internal use
5 - 8	4	Line number (byte)**
9 - 128	120	Text line

*The length of the text line in bytes is expressed as decimal representation of an ASCII character.

**A line number is a 4 byte floating point number having the integer value 1 to 999999. For example, line number 1.000 is represented by the floating point number 1000; while line number 526.520 is represented by 526520.

For example, the text:

```
1.000 This is line one.
2.000 This is line two!
```

would be represented in the word processing system as:

BASIC RECORD

```
    1      00000000  #EDIT#00000200FFFF  (header record)
    2      IIFF0000  [4 byte F.P. number 1000] This is line one.
    3      IIFF0000  [4 byte F.P. number 2000] This is line two!
```

The user is reminded that the limit in word processing is 500 lines for a single document; but documents may be "linked". (See Section 10 - The Formatter Commands for a detailed description of the "LINK" command.) The computer file will have a few more records than lines, because the header information and indexing information are stored in the same file.

THIS PAGE LEFT INTENTIONALLY BLANK

SECTION 13 - ERROR MESSAGES AND HANDLING

The Altair Word Processing System is designed to "trap" as many user-caused errors as possible and report them to the user. The user may obtain a more complete explanation of the problem by typing the Editor HELP commands (just an "H" or a "?" will suffice for HELP). (See Section 7 for a detailed description of the editor commands.)

Errors which are detected in the Formatter are reported either on the video screen or printed in the typed document at the point the error occurred. The user selects the preferred option through the default file settings. (See Section 10 for redefining the default file.) There may be times when the user prefers to have the text printed without an error message in the form:

```
*** (Error Message) ***
```

printed in the middle of the document in the hopes that the document can be used regardless of the error. In this case, the error message is merely put on the video screen. However, the more usual procedure is to have the error message embedded in the typed document to insure that it will be corrected, particularly where the printer will be left unattended.

The most common formatter error made by users tend to be the ommission of the closing '@' command from the text commands which are required to be closed. (See Section 8 or Section 9 for a list of those commands.) When this error is made, the system will generally respond with the message:

```
*** STRING OVERFLOW IN LINE ***
```

The Altair Word Processing System reports errors which may occur in the use of the Altair Disk BASIC language. An explanation of these errors is found in the Altair BASIC Disk Manual. There errors, other than STRING OVERFLOW described above, are generally caused by hardware problems. Your local Altair computer center can give guidance.

THIS PAGE LEFT INTENTIONALLY BLANK

SECTION 14 - APPLICATIONS AND EXAMPLES

The Altair Word Processing System is a powerful and flexible text editing system which can fill a wide variety of needs. The scope of the system is very broad and users have many varied applications for the system. This section is presented to provide the user with several examples of word processing documents. The applications shown by these documents are typical of a variety of firms. Local Altair computer centers will be able to advise users on the use of the system in unusual applications. The examples shown in this section are:

EXAMPLE I -	a sample session with the text editor
EXAMPLE II -	selected portions of simple source text
EXAMPLE III -	repetitive letters
EXAMPLE IV -	a "programmed" form

A. GENERAL USE

Example I below shows a short session using the text editor to initialize the editor,recall the file named "SAMPLE", and make certain changes in the file. The entire file "SAMPLE" is listed as Example II (a) and II(b). This session was begun by the user, after "bringing-up" BASIC, running "WP MENU" according to the instructions of Section 5 - Start-Up Procedures. The first two questions asked of the user will set @DATE and @NAME for this session of word processing. When the word processing session is ended with the END command, control is returned to the Word Processing Menu for running other programs.

```
                          The Altair Word Processing System
                                    Copyright 1977
                                         ASDC

WHAT IS THE DATE TO BE USED IN THE PRINTOUT? September 28, 1977
WHAT IS THE NAME TO BE USED IN THE PRINTOUT? Jan Lawrence

** TEXT EDITOR READY **

? EDIT 'SAMPLE'
*** SAMPLE ON DRIVE 0 HAS BEEN OPENED FOR EDITING WITH 40 LINES
? HELP FILES

  CURRENT EDIT FILE IS <SAMPLE> FROM DRIVE 0 WITH 40 LINES
  NO STAR FILE IS OPENED

  FILES ON DRIVE 0 ARE:
SAMPLE        ADD-9/24      FORM-1        LET-9/24

? LN 1-3
  1.000 @page
  1.500 @cen @bol SAMPLE@ @ @skip
  2.000 @just
  3.000 @P
? D 1.5
*** 1  LINES DELETED
? END
*** SAMPLE HAS BEEN CLOSED, THERE ARE 39 LINES IN THE FILE
*** WORD PROCESSING FINISHED ***
```

Example Exercise. The Edit file named SAMPLE is opened, followed by the request for file status on drive 0 via the HELP FILES command. The lines 1 through 3 are then listed on the screen with line numbers, and then line number 1.5, the title line of the document is deleted via the D 1.5 command. SAMPLE is then closed via the END command.

Example I

```
                  FILE SAMPLE  ON September 25, 1977  ( 39 LINES )

  1.000 @page
  2.000 @just
  3.000 @P
  4.000      How does a small business fully realize the cost-effectiveness
  5.000 of a microcomputer?  One major consideration is the right software.
  6.000 And the right office software is now available at all Altair Computer
  7.000 Centers.  It's the @BOL Altair Business System@ --a total office
  8.000 management software package for all Altair 8800-based systems.
  9.000
 10.000 @in10
 11.000 @P
 12.000      The @BOL Altair Business System@  brings accounting, word processin
 13.000 and inventory management software into your office at a surprisingly
 14.000 low cost.
 15.000
 16.000 @in0
 17.000 @P
 18.000 The @bol accounting@  packages include a flexible general ledger package
 19.000 that allows a business to keep a timely, detailed general ledger of
 20.000 all transactions and provide up-to-date financial reports for
 21.000 management.  Packages for receivables, payables, and payroll functions
 22.000 keep subsidiary ledgers current and produce automatic invoicing
 23.000 statements, check writing, 941s, W-2s, and other required reports.
 24.000 The @bol word processing@  system is an extremely flexible text editor
 25.000 that allows large volume text material, such as contracts, to be easily
 26.000 modified and printed.  In addition, documents can call for inserts
 27.000 from other files, making repetitive letters very easy to produce.
 28.000 The @bol inventory management@  system is an extremely flexible data base
 29.000 that allows complete inventory records to be kept on line.  The
 30.000 package is designed to permit custom structuring of inventory files
 31.000 and reports.
 32.000
 33.000 @in5
 34.000 @P
 35.000     Do something good for your own business.  Contact your local Altair
 36.000 Computer Center or the Altair Software Distribution Company directly
 37.000 for more information about the .... @skip2
 38.000 @asis @cen @bol @cap Altair Business System@ @ @
 39.000 @PAGE
```

Example II(a)

How does a small business fully realize the cost-effectiveness of a microcomputer? One major consideration is the right software. And the right office software is now available at all Altair Computer Centers. It's the **Altair Business System**--a total office management software package for all Altair 8800-based systems.

The **Altair Business System** brings accounting, word processing and inventory management software into your office at a surprisingly low cost.

The **accounting** packages include a flexible general ledger package that allows a business to keep a timely, detailed general ledger of all transactions and provide up-to-date financial reports for management. Packages for receivables, payables, and payroll functions keep subsidiary ledgers current and produce automatic invoicing statements, check writing, 941s, W-2s, and other required reports. The **word processing** system is an extremely flexible text editor that allows large volume text material, such as contracts, to be easily modified and printed. In addition, documents can call for inserts from other files, making repetitive letters very easy to produce. The **inventory management** system is an extremely flexible data base that allows complete inventory records to be kept on line. The package is designed to permit custom structuring of inventory files and reports.

Do something good for your own business. Contact your local Altair Computer Center or the Altair Software Distribution Company directly for more information about the ....

ALTAIR BUSINESS SYSTEM

Example II(b)

B. REPETITIVE LETTERS

One of the most useful applications of the Altair Word Processing System is in producing repetitive letters or similiar documents for lists for names. The Altair Word Processing System contains a number of commands which make these applications very flexible. Examples III(a) through III(d) below show how the letter file and name files are constructed.

Repetitive letter applications generally use the set of formatter commands that tranfer control from a primary document (usually the letter) to subsidiary files containing records divided into fields (usually names and addresses). To accomplish this, the primary document (the current file) must open the particular subsidiary file (a star file) with the @OPEN command. The primary document references each successive record in the star file with the @GET<number> command. It is noted that @GET44 on line 8.000 of the file named LET-9/24, shown in Example III(b), causes the control to transfer to line 44.000, in the current file when the @EOF is reached in the star file. The star file is divided first into records indicated by the @REC command. Each record is further divided into fields which correspond to the relative line location within each record. In the file named ADD-9/24, shown in Example III(a), lines 3 and 8 are each FIELD 1 for their particular record (the first line in the record), while lines 4 and 9 are each FIELD 2 for their record. The reference to a particular field is made in the primary document (the current file) with @F1 for Field 1, @F2 for Field 2, etc. It should be noted that any command may be used in the subsidiary document (the star file). The results of running Example III(b) as the edit file and Example III(a) as the star file results in the two letters Examples III(c) and III(d).

```
              FILE ADD-9/24  ON September 25, 1977  ( 12 LINES )

 1.000 @REM This is the name and address file called ADDRESS@
 2.000 @REC
 3.000 Mr. Smith
 4.000 Mr. John D. Smith
 5.000 3632 Lincoln Lane
 6.000 Washington, D. C.  20003
 7.000 @REC
 8.000 Bob
 9.000 Mr. Robert T. Jones
10.000 17 Fairway
11.000 Augusta, Georgia  30447
12.000 @EOF
```

Example III(a)

FILE LET-9/24 ON September 25, 1977 (46 LINES)

```
 1.000 @REM This is an example of a Word Processing document@
 2.000 @TSET5,30
 3.000 @SET @V1=@QUERY What is name of person sending the letter(salutation)@
 4.000 @SET @V2=@QUERY What are the initials of this person (upper case)@ @
 5.000 @set @v3=@query What is this person's title@ @
 6.000 @SET @V4=@QUERY What are your initials (lower case)@ @
 7.000 @OPEN ADD-9/24@
 8.000 @GET44
 9.000 @PAGE
10.000 @SKIP3
11.000 @cen Harris Supply Company@
12.000 @cen 3330 Peachtree Street, N.E.@
13.000 @cen Atlanta, Georgia 30329@
14.000 @skip3
15.000 @t  @t @date
16.000 @SKIP4
17.000 @ASIS
18.000 @F2
19.000 @F3
20.000 @F4
21.000 @SKIP1
22.000 Dear @F1:
23.000 @FILL @P Thank you for the opportunity to demonstrate the Altair Word
24.000 Processing System.  This text is stored in a file called LET-9/24 while
25.000 the inserts (the names and addresses) are stored in a document file
26.000 named ADD-9/24.
27.000 @P This letter is an example of using text from two different files to
28.000 produce a single document when the final is typed, in this case a
29.000 set of letters.  These letters merge the addresses of the document file
30.000 named ADD-9/24 with the master letter in the document file named
31.000 LET-9/24 to produce the final letters.
32.000 @P I am enclosing a brochure which illustrates other useful features
32.100 of the system.  Please let me know if I can provide any additional
32.200 information.
33.000 With every good wish, I am
34.000 @SKIP3 @ASIS
35.000 @t  @t Sincerely,
36.000 @SKIP4
37.000 @t  @t @V1
38.000 @t  @t @V3
39.000
40.000 Enclosure
41.000
42.000 @v2/@v4
43.000 @GOTO8
44.000 @END
```

Example III(b)

Harris Supply Company
3330 Peachtree Street, N.E.
Atlanta, Georgia 30329

September 25, 1977

Mr. John D. Smith
3632 Lincoln Lane
Washington, D. C. 20003

Dear Mr. Smith:

Thank you for the opportunity to demonstrate the Altair Word Processing System. This text is stored in a file called LET-9/24 while the inserts (the names and addresses) are stored in a document file named ADD-9/24.

This letter is an example of using text from two different files to produce a single document when the final is typed, in this case a set of letters. These letters merge the addresses of the document file named ADD-9/24 with the master letter in the document file named LET-9/24 to produce the final letters.

I am enclosing a brochure which illustrates other useful features of the system. Please let me know if I can provide any additional information. With every good wish, I am

Sincerely,

Ronald D. Robertson
Sales Manager

Enclosure

RDR /jbh

Example III(c)

Harris Supply Company
3330 Peachtree Street, N.E.
Atlanta, Georgia 30329

September 25, 1977

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

Dear Bob:

Thank you for the opportunity to demonstrate the Altair Word Processing System. This text is stored in a file called LET-9/24 while the inserts (the names and addresses) are stored in a document file named ADD-9/24.

This letter is an example of using text from two different files to produce a single document when the final is typed, in this case a set of letters. These letters merge the addresses of the document file named ADD-9/24 with the master letter in the document file named LET-9/24 to produce the final letters.

I am enclosing a brochure which illustrates other useful features of the system. Please let me know if I can provide any additional information. With every good wish, I am

Sincerely,

Ronald D. Robertson
Sales Manager

Enclosure

RDR /jbh

Example III(d)

C. A TEXT "FORM"

The Altair Word Processing System can be used to "program" a form; that is, it can be used to make inquiries of the user and type the information in the form.

This capability of word processing to help prepare forms can be very useful, particularly with small forms. However, the user should recognize that there are several limitations in using word processing for this application for longer, more complex forms. Word processing has only ten variables (@V0 through @V9) to use; if a form required more spaces than this, a variable could be used more than once, but this has certain disadvantages. The user may find it more appropriate to have the inputting of information for a form or calculations or field checking done in Altair BASIC and the output formatted by word processing for printing. (See Section 12 - System Information - Interfacing Other Software.)

The document file named "FORM-1", listed as Example IV(a) contains the form with QUERY statements which collect the information needed to "fill in the blanks." The finished form is shown as Example IV(b). Note that the form uses the logical test command @IF to control typing. Lines 37 and 38, for example, allow a user to specify that there are more invoices to prepare by typing either upper or lower case Y (or y).

```
                 FILE FORM-1  ON September 28, 1977  ( 44 LINES )

 1.000 @rem this is a sample form called FORM-1@
 1.500 @HM.5
 2.000 @TSET8,45
 3.000 @asis
 4.000 @page
 5.000 @cen @und INVOICE@ @
 6.000
 7.000 @set @v1=@query Please type invoice number@ @
 8.000 @skip2
 9.000 Harris Supply Company                  Invoice Date| @DATE
10.000 3330 Peachtree Street
11.000 Atlanta, Georgia 30329
12.000
13.000 Sold to|
14.000
15.000 @t @query Name of party sold to@
16.000
17.000 @t @query Address line one@
18.000
19.000 @t @query Address line two@
20.000
21.000 @t @query Address line three@
22.000
23.000
24.000 -----------------------------------------------------------------------
25.000     Description                             Amount
26.000     -----------                             -------
27.000
28.000 @set @v2=@query Type description (END! to end)@ @
28.500 @IF @v2 @EQU END!  @GOTO31
28.800 @SET @v3=@query Type amount@ @
28.900 @v2 @t @t @v3
30.000 @goto28
31.000                                        -----------------------
32.000          TOTAL                            $ @Query Total Amount@
32.500
33.000                                        =======================
33.500
34.000 Please reference Invoice Number @v1 in your remittance.  Thank you.
35.000 @PAGE
36.000 @set @v4=@query Any more invoices to prepare (Y, N)@ @
37.000 @IF @v4 @con Y@goto5
38.000 @IF @v4 @con y@goto5
39.000 @END
```

Example IV(a)

```
                                INVOICE

Harris Supply Company                       Invoice Date| September 28, 1977
3330 Peachtree Street
Atlanta, Georgia 30329

Sold to|

       Miller Office Equipment
       North Dekalb Center
       1324 Druid Hills Road
       Decatur, Georgia 30319

----------------------------------------------------------------------
    Description                                  Amount
    -----------                                  -------

1000 9x12 Envelopes                              23.00
Typewriter Stand, Model 4T                       51.50
2 File Cabinets, 3-drawer, Oliv                  97.30
                                           -----------------------
         TOTAL                                 $ 171.80
                                           =======================

Please reference Invoice Number I-1334C  in your remittance.  Thank you.
```

Example IV(b)

THIS PAGE LEFT INTENTIONALLY BLANK

SECTION 15 - USER INSTALLATION PROCEDURES

This section of the Manual is provided to allow a convenient place for you to insert procedures unique to your installation. In addition, it is suggested that the user default file specifications be noted for reference on the forms when the default file is modified.

	Orig.		
DATE MODIFIED			
CHARACTERS PER INCH?	10		
LINES PER INCH?	6		
PHYSICAL PAGE WIDTH AND HEIGHT (INCHES)?	8.5,11		
WIDTH OF THE LEST AND RIGHT MARGINS (INCHES)?	1,1		
WIDTH OF THE TOP AND BOTTOM MARGINS (INCHES)?	1.5,1		
SPACING DEFAULT (1-SINGLE,2-DOUBLE,3-TRIPLE)?	1		
JUSTIFICATION DEFAULT (A-ASIS, F-FILL, R-RIGHT)?	A		
INDENTATION FOR PARAGRAPHS (CHARACTERS)?	5		
CONTINUOUS OR BURST (C,B)?	C		
ERROR MESSAGES BE EMBEDDED IN THE OUTPUT (Y,N)?	N		
NUMBER OF TAB SETTINGS?	5		
TAB SETTING 1?	5		
TAB SETTING 2?	10		
TAB SETTING 3?	15		
TAB SETTING 4?	20		
TAB SETTING 5?	25		
TAB SETTING 6?			
TAB SETTING 7?			
TAB SETTING 8?			
TAB SETTING 9?			
TAB SETTING 10?			
TAB SETTING 11?			
TAB SETTING 12?			
TAB SETTING 13?			
TAB SETTING 14?			
TAB SETTING 15?			
TAB SETTING 16?			
TAB SETTING 17?			
TAB SETTING 18?			
TAB SETTING 19?			
TAB SETTING 20?			