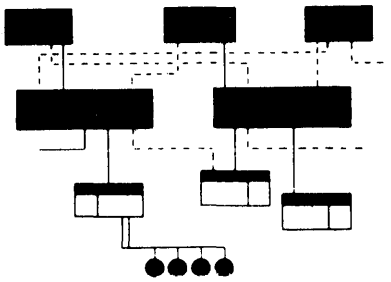
Betty J. Colonca

GE-625/635 Time-Sharing System

Information Systems

Information Systems Equipment

TEXT EDITOR



GENERAL ELECTRIC

CPB-1515

CPB-1515

GE-625/635 Time-Sharing System

TEXT EDITOR

PROGRAM NUMBER

CD600T2.001

March 1968

INFORMATION SYSTEMS

GENERAL ELECTRIC

CPB-1515

PREFACE

This is a user-oriented manual designed for either the beginner or the experienced programmer.

A user who is familiar with terminal operation and communication with the GE-625/635 Time-Sharing System can skip Chapters 1 and 2.

A functional description of each edit verb is given in Chapter 3 in addition to the operand field. Then Chapter 4 describes how each EDITOR command is used.

The appendixes supplement the chapters: Appendix A summarizes all forms of each EDITOR command; Appendix B describes the GECOS File System and the 625/635 Time-Sharing ACCESS Subsystem; Appendix C describes the HELP Subsystem which explains error messages; Appendix D contains Octal/ASCII Conversion Equivalents and Definitions; Appendix E shows the GE-625/635 Standard Character Set.

This manual is one in a series of time-sharing manuals. The others are:

GE-625/635 Time-Sharing System BASIC Language, CPB-1510

GE-625/635 Time-Sharing System Programming Reference Manual, CPB-1514

GE-625/635 Time-Sharing System File System, CPB-1513

This manual was produced using the General Electric Remote Access Editing System (RAES). RAES is a time-shared disc-resident storage and retrieval system with text-editing and manuscript formatting capabilities. The contents of the manual were entered into RAES from a remote terminal keyboard, edited using the system editing language, and formatted by RAES on reproduction masters.

The index was produced using a computer-assisted remote access indexing system. This sytem produces an index using source strings delimited at manuscript input time.

Suggestions and criticisms relative to form, content, use of this manual are invited. Comments may be sent on the

Document Review Sheet in the back of the manual or may be addressed directly to Documentation Standards and Publications, B-90, Processor Equipment Department, General Electric Company, 13430 North Black Canyon Highway, Phoenix, Arizona 85029.

CONTENTS

APPENDIXES

1. INTRODUCTION

EDITOR is a text-editing subsystem of the GE-625/635 Time-Sharing System (600TSS) which allows the user to build a body of text, edit it, save and retrieve it, all from a remote-terminal keyboard. The text may be of any description: letters, lists, documentation, business records, or programs for other 600TSS systems.

While EDITOR was conceived primarily to save the tedious and time-consuming retyping involved in drafting letters and memoranda, its range of application is limited only by the ingenuity of its users. For instance, it is useful in the production and adaptation of form letters, and in the maintenance of ordered lists which must often be amended: mailing lists, inventories, and schedules. Items in a letter or list may easily be inserted, deleted, or modified without disturbing the surrounding text.

Another feature of EDITOR is its simplicity: it is easy to use, and easy to learn. Its command language consists primarily of five editing verbs: REPLACE, DELETE, INSERT, PRINT, and FIND. Five other verbs perform auxiliary functions, e.g., BACKUP, SAVE.

The bodies of text produced by EDITOR are referred to generically as files. This name reflects the fact that EDITOR files have the same characteristics as do the files of other 600TSS systems, such as BASIC. In fact, files can be passed from one system to another.

Generally, operation of EDITOR is as follows:

> The user logs into the Time-Sharing System by dialing in and giving his user-identification and password. He then specifies EDITOR in response to the question SYSTEM? At this point a choice is offered (OLD or NEW?) of building a new file (NEW) or retrieving one that was previously saved (OLD).

EDITOR is now ready to receive input, either to build the new file or add on to the old. This is called "build-mode," and is indicated by an asterisk marking the beginning of each new line. The asterisk does not become part of the file. It merely informs the user that the EDITOR subsystem is ready for the next line of input in build-mode.

In build-mode, EDITOR recognizes only one "command" as distinct from text: a carriage-return in the first printing position, i.e., immediately following the initial asterisk. This command causes EDITOR to change to "direct-mode," in which all of the standard EDITOR commands are effective. The user is then able to perform any necessary editing of the old or new file.

The major part of this manual is devoted to a description of the EDITOR commands and examples of their use.

2. HOW TO OPERATE THE 600 TSS TERMINAL

STANDARD OPERATIONAL CONTROLS

The GE-625/635 Time-Sharing System is easy to use because the Teletype keyboard is similar to a typewriter. The present terminal device is either a Model 33 or Model 35 Teletype unit. The following paragraphs describe all of the Teletype controls that are used by the Time-Sharing System.

Model 33 Teletype Unit

The principal parts of the Model 33 Teletype unit are the Control Unit, Keyboard, Paper Tape Punch (optional), and Paper Tape Reader (optional).

Control Unit

Rotary Dial	-for dialing telephone numbers.
ORIG (Originate)	-button is depressed to obtain a dial tone.
CLR (Clear)	-button is depressed to turn off Teletype unit.
LCL (Local)	-button is depressed to perform local off-line work such as tape punching. This mode cannot connect with the computer.
BUZ-RLS (Break Release)	-button is depressed to free keyboard after a "break" signal. (Signal indicates keyboard is locked.)
NORMAL-RESTORE	-should always point to NORMAL.

Loudspeaker	-usually located under the keyboard at the right. A volume control knob permits volume adjusting of the loudspeaker. If button is turned as far as possible counterclockwise no dial tone will be audible.

Keyboard

The keyboard operates like a standard typewriter with the following exceptions. (Special keys on the keyboard which are not used by the Time-Sharing System will not be discussed.)

Letters of the alphabet are printed in capital letters only - there are no small letters.

The shift key is nonlocking and must be held depressed when typing.

RETURN - returns the carriage to the left margin.

LINE FEED - moves the paper up one line at a time.

REPT (Repeat) - to repeat the same character this key is held depressed while the desired character key is operated. The latter is released and the REPT key is held until the desired number of characters have been typed. (If the character is upper case, the shift key must be held along with the REPT key.)

CTRL - when this key is depressed simultaneously with the X key, the line currently being typed is deleted.

Model 35 Teletype Unit

The principal parts of the Model 35 Teletype unit are the Control Unit, Keyboard, Paper Tape Punch, and Paper Tape Reader.

Control Unit

Rotary or Touch Dial	-for dialing telephone numbers.
ORIG (Originate)	-button is depressed to obtain a dial tone.
CLR (Clear)	-button is depressed to turn off Teletype unit.
LCL (Local)	-button is depressed to perform local off-line work such as tape punching. This mode cannot connect with the computer.
BUZ-RLS (Buzzer Release)	-buzzer sounds when paper supply is low. Key is depressed to silence buzzer.
BRK-RLS (Break Release)	-button is depressed to free keyboard after a "break" signal. (Signal indicates keyboard is locked.)
NORMAL-RESTORE	-should always point to NORMAL.
Loudspeaker	-usually located under the keyboard at the right. A volume control knob permits adjusting of the volume of the loudspeaker. If button is turned as far as possible counterclockwise no dial tone is audible.

The six lights on the upper right side of the unit are not used by the Time-Sharing System (see Teletype Instruction Manual).

DIAL, BY, INCPT, NO CON, SVC, PA.

The Column Indicator located at the upper-right side of the keyboard indicates which column has just been printed, typed, or punched.

The End-of-Line red light indicates that an end-of-line is being approached; it has no effect on computer or Teletype unit.

The control buttons located to the left of the keyboard, all equipped with lights, are not used by the Time-Sharing System except for the following:

BREAK	-causes the unit to discontinue a printing operation. (See BREAK in index for further information.)
K (Keyboard)	-for obtaining page copy and transmission to the system.

When originating a call, the terminal is automatically switched to the Tape (T) mode. The user must depress the K button before any transmission can be effected, and he must depress K to release the keyboard after a break.

Keyboard

The keyboard operates like a standard typewriter with the following exceptions. (Special keys on the keyboard which are not used by the Time-Sharing System will not be discussed.)

Letters of the alphabet are printed in capital letters only - there are no small letters on the 33 or 35 teletypewriters.

The shift key is nonlocking and must be held depressed when typing characters in upper case positions.

RETURN - returns the carriage to the left margin.

LINE FEED - moves the paper up one line at a time.

REPT (Repeat) - to repeat the same character this key is held depressed while the desired character key is operated. The latter is released and the REPT key is held until the desired number of characters have been typed. (If the character is upper case, the shift key must be held along with the REPT key.)

CTRL - when the CTRL key is depressed simultaneously with the X key, the line currently being typed is deleted.

Four red keys, two on each side of the space bar, are used for local action only. They do not generate any code on a paper tape or transmit a code to the computer. From left to right, they are:

LOC B.SP.	-not used
LOC LF	-spaces the carriage up one line
LOC CR	-returns the carriage to the left margin
REPT	-explained previously

Special Controls

There are three Teletype control keys that have special functions for 600TSS terminal operation. They are:

Key	Function When Depressed
RE TURN	Returns the carriage and simultaneously transmits the content of the line to the system. In this manual, the use of this key is indicated, where necessary by the symbol (CR).
CTRL and X	In combination, deletes whatever has been typed on the current line and returns the carriage.
@ P	The upper case of this key, @ (using the shift key) causes the deletion of the character immediately preceding it on the line. Also +, if the key is depressed any number of times consecutive characters immediately preceding the first @ will be deleted from the line.

In addition, there is a BREAK control button on the console which permits the user to discontinue the printing of a file and, when the BRK-RLS button is depressed, has EDITOR revert to a request for the selection of another subsystem or EDITOR again. This feature is especially convenient when an error has occurred and a long output is expected. After the use of BREAK, the BRK-RLS button must be depressed to enable further operation.

Upper and Lower Case Alphabetic Characters

Upper and lower case alphabetic characters are accepted by the EDITOR subsystem from any remote terminal device capable of sending and receiving both sets of character codes, e.g., the Model 37 Teletype. However, the Models 33 and 35 units transmit only upper-case alphabetic codes to the GE-625/635 Time-Sharing System. Although the 33 and 35 may receive either upper-case or lower-case code, they print only the corresponding upper-case character.

COMMUNICATING WITH THE TIME-SHARING SYSTEM

This section provides the operational procedures for using the terminal, together with the system responses.

Call-In Procedure

To establish communications with 600TSS:

- remove the Dataphone from the cradle
- depress the ORIG button on the console and wait for a dial tone
- upon receipt of the dial tone, dial the assigned 600TSS number

When the connection is made, a high-pitched tone occurs and the log-on procedure begins.

- replace the Dataphone

Log-on Procedure

When a connection has been established, the terminal will receive:

THIS IS THE GE-600 TSS SYSTEM ON date AT time

USER ID--the user-ID is typed in followed by (CR)

PASSWORD--type password over double strike-overs and (CR)

SYSTEM? EDITOR is typed by the terminal user and (CR)

OLD OR NEW--type either OLD or NEW, depending on which is desired, followed by (CR)

OLD NAME--this only occurs when an OLD file has been selected; type file name followed by (CR)

READY FOR INPUT

* text may now be entered (CR)

The system indicates its readiness for a line of text by printing an asterisk in the first position of each line, following a carriage return by the user.

The user-ID is a string of characters assigned to identify a user so that the system can identify his files and keep account of his user charges.

The PASSWORD is a unique identification which protects the user against unauthorized use of his user-ID. The double strike-overs serve to make his entry illegible to others.

Following SYSTEM?, name the subsystem to be used; in this case it is the EDITOR subsystem.

OLD OR NEW permits either the retrieval of a file that has been saved or the building of a new file. OLD NAME must be followed by a file name when the OLD option is chosen. For further information on saved files, and how to use files that have been stored differently, see Appendix B.

Building or Adding to a File

Following the message READY FOR INPUT and the initial asterisk, the EDITOR subsystem is now in the build mode and the user may type in lines of text, either to build a new file or add to the end of an old one. For each line of text, at least one character or blank is typed before a carriage return is given. The system then responds with another initial asterisk and another line of text can be entered.

When the user is finished building, and he wants to edit or save the file, he gives a carriage-return immediately following the initial asterisk. This places EDITOR in direct mode, indicated simply by the word READY. In direct mode, the EDITOR command language is recognized (described in Chapter 3).

#TAPE Usage

Another method of building a new file or adding to the end of an old one is with the use of paper tape. Following the message READY FOR INPUT and the initial asterisk, the user may type #TAPE, which instructs the system to accept paper tape instead of keyboard input. In response to this, the message READY is generated at the terminal. The user must then place his paper tape input in the terminal reader, depress the tape mode button (KT or T) and turn on the reader by depressing the "TD ON" button. Input from this tape is accepted until either the user stops the reader by depressing the "TO OFF" button, the tape runs out, the tape jams in the reader, or an X-OFF character is met on the tape. When the tape input is finished, the system responds with an asterisk, at which time the user can continue to create or update the file or give a carriage return to place EDITOR in direct mode.

Log-Off Procedure

When a user indicates that he is finished with EDITOR, using either the SAVE or DONE command (described in Chapter 3), he arrives again at the question SYSTEM? . He then responds LOGOFF and receives the following terminating information:

**RESOURCES USED $ amount, USED TO DATE $ amount = percentage

**TIME SHARING OFF AT time ON date

The terminal is then disconnected from the system.

Terminal Disconnections

A terminal is automatically disconnected for any of the following reasons:

- An invalid user-ID or password is typed twice during the log-on request.
- The terminal is allowed to remain idle for ten minutes.
- The CLEAR control button is depressed.

Error Messages and the HELP Subsystem

During operation of 600TSS, certain usage errors produce error messages prefixed by an error-code number. These messages may also be caused by error conditons that are beyond the control of the terminal user. While all 600TSS error messages are intended to be as self-explanatory as possible, they also must of necessity be brief. Therefore, the 600TSS HELP subsystem is provided to fully explain any numbered error message, whenever such is desired. Instructions for using HELP are given in Appendix C.

3. EDITOR COMMAND LANGUAGE

An edit command is composed of an edit verb and, usually, an operand field. The verb indicates the type of operation desired, and the operand field supplies any necessary particulars, although it may be null for a given operation.

EDIT VERBS

There are 10 edit verbs:

PRINT	BACKUP
INSERT	STRING
REPLACE	LINE
DELETE	SAVE
FIND	DONE

Each of the five verbs in the left column may be employed in several different ways, as shown in the section, "Operand Field." The remaining five verbs have a fixed meaning and, except for SAVE, take no operand field. A functional description of each edit verb follows.

PRINT -indicates that the file, or a selected portion of the file, is to be printed. The operand field of the PRINT command is used to specify the portions of the file to be printed.

INSERT -allows the user to insert any number of characters, words, or lines into his file. The operand field of the INSERT command specifies the point after which the insertion is to be made. The system responds to an INSERT command with the word ENTER: and the text to be inserted is then typed.

REPLACE -allows the user to replace portions of his file with new text. It is possible to replace any number of characters, words, or lines. The new text need not correspond in length with the text to be replaced. The operand field of the REPLACE command specifies the text to be replaced. The system responds to the REPLACE command with the word ENTER: and the replacement text is then

typed. (In certain cases, multiple replacements can be made with one REPLACE command.)

DELETE -allows the user to delete a portion of his file. The operand field of the DELETE command specifies the text to be deleted. (In certain cases, multiple deletions of identical portions of the file can be deleted with one DELETE command.)

Use of each of the four edit verbs described above results in a character-by-character search through the text to locate that portion to be printed, replaced, or deleted, or the point at which an insertion is to be made. The starting point for this search is controlled by a search pointer that is associated with each file.

The search pointer initially points to the first line of the file, and always moves forward, i.e., toward the end of the file during a search operation.

FIND -provides the user with a means of moving the search pointer forward. It is most often used to pre-position the search pointer to a given line for a subsequent operation, thus saving search time when editing a lengthy file. It also may be used to ensure the proper comparison in case of multiple occurrences of the same character string in the file.

The five edit verbs (PRINT, INSERT, REPLACE, DELETE, FIND) have two distinct modes of operation: line-mode and string-mode (described in Chapter 4), with line-mode being the normal operating mode for EDITOR.

BACKUP -allows the user to return the search pointer to the beginning of the file. It is used when it becomes necessary to operate on a portion of the file already passed by the search pointer. BACKUP is the only edit verb that causes a backward movement of the search pointer.

STRING -causes subsequent edit commands to operate in string-mode, which for a single command can be specified in the operand field of that command.

LINE -returns EDITOR to line-mode operation.

SAVE -has two functions: (1) it causes the edited file to be saved under the file name specified in the operand field, i.e., stored by the system so that it can be retrieved at a later time, and (2) causes an exit from the EDITOR subsystem. The question SYSTEM? is reissued following execution of a SAVE command. At this point the user may select EDITOR again or another system.

DONE -causes an exit from the EDITOR subsystem, without saving the current file. When working with an OLD file (one previously saved), DONE allows the user to retain the original file unaltered by the modifications effected during the editing session. This is particularly advantageous if the file should be altered accidentally.

OPERAND FIELD

The general form of an edit command is:

verb (operand field)

The command can consist of only the verb. However, if the operand field is present, it must have a string specifier or a repetition factor. The verbs BACKUP, STRING, LINE, and DONE do not take any operand field, and the SAVE command always requires a file name in the operand field.

The operand field for the PRINT, INSERT, REPLACE, DELETE, and FIND commands can contain one or more of the following items:

- A mode indication, string (S) or line (L).
- A character-string field, preceded by a colon.
- A repetition factor (*), preceded by a semicolon.

If more than one of these items is employed in one command, they must appear in a given order. The form, with the individual optional specifications in parentheses, is:

verb (S) (: string field) (; repetition factor)

Both the string field and the repetition factor may be used independently of each other and of the string-mode indication, but S is not meaningful without a following string field.

The meaning and use of the string-mode indication is described in Chapter 4, "Line-Mode versus String-Mode."

The String-Field

The string-field must always be preceded immediately by a colon. The string field itself consists of a string of characters preceded and followed by a delimiting character. The user chooses a delimiter for each string field; it may be any character, except blank or @, that is not contained in the enclosed character string. The character string in turn may contain any keyboard character, including blanks, excepting of course the delimiter chosen. The commercial-at-sign character may be used in the character string, but its effect is to erase the immediately preceding character, as described in Chapter 2, "Operational Controls." The general form of the string field is:

```
:dxxx...xd
```

where d is the delimiter and x is any character other than d.

Examples of simple string fields are:

```
(1)  :/THE COMMITTEE/
(2)  :. WHEN THE VALUE OF (F/X)*N IS 0,.
(3)  :XMR. PERRY, MR. SMITH, AND MISS JONES X
```

The character-string delimiter in example (1) is the slant, in example (2) the period, and in example (3) the character X. In all succeeding examples, the slant will be used as the delimiter unless it conflicts with a slant in the string, as in item (2).

Carriage returns may not be given within an operand string-field before the terminating delimiter. However, there is a form of string-field that is used following the sytem response ENTER. This response occurs after REPLACE and INSERT commands (see Chapter 4). This form lacks the preceding colon, and carriage returns may be used within the delimiters, as is sometimes necessary.

When the character string to be identified and acted upon by a command is a lengthy one, but is either less than or more than one whole line of the file, an alternate form of the string field may be used that permits a considerable economy of typing. This form consists actually of two string fields separated by a comma, the first field identifying the beginning of the character string and the second identifying the end of the string. This is called a point-to-point string-field:

:/initial characters/,/terminal characters/

For example, the character string identified by

:/STRING FIELDS ARE ENCLOSED BY DELIMITERS./

could also be identified, point-to-point, by

:/STRING/,/ITERS./

This form is recognized only in string mode.

The user need specify only as many beginning and ending characters as are required to uniquely identify the desired string. For instance, the fields /STR/ and /ERS./ would be sufficient to identify the complete example sentence, and would probably be sufficient to identify only that sentence in a small file.

The Repetition Factor

The repetition factor, when numeric, specifies the number of times that the operation implied by the command is to be executed. The effect of this repetition varies, depending upon the nature of the operation.

The repetition factor must be immediately preceded by a semicolon and it may be a decimal number, an asterisk, or blank.

;n where n is any whole number other than zero. The operation is executed n times.

;* where the * indicates the operation is repeated until end-of-file.

; where a blank indicates the operation is executed once. This is equivalent to a factor of 1.

The effect of the repetition factor is explained in the detailed descriptions of each command (see Chapter 4). However, some examples are given here; assume that the search pointer is positioned at the beginning of the file:

The command PRINT ;5 prints the first five lines of the file.

The command PRINT :/YOU/;3 prints the first three lines that begin with the character-string YOU.

The command PRINT;* prints the complete file.

The command FIND :/YOU/;5 positions the search pointer at, the fifth line that begins with the character-string YOU.

The command FIND ;10 advances the search pointer 10 lines.

The command DELETE S:/YOU/;5 deletes the first five occurrences of the character-string YOU and not the containing lines, regardless of where that string appears in a line. (Notice that string-mode is specified. Without the S specification, the first five lines that begin with YOU, YOUR, YOU'RE, etc., would be deleted.)

The command DELETE ;3 would delete the first three lines of the file.

The description of the effect in each of the cases above is based upon the assumption that the search pointer is located at the beginning of the file. If the pointer were located elsewhere within the file, the result would be analogous: the same operation would begin with the line at which the pointer is located. For example, the last command, DELETE ;3, would delete any three consecutive lines, always starting from the one at which the pointer currently pointed.

4. HOW TO USE THE EDITOR COMMANDS

This chapter is divided into two parts:

(1) A statement of general usage rules

(2) A detailed description of the use of each command

GENERAL USAGE RULES

Several general characteristics of EDITOR's operation and of EDITOR files must be remembered when using the edit commands. These concern the search pointer, line-mode versus string-mode differences, the effect of blanks and other nonprinting characters on searching operations, and the first line of EDITOR files.

Operation of the Search Pointer

As stated in Chapter 3, each file that enters the EDITOR subsystem has a search pointer associated with it. When EDITOR is placed in direct-mode, after being in the build-mode, the pointer is automatically located at the first line of the file. It always points at the beginning of a line, and never at an intermediate point within a line. This means that if an edit operation is performed on a character-string within a given line, the search pointer still points to the beginning of that line, and the following command can also operate on that same line.

The rules governing the movement of the search pointer are as follows:

(1) The PRINT, INSERT, REPLACE, DELETE, and FIND commands all cause the search pointer to move forward towards the end-of-file, unless the command affects only the line at which the pointer is already located (usually a command with no operand field).

(2) Following the execution of any of the commands listed in rule (1), the pointer is located only at the line affected by the command.

(3) The BACKUP command moves the search pointer to the beginning of the file from wherever it is located.

(4) For commands involving a search operation--a string field is specified--the file is always searched starting at the search pointer; the search is terminated either by a successful comparison with the specified string field or by encountering the end-of-file. In the latter case, the pointer must be backed-up prior to any further editing operations.

These rules imply that if a given line has already been passed by the search pointer, the BACKUP command must be used before that line can be operated on by a subsequent command.

The current position of the search pointer can always be determined by using a PRINT command with no operand field.

Line-Mode Versus String-Mode

Line-mode is the normal mode of operation for EDITOR unless string-mode is explicitly specified by the user. The mode determines both the type of operation to be performed and how the string field is interpreted.

In line-mode, a command acts upon one or more complete lines of the file. That is, one or more complete lines are inserted, replaced, deleted, or printed.

Of equal importance, the search for a successful comparison to the string field (if any) is limited to the initial characters of each line. This implies that in line-mode, the characters specified in the string field must correspond to those at the beginning of a line, always starting with the first printing position. (The word "character" in this manual is always meant to include blanks and other nonprinting characters. See searching conventions concerning nonprinting characters in the Index for further information.) In the character-by-character comparison process involved in searching, the first character of the string field is compared with the first character of a line. If these initial characters are not the same, the search proceeds directly to the first character of the next line, rather than to succeeding characters in the same line.

For insertions and replacements, the appropriate carriage returns will be automatically supplied by EDITOR in line-mode.

In string-mode, a command can act upon any string of characters in the file, regardless of line breaks (carriage returns). The character string acted upon can range from one or more characters within a line, to any number of consecutive characters extending through several lines. For example, a complete sentence that begins in the middle of one line and ends within another line could be deleted without disturbing the rest of the first and last lines.

As in string-mode, the character string specified in the string field need not be unique to the beginning of a line. The character string may represent the beginning of a line, of course, but the whole line will not be implied, as in line mode -- only the string actually specified.

Carriage returns are not automatically supplied in string-mode, and when using the INSERT and REPLACE commands the user must be careful to give the necessary carriage returns within the ENTER string-field delimiters.

Searching Conventions

The following conventions concerning the effect of blanks, carriage returns, and other nonprinting characters are used by EDITOR in searching:

(a) Carriage returns and any other nonprinting characters in the file, excepting blanks, are ignored for comparison purposes.

(b) Consecutive blanks in the file need not be matched exactly by the blanks in the operand string-field: $\underline{n}$ consecutive blanks in the string field means at least $\underline{n}$ consecutive blanks in the corresponding position of the character string in the file. This does not apply, however, where $\underline{n} = 0$.

These conventions imply the following rules for specifying operand string-fields.

(1) Carriage returns and other nonprinting characters in the file, excepting blanks, should not be matched in the string field.

(2) One or more consecutive blanks in the file need be matched by only one blank in the string field.

(3) N consecutive blanks in the file may not be matched by more than n blanks in the string field.

The rules do not apply to insertion or replacement string fields that follow the EDITOR request "ENTER"; see INSERT and REPLACE in Chapter 4.

The searching conventions and the consequent rules must be remembered when specifying string fields for searching, to successfully locate the desired portion of the file.

The Null Line

All files, when operated on by the EDITOR subsystem, contain a null first line. What would normally be considered the first line of the file is effectively the second line for the purpose of EDITOR operations. This permits insertions to be made at the beginning of the file. If such a line insertion is made, a new null first line is automatically supplied.

The null line must be considered by the user when the search pointer is located at the beginning of the file. For instance, if the user desires to print the first four "actual" lines of the file, he must request that the first five be printed, i.e., PRINT ;5. The null line and the succeeding four nonempty lines will be printed. Similarly, when printing of the first actual line of the file is desired, the user must specify PRINT ;2 rather than simply PRINT.

EDITOR COMMAND STRUCTURE AND USAGE

The following paragraphs contain descriptions of the structure and usage of each EDITOR command.

PRINT Command

The PRINT command is used when either a selected portion of a file or the entire file is to be printed. The user can vary the PRINT command to print any one of the following:

- The entire file
- Any number of consecutive lines
- Any number of lines containing a given character string
- From one point to another
- A single line

Print the Entire File

To print the entire file, the user must give the PRINT command in line mode, and specify the repetition factor to indicate the entire file. (No string field is specified.) Printing begins at the location of the search pointer and continues until the end of file.

Example:

```
PRINT  ;*

DEAR MR.MRS.

PERHAPS YOU HAVE BEEN OUT OF TOWN OR MISPLACED OUR
STATEMENT DATED**. AS YOU KNOW, WE TOO HAVE CREDITORS
WHO MUST BE PAID. WON'T YOU KINDLY SUBMIT A CHECK IN
THE AMOUNT OF ## TO CLEAR UP THIS SMALL MATTER.

IN CASE YOU HAVE ALREADY TAKEN CARE OF THIS, PLEASE
DISREGARD THIS NOTE.

END OF FILE
```

Print Any Number of Consecutive Lines

To print any number of consecutive lines, beginning at the location of the search pointer, a PRINT command with a repetition factor of n is given, in line mode.

Example:

```
BACKUP
READY
PRINT ;4

DEAR MR.MRS.

PERHAPS YOU HAVE BEEN OUT OF TOWN OR MISPLACED OUR
```

Note the null line following the PRINT command, which accounts for one of the four lines specified.

Print Any Number of Lines Containing a Given Character String

If a repetition factor is used with a string field, printing lines containing the specified character string continues as instructed by the repetition factor. Searching begins at the location of the search pointer.

Example:

```
PRINT  S:/YOU/;2

PERHAPS YOU HAVE BEEN OUT OF TOWN OR MISPLACED OUR
STATEMENT DATED **. AS YOU KNOW, WE TOO HAVE CREDITORS

READY

PRINT  S:/YOU/;*

PERHAPS YOU HAVE BEEN OUT OF TOWN OR MISPLACED OUR
STATEMENT DATED**. AS YOU KNOW, WE TOO HAVE CREDITORS
WHO MUST BE PAID. WON'T YOU KINDLY SUBMIT A CHECK IN THE
IN CASE YOU HAVE ALREADY TAKEN CARE OF THIS, PLEASE

END OF FILE
```

Print from One Point to Another

Printing from one point to another requires a point-to-point string field. The first string field contains a data comparison to a point in the line at which printing is to begin; the second string field contains data unique to a point in the line at which printing is to terminate. String-mode must be used.

Example:

```
BACKUP
READY
PRINT  S:/CREDITORS/,/CLEAR/
STATEMENT DATED**. AS YOU KNOW, WE TOO HAVE CREDITORS
WHO MUST BE PAID. WON'T YOU KINDLY SUBMIT A CHECK IN THE
AMOUNT OF ## TO CLEAR UP THIS SMALL MATTER.

READY
```

Print a Single Line

When the PRINT command operand field is omitted, the line pointed to by the search pointer is printed.

Example:

```
PRINT
AMOUNT OF ## TO CLEAR UP THIS SMALL MATTER.
READY
```

When only a string field is specified in the PRINT command, the line identified by the string field is printed. The string field must contain a word or string of characters unique to the beginning of the line to be printed.

Example:

```
BACKUP
READY
PRINT  :/WHO/
WHO MUST BE PAID. WON'T YOU KINDLY SUBMIT A CHECK IN THE

READY
```

STRING Command

The STRING command places EDITOR in string-mode. All subsequent commands are executed in string-mode until a LINE command is given. The STRING command never takes an operand field. Use of STRING is equivalent to using the S specification in the operand field of each subsequent command.

An example using both STRING and LINE is given in the second example for the FIND command (described later in this chapter).

LINE Command

The LINE command counteracts the effect of STRING, placing EDITOR back into line-mode, its normal mode of operation.

BACKUP Command

The BACKUP command moves the search pointer back to the beginning of the file, i.e., to the null line. It is the only command that causes a backward movement of the search pointer. It must be used whenever the user wishes to operate on a portion of the file already passed by the pointer. BACKUP takes no operand field.

The use of BACKUP is amply illustrated in the examples given for other edit commands.

FIND Command

With the FIND command, the user can advance the search pointer without causing any other operation to take place. The command can be used in either line-mode or string-mode.

Example:

```
PRINT  ;*

YOUR ACCOUNT IS NOW NINETY DAYS OVERDUE. UNLESS
WE RECEIVE YOUR CHECK IN THE AMOUNT OF '' BEFORE
%%, WE SHALL BE FORCED TO TURN THIS MATTER OVER
TO OUR ATTORNEY FOR COLLECTION.

IF THERE IS ANY REASON WHY YOU THINK THIS BILL
SHOULD NOT BE GIVEN TO OUR LEGAL OFFICE, PLEASE
CONTACT ME AT CL2-7197.

END OF FILE
BACKUP
READY
FIND  S:/''/
READY
PRINT
WE RECEIVE YOUR CHECK IN THE AMOUNT OF '' BEFORE

READY
```

The repetition factor may be used with a string field in the FIND command. The search and comparison continues until the comparison is made as many times as indicated. When execution is completed, the response READY appears. If the repetition factor is used without a string field, the search pointer moves forward n number of lines as indicated by the factor.

Example:

```
READY
STRING

READY
FIND  :/CHECK/
READY
PRINT
WE RECEIVE YOUR CHECK IN THE AMOUNT OF '' BEFORE

READY
FIND  :/WE/;2
READY
PRINT
%%, WE SHALL BE FORCED TO TURN THIS MATTER OVER

READY
BACKUP
READY
LINE
READY
FIND  ;5
READY
PRINT
TO OUR ATTORNEY FOR COLLECTION.

READY
```

When editing a lengthy file, it is advantageous to move the search pointer, in line mode, to the beginning of the portion of the file to be edited, thus saving search time.

If the user is ever in doubt as to where the search pointer is located, he should give the PRINT command, with no operand field. The resulting printout is the line pointed to by the search pointer.

REPLACE Command

Replace a String of Characters

The string field in the REPLACE command must contain a copy of the text to be replaced. Following a successful search for this text, EDITOR responds with ENTER. The replacement text is typed and must also be enclosed by delimiters, as in an operand string-field, but the preceding colon must not be given. When the REPLACE command has been executed, the computer responds with READY.

Note that the replacement-text string must be an exact representation of what is to be entered into the file, including all desired blanks. Also, the replacement string need not match in length the string to be replaced. Replacement begins at the first character position specified in the operand string-field. String-mode must be specified. A repetition factor may be specified. If so n identical replacements are performed (unless end-of-file is encountered first).

Example:

```
PRINT  ;*
PLEASE INCLUDE IN YOUR NEXT ORDER THE FOLLOWING ITEMS:

STOCK NO.     QTY.    STOCK NO.     QTY.    STOCK NO.    QTY.

 AX-4         10/EA     AB-4         2/C       DA-9       7/EA
 AB-1         12/EA     AD-7        33/EA      DD-9       9/EA

WE HAVE BACKORDERED THE FOLLOWING ITEMS WHICH HAVE NOT BEEN
RECEIVED:

STOCK NO.     QTY.

 YT-0         47/EA
 GF-67        15/C**

PLEASE EXPEDITE WITH FIRST-CLASS DELIVERY

END OF FILE
BACKUP
READY
```

```
REPLACE  S:/AB-4/
ENTER
/AC-4/
READY
REPLACE  S:.33.
ENTER
/ 3/
READY
REPLACE  S: /FIRST-CLASS/
ENTER
/PRIORITY/
READY
BACKUP
READY
PRINT  ;*

PLEASE INCLUDE IN YOUR NEXT ORDER THE FOLLOWING ITEMS:

STOCK NO.    QTY.   STOCK NO.    QTY.   STOCK NO.   QTY.

 AX-1        10/EA    AC-4        2/C      DA-9      7/EA
 AB-1        12/EA    AD-7        3/EA     DD-9      9/EA

WE HAVE BACKORDERED THE FOLLOWING ITEMS WHICH HAVE NOT BEEN
RECEIVED:

STOCK NO.    QTY.

 YT-0        47/EA
 GF/67       15/C**

PLEASE EXPEDITE WITH PRIORITY DELIVERY

END OF FILE
```

Replace a Line

When the user wishes to replace a complete line of the file, the line must be indicated in the string field by text unique to the beginning of that line. When EDITOR responds with ENTER, the user types in the new line, enclosed by appropriate delimiters. If a repetition factor is given, the first n lines that begin with the same specified characters are replaced, if n such lines exist in the file.

If the operand field is omitted entirely, the line at which the search pointer is positioned is replaced by the new line.

Example:

```
PRINT  ;*

A PROMOTIONAL SALE INTRODUCING A NEW LINE OF GARDEN
TOOLS WILL BE HELD THE WEEK FOLLOWING LABOR DAY.
NEWSPAPER AND RADIO ADVERTISEMENTS WILL BEGIN THE
FIRST OF SEPTEMBER AND CONTINUE FOR THE ENTIRETY
OF THE SALE.

WE ARE SUBMITTING A REQUEST TO HIRE FOUR TEMPORARY
SALES CLERKS FOR THIS PERIOD. ANY APPLICANTS WILL
BE INTERVIEWED BY MR. SMITH, MANAGER - HARDWARE.

END OF FILE
BACKUP
READY
REPLACE  :/FIRST/
ENTER
/LAST WEEK IN AUGUST AND CONTINUE FOR THE ENTIRETY/
READY
REPLACE  :/BE/
ENTER
/BE INTERVIEWED BY R. JAMES, HARDWARE DEPARTMENT./
READY
BACKUP
READY
PRINT  ;*

A PROMOTIONAL SALE INTRODUCING A NEW LINE OF GARDEN
TOOLS WILL BE HELD THE WEEK FOLLOWING LABOR DAY.
NEWSPAPER AND RADIO ADVERTISEMENTS WILL BEGIN THE
LAST WEEK IN AUGUST AND CONTINUE FOR THE ENTIRETY
OF THE SALE.

WE ARE SUBMITTING A REQUEST TO HIRE FOUR TEMPORARY
SALES CLERKS FOR THIS PERIOD. ANY APPLICANTS WILL
BE INTERVIEWED BY R. JAMES, HARDWARE DEPARTMENT.

END OF FILE
```

Replace from Point-to-Point

If a point-to-point string field is specified, all data between and including the two points indicated is replaced. String-mode must be specified. A repetition factor may be used to obtain successive identical replacements.

Example:

```
PRINT  ;*

NOTICE TO ALL EMPLOYEES:

THE FOLLOWING POLICIES WILL GO INTO EFFECT IMMEDIATELY

1.  LUNCH HOURS SHOULD BE TAKEN BETWEEN THE HOURS OF
    11:30 AND 1:00, IF POSSIBLE. THIS DOES NOT APPLY
    TO THE SHIPPING DEPARTMENT.

2.  VACATIONS SHOULD BE TAKEN BEFORE SEPTEMBER 1.
    EMPLOYEES IN THE BOOKKEEPING DEPARTMENT SHOULD PLAN
    TO TAKE THEIR VACATIONS THE SECOND AND THIRD WEEKS
    OF THE MONTH IN ORDER TO BE PRESENT FOR THE MONTHLY
    ACCOUNTS CLOSING.

3.  ANY DEVIATIONS FROM THE ABOVE SHOULD BE CLEARED WITH
    MRS. JACKSON IN PERSONNEL.

END OF FILE
REPLACE  S:/11:30/,/BLE./
ENTER
/NOON AND ONE O'CLOCK./
READY
REPLACE S:/3. /,/PERSONNEL./
ENTER
/3.  MRS. JACKSON IN PERSONNEL SHOULD BE CONSULTED
     REGARDING ANY VARIATIONS IN THE ABOVE./

READY
BACKUP
READY
PRINT  ;*
```

NOTICE TO ALL EMPLOYEES:

THE FOLLOWING POLICIES WILL GO INTO EFFECT IMMEDIATELY.

1. LUNCH HOURS SHOULD BE TAKEN BETWEEN THE HOURS OF NOON AND ONE O'CLOCK. THIS DOES NOT APPLY TO THE SHIPPING DEPARTMENT.

2. VACATIONS SHOULD BE TAKEN BEFORE SEPTEMBER 1. EMPLOYEES IN THE BOOKKEEPING DEPARTMENT SHOULD PLAN TO TAKE THEIR VACATIONS THE SECOND AND THIRD WEEKS OF THE MONTH IN ORDER TO BE PRESENT FOR THE MONTHLY ACCOUNTS CLOSING.

3. MRS. JACKSON IN PERSONNEL SHOULD BE CONSULTED REGARDING ANY VARIATIONS IN THE ABOVE.

END OF FILE

DELETE Command

To Delete a String of Characters

To delete a string of characters, the user specifies string mode in the DELETE command and gives the text to be deleted in the string field. If a repetition factor is given, successive instances of the same string are deleted as indicated by the factor.

Example:

```
PRINT  ;*

A NEW DEPARTMENT HAS BEEN ESTABLISHED TO COORDINATE
ALL WINDOW AND FLOOR DISPLAYS. MR. JOHN R. PERRY
HAS BEEN SELECTED TO DIRECT THIS FUNCTION.
MR. PERRY WAS FORMERLY IN SALES PROMOTION.
HIS DUTIES WILL ALSO INCLUDE THE SELECTION OF
PERMANENT INTERIOR DECORATIONS, AS WELL AS THOSE
OF A SEASONAL NATURE. THIS DEPARTMENT WILL ALSO HAVE
RESPONSIBILITY FOR BUYING AND STOCKING GREETING CARDS.
STATIONERY, AND ASSOCIATED ITEMS. THIS WAS FORMERLY
UNDER THE NOTIONS DEPARTMENT.

END OF FILE
BACKUP
READY
DELETE  S: /MR./
READY
DELETE  S: /ALSO/
READY
BACKUP
READY
PRINT  ;*

A NEW DEPARTMENT HAS BEEN ESTABLISHED TO COORDINATE
ALL WINDOW AND FLOOR DISPLAYS. JOHN R. PERRY
HAS BEEN SELECTED TO DIRECT THIS FUNCTION.
MR. PERRY WAS FORMERLY IN SALES PROMOTION.
HIS DUTIES WILL INCLUDE THE SELECTION OF
PERMANENT INTERIOR DECORATIONS, AS WELL AS THOSE
OF A SEASONAL NATURE. THIS DEPARTMENT WILL ALSO HAVE
RESPONSIBILITY FOR BUYING AND STOCKING GREETING CARDS,
STATIONERY, AND ASSOCIATED ITEMS. THIS WAS FORMERLY
UNDER THE NOTIONS DEPARTMENT.

END OF FILE
```

Delete One or More Lines

A single line may be deleted by including a word or string of characters unique to the beginning of that line in the string field of the DELETE command. If the DELETE command is used without an operand field the line at which the search pointer sits will be deleted.

If a repetition factor is given in addition to a string field, successive lines containing the same beginning characters are deleted. If only a repetition factor is specified in the operand, the number of consecutive lines indicated by the factor are deleted, starting from the search pointer location.

Example:

```
DELETE :/MR. PERRY/
READY
BACKUP
READY
PRINT ;*

A NEW DEPARTMENT HAS BEEN ESTABLISHED TO COORDINATE
ALL WINDOW AND FLOOR DISPLAYS. JOHN R. PERRY
HAS BEEN SELECTED TO DIRECT THIS FUNCTION.
HIS DUTIES WILL INCLUDE THE SELECTION OF
PERMANENT INTERIOR DECORATIONS, AS WELL AS THOSE
OF A SEASONAL NATURE. THIS DEPARTMENT WILL ALSO HAVE
RESPONSIBILITY FOR BUYING AND STOCKING GREETING CARDS,
STATIONERY, AND ASSOCIATED ITEMS. THIS WAS FORMERLY
UNDER THE NOTIONS DEPARTMENT.

END OF FILE
```

Delete from Point-to-Point

To delete text in the file from one point to another, the DELETE command must indicate in the point-to-point string field the extent of the desired deletion. String-mode must be used. A repetition factor may be specified for successive identical deletions.

Example:

```
DELETE S:/BUYING/,/CKING/
READY
BACKUP
READY
PRINT ;*

A NEW DEPARTMENT HAS BEEN ESTABLISHED TO COORDINATE
ALL WINDOW AND FLOOR DISPLAYS. JOHN R. PERRY
HAS BEEN SELECTED TO DIRECT THIS FUNCTION.

HIS DUTIES WILL INCLUDE THE SELECTION OF
PERMANENT INTERIOR DECORATIONS, AS WELL AS THOSE
OF A SEASONAL NATURE. THIS DEPARTMENT WILL ALSO HAVE
RESPONSIBILITY FOR GREETING CARDS,
STATIONERY, AND ASSOCIATED ITEMS. THIS WAS FORMERLY
UNDER THE NOTIONS DEPARTMENT.

END OF FILE
```

INSERT Command

Insert One or More Lines

To insert a line into a file, use the INSERT command with a string field identifying the line that precedes the desired insertion point. When the INSERT command is given, EDITOR responds with ENTER. The user must then enter the new line(s), enclosed by delimiters. For multi-line insertions, each new line except the last must contain a carriage return to prevent it from running into the line following.

Example:

```
PRINT ;*

THE FOLLOWING ITEMS WILL BE OFFERED AT 10% DISCOUNT 9/1/67.

STOCK NO.     DEPARTMENT

45634127      HARDWARE
80076873      HARDWARE

65644653      LINEN

78566435      PATIO

END OF FILE
BACKUP
READY
INSERT :/THE FOLL/
ENTER
/THIS APPLIES TO CASH PURCHASES ONLY./
READY
INSERT :/656/
ENTER
/23456787      LINEN/
READY
BACKUP
READY
PRINT ;*

THE FOLLOWING ITEMS WILL BE OFFERED AT 10% DISCOUNT 9/1/67.
THIS APPLIES TO CASH PURCHASES ONLY.

STOCK NO.     DEPARTMENT
45634127      HARDWARE
80076873      HARDWARE

65644653      LINEN
23456787      LINEN

78566435      PATIO

END OF FILE
```

Insert a String of Characters

A string of characters of any length may be inserted into the file by use of the INSERT command. The operand string field must indicate the point after which the new text will be inserted. The length of the character string to be inserted must be considered. Long character strings may run off the right-hand margin of the paper unless a carriage return is included, prior to the closing delimiter, in the ENTER string-field. String-mode is specified.

Example:

```
PRINT ;*

MEMO TO CREDIT DEPARTMENT                ACCOUNT #

A CREDIT FOR $ HAS BEEN ISSUED TO
FOR THE RETURN OF ONE.

END OF FILE
BACKUP
READY
INSERT S:/#/
ENTER
/A54421/
READY
INSERT S:/$/
ENTER
/8.32/
READY
INSERT S:/ISSUED TO/

ENTER
/ A.K. HENRY/
READY
INSERT S:/ONE/
ENTER
/BEDSPREAD/
READY
BACKUP
READY
PRINT ;*

MEMO TO CREDIT DEPARTMENT                ACCOUNT #A53421

A CREDIT FOR $8.32 HAS BEEN  ISSUED  TO A.K. HENRY
FOR THE RETURN OF ONE BEDSPREAD.

END OF FILE
```

SAVE Command

The SAVE command allows the user to save the current file for future use. The form of this command is always the same:

SAVE filename

where filename is a 1 to 8 character name identifying the saved file. The name may be composed of any combination of alphabetic and numeric characters, plus the period and the dash.

The SAVE command causes the file that is currently being edited to be stored, as a permanent (OLD) file, under the specified file name. The file can then be retrieved at a later time by responding with OLD to the question OLD OR NEW, and giving the file name in response to OLD NAME--.

The SAVE command also causes an exit from the EDITOR subsystem; following this command the question SYSTEM? is reissued. (EDITOR may be selected again, of course.)

The use of SAVE must be considered in two separate cases:

(1) when a file is originally built, and

(2) when an OLD (previously saved) file has been retrieved and modified.

When a file is built -- and then possibly edited -- the SAVE command must be used if the user wishes to retain this file. Otherwise, if the user leaves the EDITOR system via the DONE command the file disappears. Where an OLD file has been retrieved, a copy of the specified file is made available for editing. The previously saved original remains undisturbed, as a permanent file, no matter what modifications are made to the copy, unless the edited copy in turn is saved in place of the original, i.e., under the same filename. If the copy, with its modifications, is not saved, it disappears at the end of the user's session at the terminal, and the original file remains for later retrieval.

Alternatively, if the user wishes to retain both the original OLD file and the modified copy, he may save the latter under some other file name. Clearly the user must choose unique file names for each file that he wishes to retain. Each file name will represent only one permanent file for a given user; each such file will contain only what has been last saved under that file name.

If a user wishes to retrieve files saved by another user, or has some other unusual requirements concerning file storage and retrieval, refer to Appendix B.

DONE Command

The DONE command allows the user to leave the EDITOR system without saving the file currently being edited. The system-selection question, SYSTEM?, is issued. The user then has the option of selecting another system (possibly EDITOR again), or of responding with LOGOFF or NEWUSER to obtain his resources-used information. LOGOFF causes the terminal to be disconnected. NEWUSER allows another user to commence terminal operations while it is still active and connected to the Time-Sharing System.

APPENDIX A

Summary of EDITOR Command Structure

In this summary, the following conventions are used:

xxx	- represents a string of characters, of any length, that matches the beginning of one or more lines in the file.
yyy and zzz	- represent strings of characters, of any length, that do not necessarily match the beginning of a line in the file.
n	- represents a decimal number.
/	- slants are used as delimiter characters; any other appropriate character might be used instead.

Command	Execution
PRINT	Print one line.
PRINT ;n	Print n consecutive lines.
PRINT ;*	Print entire file.
PRINT :/xxx/	Print line identified by xxx.
PRINT :/xxx/;n	Print n successive lines identified by xxx. (* may be used instead of n to print all such lines.)

PRINT S:/yyy/	Print line containing specified string.
PRINT S:/yyy/;n	Print n successive lines containing specified string. (* may be used instead of n to print all such lines.)
PRINT S:/yyy/,/zzz/	Print from line containing string yyy to line containing string zzz, inclusive. (A repetition factor may be used with this form also.)
FIND	Advance search pointer one line.
FIND ;n	Advance search pointer n lines.
FIND :/xxx/	Find line identified by xxx.
FIND :/xxx/;n	Find nth line identified by xxx.
FIND S:/yyy/	Find line containing specified string.
FIND S:/yyy/;n	Find nth line containing specified string.
REPLACE	Replace line at which search pointer is currently located.
REPLACE :/xxx/	Replace line identified by xxx.
REPLACE :/xxx/;n	Replace n successive lines identified by xxx. (* may be used instead of n to replace all such lines.)

REPLACE S:/yyy/	Replace specified string.
REPLACE S:/yyy/;n	Replace n successive occurrences of the specified string. (* may be used instead of n to replace all such occurrences.)
REPLACE S:/yyy/,/zzz/	Replace text between points yyy and zzz, inclusive. (A repetition factor may be used with this form also.)
DELETE	Delete line at which search pointer is currently located.
DELETE :/xxx/	Delete line identified by xxx.
DELETE :/xxx/;n	Delete n successive lines identified by xxx. (* may be used instead of n to delete all such lines.)
DELETE S:/yyy/	Delete specified string.
DELETE S:/yyy/;n	Delete n occurrences of specified string. (* may be used instead of n to delete all such occurrences.)
DELETE S:/yyy/,/zzz/	Delete text between points yyy and zzz, inclusive. (A repetition factor may be used with this form also.)
INSERT	Insert after the line at which the search pointer is currently located.
INSERT :/xxx/	Insert after the line identified by xxx.

INSERT :/xxx/;n	Insert after each of n successive lines identified by xxx. (* may be used instead of n to insert after all such lines.)
INSERT S:/yyy/	Insert after point yyy.
INSERT S:/yyy/;n	Insert after each of n successive occurrences of point yyy. (* may be used instead of n to insert after all such occurrences.)
BACKUP	Return search pointer to the beginning of the file.
STRING	Place EDITOR into string-mode.
LINE	Return EDITOR to line-mode.
SAVE filename	Save edited file under specified name (8 characters or less), and exit from EDITOR.
DONE	Exit from EDITOR without saving edited file. Original (OLD) file remains unaltered.

An abbreviated verb form may be used in the command format with the exception of the termination verbs SAVE and DONE.

For example:

P;*	performs exactly as PRINT ;*
F:/xxx/	performs exactly as FIND :/xxx/
DS:/xxx/;3	performs exactly as DELETE S:/xxx/;3
I	performs exactly as INSERT
B	returns search pointer to the beginning of the file.

APPENDIX B

THE GECOS FILE SYSTEM AND ACCESS

This appendix describes the function and use of the 600TSS ACCESS subsystem in relationship to the GECOS File System. If the user has referenced files only with the SAVE and OLD commands, he does not have to be concerned with ACCESS.

THE GECOS FILE SYSTEM AND ACCESS

The GE-600 Time-Sharing System utilizes the capabilities of the GECOS file system which is a logical mechanism for storing and retrieving permanent files and is common to all programs operating under the GE-600 Comprehensive Operating Supervisor. Since a file system can store many files on some external, "background" storage device, the user need not be concerned with the device his file is on nor the characteristics of the device.

Structure of the File System

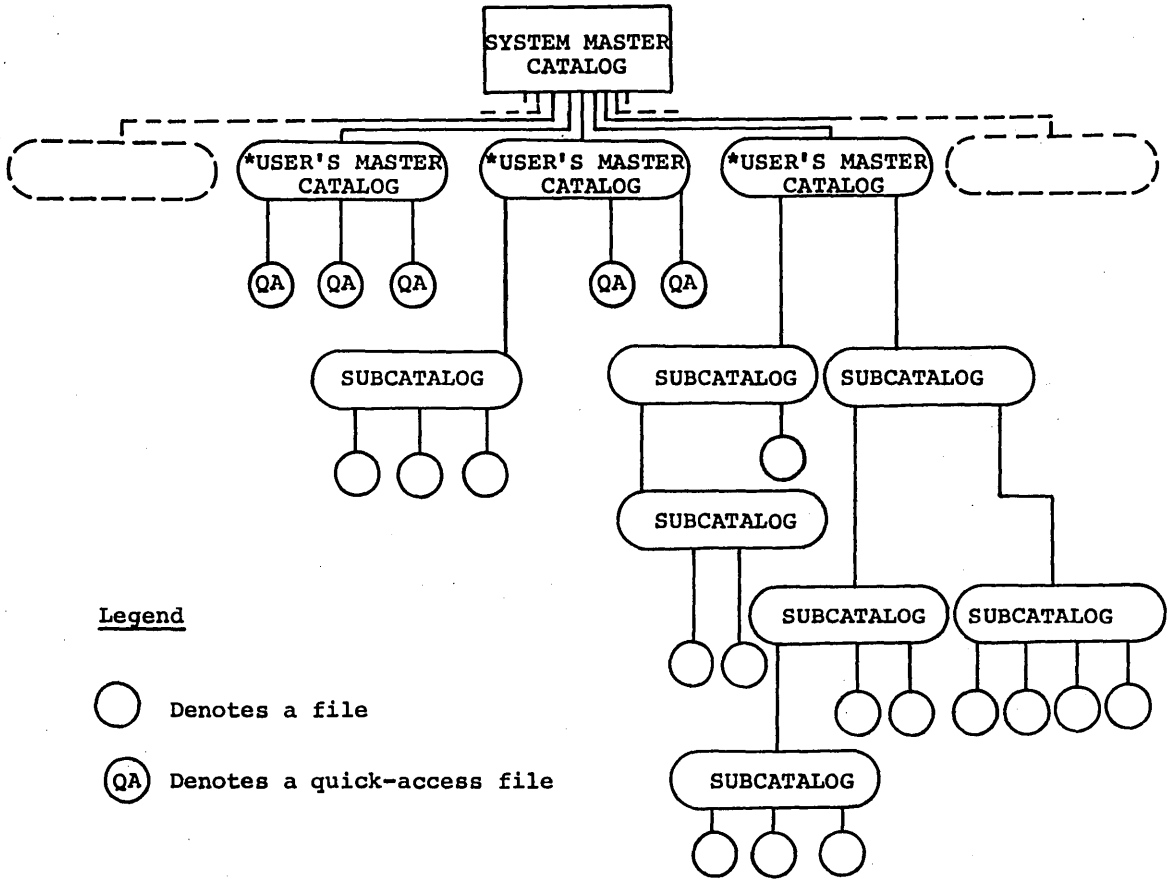
The GECOS file system is described in detail in the GECOS File System manual, CPB-1513. However, the main features of interest to the user will be repeated here.

The GECOS file system represents a tree structure of indefinite length whose origin is the system master catalog. The primary nodes of the tree are user's master catalogs; the lower-level nodes are subcatalogs created by the user. The terminal points of the structure are the files themselves. Figure 1 shows the file system's hierarchical structure.

Catalogs and Files

A catalog consists of a definition containing a catalog name, password, and permissions. Since it contains no user data, a catalog can be neither read nor written, but it is constructed and maintained by the file system itself. The ACCESS utility routine is provided, however, to make catalog changes desired by a user.

A file known to the GECOS file system consists of a definition containing file name, file size, password, permissions, and a description of the physical file space. The file definition is distinct from the physical file space which may contain user data and can be read or written.



All user-ID's must be unique within the system; all subcatalog and file names are automatically qualified by the user's master catalog name and the names of any intermediate subcatalogs. The system master catalog cannot be accessed by the normal user.

---

*Identified by the user-ID.

Figure 1.

Logical Structure of the File System

Passwords

Passwords may be attached to any catalog or file. A password simply allows a user to traverse a catalog/file string. A user can get to a given catalog or file only if he can give the passwords for all higher-level catalogs in the string. (When traversing a string, a password must not be given if none has been attached.) The originator of a given string is required to give the necessary passwords when traversing a string.

Permissions

Users permissions, both general and specific, can be attached to any catalog or file. When permissions are attached at the catalog level, they apply to all subordinate catalogs and files. The originator of a catalog/file string implicitly has all permissions for that string but must give all applicable passwords.

The allowable permissions are:

READ - allows a file to be read
WRITE - allows a file to be written
APPEND - (presently treated as WRITE)
EXECUTE - (presently treated as READ)
PURGE - allows catalogs and/or files to be purged from the system, but only with specific permission
MODIFY - allows catalog and/or file definitions to be changed, but only with specific permission

SAVING AND RETRIEVING FILES IN 600TSS

When operating under the Time-Sharing System, each updated copy of an OLD file or NEW file operated on by BASIC or EDITOR is a temporary working file that will "disappear" at the end of a user's session at the terminal, unless he saves it. This temporary working file, or "scratch copy," allows updates to be made and tested without destroying the original OLD file.

Files are stored with the command SAVE followed by a file name. If the named file has not been previously created, the system automatically creates a permanent, external file and writes the contents of the working file onto it. These files are referred to as quick-access. This means that the file emanates directly from a user's master catalog without intervening subcatalogs (see Figure 1.). Files created by the command SAVE have, by default, general read permission, and are protected against any other form of access, such as write, append or purge.

If the file specified by the SAVE command has been created previously, either by a prior SAVE or by the ACCESS subsystem, the content of the working file is written onto it. This file, if created through ACCESS, is not necessarily of the quick-access type.

The creator of a quick-access file can retrieve it by using OLD and the file name. Therefore, for the user whose needs are met by the characteristics of quick-access files, TSS provides a simple means for utilizing the file system.

ACCESS

Capabilities of the ACCESS Subsystem

For users who wish to utilize some or all of the capabilities of the file system, the ACCESS subsystem provides the interface thus allowing the user to:

- Create hierarchical structures of subcatalogs and files
- Attach passwords to his subcatalogs and files
- Give general permission to all other users to access his files in specified ways
- Give specific permissions, by user-ID
- Protect a given file or set of files against any mode of access
- Gain the permitted types of access to other TSS user's files

- Gain the permitted types of access to files created in the batch-processing environment
- Modify catalog name, password, and/or permissions on an existing catalog
- Modify file name, size, password, and/or permissions on an existing file
- Purge an existing file or catalog/file string
- List all of the catalogs and files which emanate from a given catalog
- Rename files temporarily for a given job

HOW TO USE ACCESS

ACCESS consists of ten functions, which together provide a conversational facility for:

- Creating and purging catalogs and files
- Modifying catalog and file attributes (name, size, password, permissions)
- Accessing and deaccessing files
- Listing catalogs and files

The operation of ACCESS consists of responses, via the terminal, to a sequence of English-language questions. Each function is characterized by a given sequence of questions. All of the "standard vocabulary" associated with the user's responses may be abbreviated for convenience in keying-in.

It should be remembered that ACCESS is not a means of reading or writing permanent file content. OLD and SAVE perform these functions. ACCESS is selected before proceeding to a desired processing subsystem; it is used to "create" or "access" the file, i.e., the file definition and the file space, before the substantive file is built or modified under another subsystem. The OLD/NEW sequence and the SAVE command of the succeeding subsystem(s) are still applicable.

Several general points need to be made in connection with the use of ACCESS:

(1) The definition of a particular catalog or file must include the names of all higher-level catalogs that must be traversed to arrive at that point. The catalog string would include at least the user's master catalog. A file definition, then, is the complete catalog string plus the file name.

It would be inconvenient to give the full file definition each time a file were referred to, so the processing subsystem through the use of the commands OLD and SAVE requires a reference by file name only. This file name is usually the actual one, i.e., the same file name that terminates the full definition. However, since this actual file name might not be unique for all files to be used in one session at the terminal, it is necessary that an alternate file name be supplied for unusual situations. The ACCESS specification of an alternate file name does not change the file definition; it is a local and temporary renaming for the duration of the user's session at the terminal.

(2) Each user's master catalog must be created for him before he can use the system. It has no password or permissions associated with it, and is unalterable. The installation usually controls the generation of this catalog.

(3) The Time-Sharing System maintains an available file table (AFT) for each user. This table is a list, by file name, of the files that the user is going to use during a session at his terminal. All files to be referenced by the SAVE and OLD must either have been placed in AFT by using ACCESS, or be of the quick-access type. When quick-access files are referenced by SAVE or OLD, they are placed in the AFT automatically by the system.

(4) Specific permissions replace general permissions; they do not add to them. That is, if all time-sharing users are given READ permission (general), and users JDOE and FSMITH are given specific WRITE and MODIFY permissions only, for a given file, JDOE and FSMITH cannot read the file. Therefore, in assigning specific permissions, the assignor must specify all permissions granted to the user.

Functions

The initial communication from ACCESS, following system selection, is a request for a choice of function, i.e., FUNCTION?.

The functions that may be requested are:

	(Abbrv.)
• CREATE CATALOG	CC
• CREATE FILE	CF
• ACCESS FILE	AF
• DEACCESS FILE	DF
• MODIFY CATALOG	MC
• MODIFY FILE	MF
• PURGE CATALOG	PC
• PURGE FILE	PF
• LIST CATALOG	LC
• LIST SPECIFIC	LS

The effect produced by each function is as follows:

CREATE CATALOG - this function creates a subcatalog.

CREATE FILE - this function defines file space and attributes for a given file name. It does not bring a file into the available file table.

ACCESS FILE - this function brings a file into the available file table.

DEACCESS FILE - this function takes a file out of the available file table.

MODIFY CATALOG - this function modifies the name, password, and/or permissions associated with a given catalog.

MODIFY FILE	- this function modifies the name, size, password, and/or permissions associated with a given file.
PURGE FILE	- this function deletes a file from the system.
PURGE CATALOG	- this function deletes a catalog from the system along with any catalogs and files which are subordinate to it.
LIST CATALOG	- lists the names of the catalogs and files which emanate from this catalog.
LIST SPECIFIC	- lists in detail the description of the catalog or file specified.

Following the response to FUNCTION, ACCESS asks the user to describe the catalog-string, catalog, or file. Each function has a fixed set of questions with several of the questions common to each set. Some of the questions do not logically require a response, e.g., PASSWORD? (there may be none). If no response is applicable, only a carriage return is given.

All the functions, except DEACCESS FILE, first request a definition of the existing catalog-string. Then the name of the catalog or file to be processed is next, along with size attributes in the case of a file. Passwords and permissions are then requested, as appropriate.

Questions and Responses

The sets of questions associated with each function follow, along with the general form of the response to each question. The minimum required response, if any, is underlined. Each set is followed by illustrative examples.

(1) FUNCTION? CREATE CATALOG (CR)

CATALOG STRUCTURE TO WORKING LEVEL?

user-ID/cat-name$password/.../cat-name$password (CR)

NEW CATALOG? cat-name (CR)

PASSWORD? password (CR)

GENERAL PERMISSIONS? access-type,...,access-type (CR)

The access types are:

READ	(or R)	
WRITE	(or W)	
APPEND	(or A)	
EXECUTE	(or E)	
MODIFY	(or M)	(Specific permission only)
PURGE	(or P)	(Specific permission only)

SPECIFIC PERMISSIONS?

access-type,...,access-type/user-ID/.../user-ID (CR)

The access types are the same as for general permissions.

NOTE: If no response to the question SPECIFIC PERMISSION? is given, i.e., only a carriage-return, the catalog is created and the question NEW CATALOG? is reissued.

Example Replies (user responses are underlined):

FUNCTION? CREATE CATATOG (CR)

CATALOG STRUCTURE TO WORKING LEVEL?

JDOE/CAT1$ABC (CR)

This response says that there is a subcatalog named CAT1 that is concatenated directly to the user's master catalog identified by the user-ID JDOE, and that it is desired to create a new catalog from this level. The password ABC was attached to catalog CAT1 when it was created.

NEW CATALOG? CAT2 (CR)

This response indicates the name of the catalog created at this point.

PASSWORD? AOK (CR)

This response associates the password AOK with this catalog. A carriage-return alone would indicate that no password is to be assigned.

GENERAL PERMISSIONS? (CR)

The lack of a response here indicates that general permission is not granted at this level for any type of access to subsumed files. A response of READ, EXECUTE indicates that any unspecified user has permission to read and execute (if meaningful) any file that emanates from this catalog.

SPECIFIC PERMISSIONS? READ/BJONES/ASMITH (CR)

SPECIFIC PERMISSIONS? READ,WRITE,PURGE/ALLONG (CR)

This combination of responses says that the users who have logged onto the TSS system under the names BJONES and ASMITH can pass through this level with read permission for any files below, and that the user ALLONG can pass through with read, write, and purge permissions.

SPECIFIC PERMISSION? (CR)

The carriage-return alone means that no further specific permissions are to be given; the catalog is now created and the question:

NEW CATALOG?

is reissued, allowing the user to create another catalog at the same level, i.e., also emanating from CAT1.

Alternative forms of the response to CATALOG STRUCTURE TO WORKING LEVEL? are as follows:

/CAT1$ABC (CR)

Assuming the user to be JDOE, this response is equivalent to the one given above, JDOE/CAT1$ABC. The initial slant indicates the user's own master catalog.

A response of simply a slant, i.e.:

/ (CR)

indicates that the user desires to create directly from his master catalog. This response is equivalent to his user-ID alone.

(2) FUNCTION? CREATE FILE (CR)

CATALOG STRUCTURE TO WORKING LEVEL?

user-ID/cat-name$password/.../cat-name$password (CR)

FILE NAME, SIZE, MAX SIZE?

file name, initial size (links), maximum size (links) (CR)

PASSWORD? password (CR)

GENERAL PERMISSIONS? access-types,...,access-type (CR)

The access-types are:

READ	(or R)	
WRITE	(or W)	
APPEND	(or A)	
EXECUTE	(or E)	
MODIFY	(or M)	(Specific permission only)
PURGE	(or P)	(Specific permission only)

SPECIFIC PERMISSIONS?

access-type,...,access-type/user-ID.../user-ID (CR)

Example Replies (responses are underlined):

FUNCTION? CF (CR)

CATALOG STRUCTURE TO WORKING LEVEL?

/CAT1$ABC/CAT2$AOK (CR)

This response defines user-ID/CAT1/CAT2 as the catalog-string from which the file is to emanate. The initial slant indicates that the succeeding string is concatenated to the user's own master catalog.

FILE NAME, SIZE, MAX SIZE? FIL1,1,3 (CR)

This response asks for a file space of 1 link, initially, with a maximum eventual size limit of 3 links, named FIL1.

PASSWORD? (CR)

No password is assigned to this individual file.

GENERAL PERMISSIONS? READ (CR)

SPECIFIC PERMISSIONS? (CR)

None are granted at this level, but the ones granted at the level of CAT2 (CREATE CATALOG in the previous example), apply to this file.

The lack of a response means the end of the information relevant to the creation of this file. The file is created, and the question

FILE NAME, SIZE, MAX SIZE?

is reissued. This permits creation of other files at the same level.

(3) FUNCTION? ACCESS FILE (CR)

CATALOG STRUCTURE TO WORKING LEVEL?

user-ID/cat-name$password/.../cat-name$password (CR)

FILE NAME$PASSWORD? file name (alternate name) $password (CR)

PERMISSIONS DESIRED?

access-type,...,access-type (CR)

The access types are:

- READ (or R)
- WRITE (or W)
- APPEND (or A)
- EXECUTE (or E)

Example Replies (responses are underlined):

FUNCTION? ACCESS FILE (CR)

CATALOG STRUCTURE TO WORKING LEVEL?

JDOE/CAT1$ABC/CAT2$AOK (CR)

The user in this case is not the creator of the file to be accessed, so he must define the user's master catalog (e.g., JDOE) from which the file emanates, along with any required subcatalogs and password.

FILE NAME$PASSWORD? FIL1 (CR)

If a password were required, it would be concatenated to the name with a dollar-sign ($), i.e., FIL1$ABC.

PERMISSIONS DESIRED? READ (CR)

General read permissions was granted for this file. (Several specific read permissions were also granted at the level immediately above CAT2). Termination of this response with only a carriage return causes the file to be accessed and the request:

FILENAME$PASSWORD?

to be reissued.

(4) FUNCTION? DEACCESS FILE (CR)

FILE NAME? file name (or CLEARFILES) (CR)

The response for this function is the name of the file to be deaccessed. The name supplied is always the name under which the file was accessed, whether this was the actual name or a temporary alternate name. If CLEARFILES is used, all of the user's available files are deaccessed including his temporary files.

(5) FUNCTION? PURGE CATALOG (CR)

CATALOG STRUCTURE TO WORKING LEVEL?

user-ID/cat-name$password/.../cat-name$password (CR)

CAT. TO BE PURGED? cat-name (CR)

PASSWORD? password (CR)

Example Replies (responses are underlined):

FUNCTION? PC (CR)

CATALOG STRUCTURE TO WORKING LEVEL?

/CAT$ABC (CR)

This response defines the subcatalog CAT1 concatenated to the user's own master catalog.

CAT. TO BE PURGED? CAT2 (CR)

PASSWORD? AOK (CR)

The dollar-sign is used only when the password is concatenated directly to a file or catalog name. The request

CAT. TO BE PURGED?

is reissued.

(6) FUNCTION? PURGE FILE (CR)

CATALOG STRUCTURE TO WORKING LEVEL?

user-ID/cat-name$password/.../cat-name$password (CR)

FILE TO BE PURGED? file name (CR)

PASSWORD? password (CR)

Example Replies (responses are underlined):

FUNCTION? PF (CR)

CATALOG STRUCTURE TO WORKING LEVEL?

JDOE /CAT1$ABC/CAT2$AOK (CR)

The user in this case is ALLONG, not the file creator.

FILE TO BE PURGED? FIL1 (CR)

PASSWORD? (CR)

The user (ALLONG) was given specific purge permission at the level of CAT2.

The request

FILE TO BE PURGED?

is reissued.

(7) FUNCTION? MODIFY CATALOG (CR)

CATALOG STRUCTURE INCLUDING CATALOG TO BE MODIFIED?

user-ID/cat-name$password,...,cat-name$password (CR)

NEW NAME? new cat-name (CR)

PASSWORD? { new password / DELETE } (CR)

GENERAL PERMISSIONS? { access-type,...,access-type / DELETE } (CR)

SPECIFIC PERMISSIONS? { access-type,...,access-type/user-ID.../user-ID / DELETE/user-ID/.../user-ID } (CR)

Example Replies (user responses are underlined):

FUNCTION? MC (CR)

CATALOG STRUCTURE INCLUDING CATALOG TO BE MODIFIED?

/CAT1$ABC/CAT2$AOK (CR)

NEW NAME? (CR)

A carriage-return only response means that the catalog name is to remain unchanged.

PASSWORD? XYZ (CR)

The original password AOK is replaced by XYZ.

GENERAL PERMISSIONS? READ (CR)

As originally created, general permissions were not assigned at this level. This reponse replaces this null set with READ permission.

SPECIFIC PERMISSIONS? R,W/BJONES (CR)

This response replaces the original specific READ permission for BJONES with READ and WRITE permission.

SPECIFIC PERMISSIONS? DELETE/ASMITH (CR)

This response cancels any permissions for ASMITH that previously existed.

SPECIFIC PERMISSIONS? R,W,P,M/ALLONG (CR)

This response replaces the original set of READ, WRITE and PURGE permissions for ALLONG with READ, WRITE, PURGE, and MODIFY.

SPECIFIC PERMISSIONS? (CR)

The carriage-return above implies that no further modifications are to be made; the changes are now processed and the question:

CATALOG STRUCTURE INCLUDING CATALOG TO BE MODIFIED?

is reissued.

(8) FUNCTION? MODIFY FILE (CR)

CATALOG STRUCTURE INCLUDING FILE TO BE MODIFIED?

user-ID/cat-name$password/.../
cat-name$password/file-name$password (CR)

NEW NAME? new file name (CR)

NEW MAX SIZE? new maximum size (in links) (CR)

PASSWORD? { new password / DELETE } (CR)

GENERAL PERMISSIONS? { access-type,...,access-type / DELETE } (CR)

SPECIFIC PERMISSIONS? { access-type/user-ID/.../user-ID / DELETE/user-ID/.../user-ID } (CR)

Example Replies (responses are underlined):

FUNCTIONS? MF (CR)

CATALOG STRUCTURE INCLUDING FILE TO BE MODIFIED?

/CAT1$ABC/CAT2$XYZ/FIL1 (CR)

NEW NAME? MASTER1 (CR)

NEW MAX SIZE? 5 (CR)

This response increases the maximum file size to 5 links (originally 3).

PASSWORD? DEPT37 (CR)

This response attaches the password DEPT37 to this file (none originally assigned).

GENERAL PERMISSIONS? DELETE (CR)

The original general READ permission is deleted.

SPECIFIC PERMISSIONS? P/BJONES (CR)

PURGE permissions for user BJONES is added at this level. This permission applies to this file only, but he also has READ and WRITE from the CAT2 level.

(9) FUNCTION? LIST CATALOG (CR)

CATALOG STRUCTURE INCLUDING CATALOG TO BE LISTED?

user-ID/cat-name,...,cat-name (CR)

Example Replies (user responses are underlined):

FUNCTION? LC (CR)

CATALOG STRUCTURE INCLUDING CATALOG TO BE LISTED?
/CAT1

Passwords need not be given in the catalog structure. A user is permitted to list only his own catalogs on the LIBRARY catalog.

A list of the catalogs and files emanating from CAT1 would now be output.

(10) FUNCTION? LIST SPECIFIC (CR)

CATALOG STRUCTURE INCLUDING CATALOG OR FILE TO BE LISTED?

user-ID/cat-name,...,cat-name(or)file-name (CR)

Example Replies (user responses are underlined):

FUNCTION? LS (CR)

CATALOG STRUCTURE INCLUDING CATALOG OR FILE TO BE LISTED?

/CAT1

Passwords need not be given in the catalog structure and will not be included in the catalog or file description which is output. A user can list only his own catalogs or files.

The description of CAT1 would now be output.

Identifiers and Delimiters in User Responses

User responses are composed of the following:

- Identifiers
- Keywords
- Word delimiters
- Line delimiters

Identifiers consist of file names, catalog names, user-IDs and passwords. They can consist of alphabetics, numerics, periods, and minus signs. Each identifier can be up to 12 characters in length except file names which are limited in length to 8 characters.

In response to the question FILE NAME$PASSWORD?, issued by the Access File function, a file name of up to 12 characters may be specified, i.e., the name of a batch-environment file, if followed by an alternate name of 8 characters or less, enclosed in parentheses. Also, in response to FILE TO BE PURGED?, a file name of up to 12 characters could be specified, if the file to be purged were not created in the TSS environment.

Keywords consist of function names, access types (permissions), and several file-type parameters, of limited interest, that are described under "Special Features." Keywords are used in responses to questions, and can always be abbreviated to the initial character, or a two-character acronym in the case of function name, e.g., R for READ permission or CC for CREATE CATALOG function.

The file-size specification in the response to FILE NAME, SIZE, MAX SIZE? (Create File), is a decimal number denoting the number of links required. This may be considered a special case of keyword.

Word delimiters are used in user responses: the slant or virgule (/), the dollar-sign ($), and the comma (,). Blanks may be used freely in responses except within function names; they are in no sense delimiters and are ignored.

The use of the three delimiters is as follows:

The / delimiter has two functions:

(1) In catalog-strings a slant indicates that a subcatalog name follows and is concatenated to the preceding catalog in the string. An initial slant indicates that the following subcatalog-string (if any) is concatenated to the user's master catalog. A response to CATALOG STRUCTURE TO WORKING LEVEL? of / (CR) is equivalent to the user's own user-ID, i.e., it positions the user to his own master catalog.

(2) In specific permissions a slant indicates that a user-ID follows.

The $ is used only to concatenate a password to a catalog or file name.

The , is used as a general separator for keywords, i.e., for separating access-types and sizes, and separating file names from the following keywords or sizes.

The line delimiters are a carriage-return, an asterisk (*) plus carriage-return, or a double asterisk (**) plus a carriage-return. Each of these serves to terminate a response, but with a different effect.

(1) Carriage Return: A carriage-return following a response generally signifies that the user wishes to remain at the same catalog position (if relevant), and proceed to the next question in logical sequence. This may be the next question in a set, or the initial question again.

When only a carriage-return is given, however, i.e., a "null" response, it has several possible meanings:

- In response to the question CATALOG STRUCTURE TO WORKING LEVEL? a carriage-return-only is equivalent to the user's own user-ID or a slant, i.e., / (CR). Any of these responses requests that the user be positioned to his own master catalog.

- A carriage-return-only following a question that logically requires a response, e.g., NEW CATALOG?, causes an immediate return to the question FUNCTION?.

- The question SPECIFIC PERMISSIONS? recurs each time a response is given (delimited by a carriage-return), since only one set of specific permissions can be given in each. If only a carriage-return is given, the information received so far is processed, and the first question below CATALOG STRUCTURE TO WORKING LEVEL? is reissued, i.e., NEW CATALOG? or FILE NAME, SIZE, MAX SIZE?, allowing a new catalog or file to be created at the same catalog level.

- A carriage-return only response to FUNCTION? returns the question SYSTEM?.

(2) Single Asterisk Plus Carriage-Return: If a single asterisk plus a carriage-return is given in reply to a question, either with or without a substantive response, ACCESS processes the information it has and returns to the first question at the same catalog level, e.g., to skip any further questions in the set.

(3) Double Asterisk Plus Carriage-Return: If a double asterisk plus a carriage-return is given, either with or without a substantive response, ACCESS processes the information it has and returns to the question "FUNCTION?". It implies that the user is finished with the current function.

In addition to the changes in level of operation produced by the several line delimiters, a response of "DONE" to any question causes an exit from ACCESS. No processing is performed and the question "SYSTEM?" results.

The line delimiters show that there are several ways of either shortening the question-response sequence, or terminating a function at any given point.

Examples of the effect of different response terminations:

FUNCTION? CC (CR)

CATALOG STRUCTURE TO WORKING LEVEL?

(CR) (The carriage-return only implies master catalog.)

NEW CATALOG? 001* (CR)

(This indicates that passwords or permissions are not wanted for this catalog and no further questions are wanted.)

NEW CATALOG? 002 (CR)

PASSWORD? PASS2** (CR)

(This implies that no permissions are to be assigned to this catalog, and that creating catalogs at this position is finished.)

FUNCTION? CF (CR)

CATALOG STRUCTURE TO WORKING LEVEL?

/002$PASS2 (CR)

FILE NAME, SIZE, MAX SIZE? 02.1,1,3 (CR)

GENERAL PERMISSIONS? READ (CR)

SPECIFIC PERMISSIONS? W/RJJONES** (CR)

(This implies that creating files at this level has been completed.)

FUNCTION? (CR) (or DONE (CR))

(finished with ACCESS)

SYSTEM?

General Usage Rules

The ability of a user to access files and otherwise manipulate catalog/file structures, e.g., modifying and purging, depends upon his knowing the necessary file definitions. Beyond this, the file system has two file and catalog protection features: passwords and permissions.

Permissions provide the file creator a positive protection feature; if permissions are not explicity granted, his catalogs and files are completely protected by default. The user must assign to others any degree of access he wishes them to have. But, note that since specific permissions for a given user do not add to, but replace, any general permission that may have been given, specific permissions may be used to exclude a given set of users from one or more types of access.

Passwords provide an additional level of protection. If passwords are assigned by the creator of a catalog/file string they must be supplied in order to pass through the string.

The creator of a catalog/file string is exempt from any access-mode restrictions he imposes, i.e., he implicitly has all permissions for his catalogs and files, but he must give all passwords.

The MODIFY permission, which allows another user to change file names, catalog names, file size, passwords, and/or permissions, also implies the ability of this other user to create catalogs and/or files emanating from the master catalog.

Special Features

Using the Create-File function, previously described, the files created are not necessarily contiguous; i.e., successive links of a multi-link file are not necessarily in physical sequence on the storage device. Furthermore, both

the Create-File and Access-File functions assume that the file will be treated as a linked file. For the standard subsystems provided with 600TSS, these file characteristics are suitable because linked files are required.

If, however, in the use of a given subsystem, it would be advantageous to have contiguous files, this characteristic can be specified in response to FILE NAME, SIZE, MAX SIZE?. The form of this response is:

file name,initial size C

The parameter "C" indicates, in Create File only, that a contiguous file is desired. No maximum size may be specified.

Similarly, if random treatment of files is required in a given user-written subsystem, a file can either be created as a random file or accessed as a random file. If created as such, it is always treated by the GECOS I/O Supervisor as a random file. If it is created as a linked file, it can be accessed as a random file, but in that case, the random treatment indication is temporary, i.e., it applies to that access only.

The forms of the random specification are as follows:

For Create-File, the response to FILE NAME, SIZE, MAX SIZE? is:

file-name,initial size,maximum size,R

(or)

file name,initial sizeC,R

For Access-File, the response to FILE NAME$PASSWORD? is:

file-name,R$password

In both responses, the parameter R (always preceded by a comma) indicates that the named file is to be treated as a random file.

REQUEST DENIED MESSAGE

The following messages are printed following a complete function request, and indicate that the request could not be satisfied. The reason for denial is given in each case.

REQUEST DENIED-NEW NAME SAME AS AN EXISTING NAME

A new catalog or file name has been given that is the same as an existing catalog or file name at the same level.

REQUEST DENIED-FILE SPACE REQUESTED EXCEEDS ALLOWED

The user has requested file space exceeding the amount that has been allotted to him in his System Master Catalog entry.

REQUEST DENIED-NEW SIZE LESS THAN CURRENT SIZE

In MODIFY FILE, a new file size has been specified which is less than that currently used by the file.

REQUEST DENIED-SYSTEM MALFUNCTION

An unrecoverable I/O error has occurred.

REQUEST DENIED-PERMISSION NOT GRANTED

The user does not possess the requested permission(s).

REQUEST DENIED-FILE BUSY

The requested file is currently busy to the type of permission(s) requested.

REQUEST DENIED-INCORRECT CAT/FILE DESCRIPTION

This denial is given whenever required passwords are not included or the catalog/file description is not logically correct.

REQUEST DENIED-SYSTEM LOADED

The requested file function cannot be completed because there is temporarily no file space available.

REQUEST DENIED-YOUR AVAILABLE FILE TABLE IS FULL

The user has too many files accessed (open) at the same time. This situation can be eliminated by deaccessing some of the accessed files.

REQUEST DENIED-FILE NAME A DUPLICATE, MUST GIVE ALTERNATE NAME

An ACCESS FILE has been done where the file name is a duplicate of a file which the user currently has open. The alternate name capability can be used to avoid this situation.

INPUT ERROR MESSAGES

The following messages are printed immediately following the input in error, and the original question is repeated.

ERR-ILLEGAL CHARACTER

A character other than an alphabetic, numeric, period, or dash has been included in an identifier. An upward arrow (↑) points at the character in error.

ERR-INVALID DELIMITER

An otherwise valid delimiter has been given out of place. An upward arrow (↑) points at the delimiter in error.

ERR-XXXXXXXXXXXX-MUST BE LESS THAN 13 CHARACTERS

The designated identifier is limited to 12 characters.

ERR-XXXX-IS NOT A LEGITIMATE PERMISSION

Legitimate permissions are READ, WRITE, APPEND, and EXECUTE, plus PURGE and MODIFY as specific permissions only.

ERR-XXXXXXXXXX-MUST BE LESS THAN 9 CHARACTERS

The designated identifier is limited to 8 characters.

ERR-XXXX-MUST BE ALL NUMERIC

A non-numeric character has been included in field XXXX.

ERR-XXXXX-MUST BE LESS THAN 1000

The field is limited to three digits.

ERR-INPUT REQUIRED

A null response was given to a question which requires input.

ERR-INITIAL SIZE GREATER THAN MAX SIZE

In defining the file size an initial size greater than the maximum size was given.

APPENDIX C

The HELP Subsystem

The HELP subsystem will supply, at the user's terminal, an explanation of any number coded error message encountered while using BASIC. For example, if the message

```
009 - SYSTEM UNKNOWN
```

appears, the HELP subsytem will provide an explanation and suggestions for corrective action.

The HELP subsystem is very easy to use. It is selected in the same manner the user would employ to obtain access to EDITOR. In response to SYSTEM ? HELP is typed. The HELP subsystem then prints

```
PLEASE ENTER ERROR CODE -
```

If the user required an explanation of the message given in the example, he would type in the number-code 009.

The example can be illustrated as follows:

```
SYSTEM ? HELP

PLEASE ENTER ERROR CODE - 009

THE REQUESTED SYSTEM IS UNKNOWN OR IS
NOT INCLUDED IN THE SYSTEM FOR THIS
INSTALLATION. CHECK THE NAME FOR
SPELLING TOO.

SYSTEM ? done
```

APPENDIX D

OCTAL/ASCII II CONVERSION EQUIVALENTS

OCTAL NUMB.	ASCII CHAR.	OCTAL NUMB.	ASCII CHAR.	OCTAL NUMB.	ASCII CHAR.	OCTAL NUMB.	ASCII CHAR.
000	NULL	040	SP	100	@	140	GRA
001	SOH	041	EXP	101	A	141	a
002	STX	042	"	102	B	142	b
003	ETX	043	#	103	C	143	c
004	ECT	044	$	104	D	144	d
005	ENQ	045	%	105	E	145	e
006	AXK	046	&	106	F	146	f
007	BELL	047	'	107	G	147	g
010	BSP	050	(	110	H	150	h
011	HT	051	)	111	I	151	i
012	LF	052	*	112	J	152	j
013	VT	053	+	113	K	153	k
014	FFD	054	,	114	L	154	l
015	CR	055	-	115	M	155	m
016	SO	056	.	116	N	156	n
017	SI	057	/	117	O	157	o
020	DLE	060	0	120	P	160	p
021	DC1	061	1	121	Q	161	q
022	DC2	062	2	122	R	162	r
023	DC3	063	3	123	S	163	s
024	DC4	064	4	124	T	164	t
025	NAK	065	5	125	U	165	u
026	SYN	066	6	126	V	166	v
027	ETB	067	7	127	W	167	w
030	CAN	070	8	130	X	170	x
031	EM	071	9	131	Y	171	y
032	SUB	072	:	132	Z	172	z
033	ESC	073	;	133	LBK	173	LBR
034	FS	074	LTN	134	RSL	174	VTL
035	GS	075	=	135	RBK	175	RBR
036	RS	076	GTN	136	CFX	176	NOT
037	US	077	?	137	_	177	DEL

APPENDIX E

COMMUNICATIONS CONTROL

ACK	Acknowledgement
CAN	Cancel
DC1	Device Control 1
DC2	Device Control 2
DC3	Device Control 3
DC4	Device Control 4
DLE	Data Link Escape
EM	End of Medium
ENQ	Enquiry
EOT	End of Transmission
ESC	Escape (Alternate Mode)
ETB	End of Transmission Bloc
ETX	End of Text
NAK	Negative Acknowledgement
SOH	Start of Heading
STX	Start of Text
SUB	Substitute Character
SYN	Synchronous Idle

FORM EFFECTORS

BSP	Backspace
CR	Carriage Return
FFD	Form Feed
HT	Horizontal Tabulation
LF	Line Feed
VT	Vertical Tabulation

ITEM SEPARATORS

FS	File Separator
GS	Group Separator
RS	Record Separator
US	Unit Separator

TEXT MATERIAL

BELL	Bell, or other attention signal
CFX	Circumflex
DEL	Delete (Rubout)
EXP	Exclamation Point
GRA	Grave (reversed) Accent
GTN	Greater Than: Mathematical symbol
LBK	Left Bracket
LBR	Left Brace
LTN	Less Than: Mathematical symbol
NOT	Not: Mathematical symbol
NULL	Null
RBK	Right Bracket
RBR	Right Brace
RSL	Reverse Slash
SI	Shift In
SO	Shift Out
SP	Space
VTL	Vertical Line

APPENDIX F

GE-625/635 STANDARD CHARACTER SET

Standard Character Set	GE-Internal Machine Code	Octal Code	Hollerith Card Code	Standard Character Set	GE-Internal Machine Code	Octal Code	Hollerith Card Code
0	00 0000	00	0	↑	10 0000	40	11-0
1	00 0001	01	1	J	10 0001	41	11-1
2	00 0010	02	2	K	10 0010	42	11-2
3	00 0011	03	3	L	10 0011	43	11-3
4	00 0100	04	4	M	10 0100	44	11-4
5	00 0101	05	5	N	10 0101	45	11-5
6	00 0110	06	6	O	10 0110	46	11-6
7	00 0111	07	7	P	10 0111	47	11-7
8	00 1000	10	8	Q	10 1000	50	11-8
9	00 1001	11	9	R	10 1001	51	11-9
[	00 1010	12	2-8	-	10 1010	52	11
#	00 1011	13	3-8	$	10 1011	53	11-3-8
@	00 1100	14	4-8	*	10 1100	54	11-4-8
:	00 1101	15	5-8	)	10 1101	55	11-5-8
>	00 1110	16	6-8	;	10 1110	56	11-6-8
?	00 1111	17	7-8	'	10 1111	57	11-7-8
↗	01 0000	20	(blank)	+	11 0000	60	12-0
A	01 0001	21	12-1	/	11 0001	61	0-1
B	01 0010	22	12-2	S	11 0010	62	0-2
C	01 0011	23	12-3	T	11 0011	63	0-3
D	01 0100	24	12-4	U	11 0100	64	0-4
E	01 0101	25	12-5	V	11 0101	65	0-5
F	01 0110	26	12-6	W	11 0110	66	0-6
G	01 0111	27	12-7	X	11 0111	67	0-7
H	01 1000	30	12-8	Y	11 1000	70	0-8
I	01 1001	31	12-9	Z	11 1001	71	0-9
&	01 1010	32	12	←	11 1010	72	0-2-8
.	01 1011	33	12-3-8	,	11 1011	73	0-3-8
]	01 1100	34	12-4-8	%	11 1100	74	0-4-8
(	01 1101	35	12-5-8	=	11 1101	75	0-5-8
<	01 1110	36	12-6-8	"	11 1110	76	0-6-8
\	01 1111	37	12-7-8	!	11 1111	77	0-7-8

INDEX

Please cut along this line

DOCUMENT REVIEW SHEET

TITLE: GE-625/635 Time-Sharing System

CPB #: 1515

From:

Name: _____

Position: _____

Address: _____

_____

Comments concerning this publication are solicited for use in improving future editions. Please provide any recommended additions, deletions, corrections, or other information you deem necessary for improving this manual. The following space is provided for your comments.

COMMENTS: _____

NO POSTAGE NECESSARY IF MAILED IN U.S.A.
Fold on two lines shown on reverse
side, staple, and mail.

STAPLE

STAPLE

FOLD

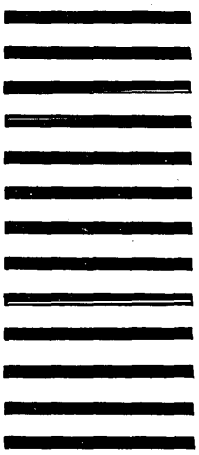
FIRST CLASS
PERMIT, No. 4332
PHOENIX, ARIZONA

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

GENERAL ELECTRIC COMPANY
PROCESSOR EQUIPMENT DEPARTMENT
13430 NORTH BLACK CANYON HIGHWAY
PHOENIX, ARIZONA 85029

ATTENTION: Program Documentation B-90
Systems and Processors Operation



FOLD

INFORMATION SYSTEMS

GENERAL ELECTRIC

LITHO U.S.A.