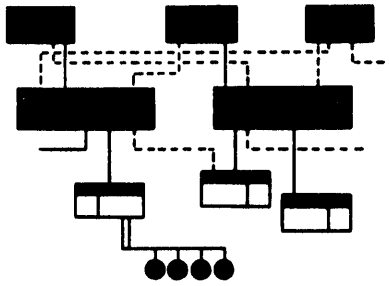
GENERAL ELECTRIC
COMPUTERS

GE-625/635
File and Record Control



GENERAL ELECTRIC

CPB-1003E

TBS:-RF-2113252

GENERAL ELECTRIC INFORMATION SYSTEMS DIVISION COMPUTER EQUIPMENT DEPARTMENT	GE-600 SERIES TECHNICAL INFORMATION BULLETIN	DATE Sept. 1968 NO. 600-232
SUBJECT: Additions to File Control Block (Section 1, Word -6)		REF. CPB-1003E

This TIB includes features implemented in GECOS-III System Development Letter 1.

Replace old pages in GE-625/635 File and Record Control reference manual, CPB-1003E, with attached new pages as follows:

Old	New
49-54	49-54

Vertical bars in the margins of these new pages indicate changes or additions to the existing text. This new information will be included in the next edition of the manual. The index changes required will also be made at that time.

Place this sheet in the front of your manual to show that the contents of this TIB have been incorporated.

TIBs that currently apply to CPB-1003E are:

600-194
600-211
600-232

GENERAL ELECTRIC INFORMATION SYSTEMS DIVISION COMPUTER EQUIPMENT DEPARTMENT	GE-600 SERIES TECHNICAL INFORMATION BULLETIN	DATE June 1968
		NO. 600-211
SUBJECT: Changes to: GEFRC reference manual		REF. CPB-1003E

Replace the old pages 65, 66 in the GE-625/635 File and Record Control Reference Manual, CPB-1003E, with the attached pages 65, 66.

GENERAL ELECTRIC INFORMATION SYSTEMS DIVISION COMPUTER EQUIPMENT DEPARTMENT	GE-600 SERIES TECHNICAL INFORMATION BULLETIN	DATE March 1968
		NO. TIB 600-194
SUBJECT: Addition of GEPR OVERRIDE		REF. CPB-1003E

This TIB includes features implemented in System Development Letter 12.

Please remove the following pages from your GE-625/635 File and Record Control reference manual and replace them with the attached revised pages of the same number.

45-46
47-48
49-50
51-52
53-54

It is suggested that you add this page to the front of your manual to show that this TIB has been entered.

GE-625/635
FILE AND RECORD CONTROL

REFERENCE MANUAL

April 1964

Rev. January 1968

GENERAL ELECTRIC

INFORMATION SYSTEMS DIVISION

CPB-1003E

PREFACE

The GEneral File and Record Control program (GEFRC) provides another category of operations control related to input and output operations of the Compatibles/600 family of information processing systems. Much of the operation of peripheral subsystems is accomplished by requests on GEFRC which, in turn, operates through the Input/Output Supervisor (IOS).

GEFRC permits job programs to regard all input/output data as being composed of records and files. Within similar classes or devices, information can be requested in these terms entirely independent of recording medium. Suitable tables supplied to GEFRC describe the location and format of each record and file. GEFRC interprets the request for a particular record and initiates a specific peripheral action to be transmitted to IOS for servicing.

This manual contains a general description of GEFRC, file control blocks and record format, buffers, calling sequences, label processing and unit switching, and error procedures.

This revised manual includes information previously published in CPB-1003D and supplemented with information previously published in Technical Information Bulletins 600-141, 600-156, and 600-167. In this revised edition, changes in technical content from the previous edition are identified with a bar in the margin opposite the change.

Suggestions and criticisms relative to form, content, purpose, or use of this manual are invited. Comments may be sent on the Document Review Sheet in the back of this manual or may be addressed directly to Documentation Standards and Publications, B-90, Computer Equipment Department, General Electric Company, 13430 North Black Canyon Highway, Phoenix, Arizona, 85029.

(1.8 1-68)

CONTENTS

APPENDICES

ILLUSTRATIONS

Figure

1. INTRODUCTION

The GEneral File and Record Control program (GEFRC), developed by the General Electric Computer Equipment Department, functions on any GE-625/635 computer configuration handling input and output media. The system is designed to relieve GE-625/635 computer users of the necessity of programming input/output routines. When using the system, the user need only concern himself with the information content of his files; the physical processes by which information is read or written need not be considered.

With modular construction furnishing maximum flexibility, GEFRC provides the user the following:

- The ability to consider inputs and outputs as files of arbitrary lengths, eliminating the need to program special routines for the operation of the devices involved
- The ability to interchange input/output media without performing extensive reprogramming
- The ability to improve the performance of a program through the use of record blocking and buffering and still retain program simplicity of unit record processing
- Source language compatibility of basic functions through the GE-625/635 Macro Assembler Program (GMAP) with Input/Output Supervisor (IOS) and the GEneral Comprehensive Operating Supervisor (GECOS)
- Automatic error detection and correction procedures
- Standardized operating procedures

As part of the user's programs GEFRC operates in close harmony with the system components. It is maintained in the system library and loaded into memory by the GEneral LOADer (GELOAD), which accomplishes the appropriate linking and symbol definition required to integrate the GEFRC routines with the object program.

The GEneral Comprehensive Operating Supervisor (GECOS) will provide certain commonly used exception routines which would otherwise be a part of GEFRC. Functions such as file control block initialization and issuing operator instructions are examples of routines available from GECOS. GECOS will retain certain data contained in the system file cards for use in the file control blocks of the object program. GECOS will also provide checkpoint facilities to the object program which may be optionally recorded on one of the files serviced by GEFRC.

An important component of GECOS is the Input/Output Supervisor (IOS), which provides the following functions of GEFRC:

- Initiates input/output activity on any device in the system
- Queues input/output requests on busy devices
- Responds to input/output termination interrupts
- Executes a logical to physical unit translation
- Executes a limit test on disc storage unit WRITE instructions achieving programmed file protection

GEFRC is an integral part of the GE-625/635 software system and eliminates much of the detailed coding for input/output operations. GEFRC is capable of handling input/output requirements for any of the common peripheral devices such as the perforated tape reader/punch, high-speed card reader, low-speed card punch, high-speed printer, magnetic tape subsystem, and disc/drum storage subsystem.

All calls to GEFRC subroutines are written in GMAP programs and in the standard GMAP instruction format. The programmer may never call a GEFRC subroutine when writing in any compiler language.

2. GENERAL DESCRIPTION

The File and Record Control (GEFRC) system consists of an integrated set of subroutines which perform all necessary input/output functions for user programs. Based upon the requests made by various calling sequences within the user program, the proper subset of GEFRC is appended at load time from the system library. The subset of GEFRC resides in slave memory with the user program.

GEFRC performs operations at the request of the various calling sequences in the program; the functional routines are further controlled by the file control blocks. A file control block must be created in assembler language for each file in the program and should contain information regarding the physical and processing characteristics of the file.

The user must reserve a buffer area for each file to be processed in terms of logical records. The buffer area for each file may contain either one or two buffers. Each buffer corresponds to a physical record of that file, and the number of buffers is specified in the file control block. The choice of one or two buffers must be based upon the program storage demand and upon the current file activity. GEFRC provides efficient processing with either choice.

The user program is written with calls to GEFRC routines to perform the necessary functions. The user program may request that a record be written on some file; however, the write request references a description of the file (file control block) rather than the device. The file control block, supplied by the user for each file, gives the necessary processing characteristics but not specific references to a device. At execution time the user supplies control cards to GECOS that specify the actual device (or type of device) to be assigned to that file for the current execution.

The label processing routines of GEFRC require that files have a standard format and that GE-625/635 standard format 14-word labels be used. These formats and standards are described in Chapter 12 of this manual.

Checkpoints--GEFRC does not provide any checkpoint facilities to the object program. Checkpoints may be taken by a special calling sequence to the Checkpoint routine in GECOS. GEFRC does, however, provide the Checkpoint routine with information regarding the record counts and file positions required by the checkpoint procedure. Checkpoints may be recorded anywhere within the text of a file. When checkpoints are contained within a file, they are recorded in the binary mode, preceded and followed by a special checkpoint sentinel file mark (octal 00) recorded in the mode of the file. When the special checkpoint is encountered by GEFRC, the checkpoint records will be bypassed.

Error Procedures--The user has the option of providing a routine for taking corrective action on certain types of operational errors. The location of this routine is specified in the file control block, with GEFRC entering the routine with an indication of the type of error encountered. After the routine has taken the desired action, a standard return will cause GEFRC to take specific action depending upon the original error.

To meet the input/output requirements of a program, GEFRC is dependent upon the user-supplied file specifications and selected calling sequences. The remainder of the GEFRC manual defines these items and explains the GEFRC responses to them.

3. FILE AND RECORD FORMAT

The format of a file varies greatly with the type of device to which it is assigned. For example, a file on a card reader or card punch is a deck of cards with each card representing a physical record of that file. On magnetic tape, a file consists of all the data between two standard file marks which may or may not have labels associated with them. The data recorded between two interrecord gaps constitute a physical record. In addition, if logical record processing is performed on this file, then the physical record will contain one or more logical records.

For disc and drum, two types of file formats are provided--linked and random. The linked type operates in much the same manner as magnetic tape. The data must be read and written in a serial manner with physical records of 320 words each. These files must be processed in terms of logical records, one or more of which occupy each 320-word physical record. Random disc and drum storage files are processed in a random manner. The order of processing and the amount read and written is entirely at the discretion of the user. Therefore, random files must be processed using GEFRC physical processing commands.

In order to achieve device interchangeability for data files, the standard system format must be used. This format coincides with the requirements of linked disc and drum (the devices with the strictest requirements). Therefore, when this format is used, the file can easily be assigned to other types of devices.

STANDARD SYSTEM FORMAT

A standard system format has been designed for the GE-625/635 to provide several significant capabilities:

- Complete freedom of device allocation between the drum, DSU, and magnetic tape
- Output media conversion from a "stacked" tape, where that tape may be accepting output from each of several programs
- Use of a common input/output routine for each of the various system processors, such as GMAP and GELOAD

The details of the standard system format are as follows:

1. Block size--Data blocks are variable in length up to a maximum block size of 320 words.
2. Block ~~serial number~~--A block serial number will exist as the first word of each data block and will contain two binary values as follows:

bits 0-17	Block serial number--The sequential number of this physical record (within the current reel if this file is on tape).
bits 18-35	Block size--The size of the block in words, not including the control word.

PHYSICAL RECORD = DATA BLOCK
LOGICAL RECORD = DATA RECORD

3. Record Mode--File will be recorded in the binary mode and high density, when magnetic tape is used.

4. Record Format--Records contained within each block are variable length and contain a record size control word as the first word of each variable-length record. The contents of this control word are as follows:

bits 0-17	Binary equivalent of record size in words, not including the control word. When the file is assigned to the drum or DSU, this value may be zero and the word is interpreted as a file mark analogous to a tape end-of-file marker.
bits 18-23	Not used unless bits 0-17 are zero, in which case this character position contains the specific file mark character.
bits 24-27	Zeros
bits 28-29	Logical record type code--Used by media conversion to determine the specific action required for each record. The following codes are assigned: 0 - Not a media conversion record 1 - Binary card image 2 - Hollerith card image 3 - Print-line image
bits 30-35	Report code identifying this record as one which belongs to a specific report or punch deck.

5. Labels--System format files contain standard header and trailer labels, when magnetic tape is used.

6. Stacked output files--When program output is to be stacked on the system output tape (Tape-SYSOUT), the first logical record of each block will be created by GEFRC as a special requirement of the media conversion program. This record will consist of the standard record size indicator word plus a single word containing program identification. Space for these two words must be included in the buffer(s).

TYPES OF RECORDS

Figure 1 illustrates the three types of records and shows how GEFRC obtains the record size for each.

Fixed-Length Records

When a file is defined as one containing fixed-length records, all data records of the file must conform to the established record size. When fixed-length records are blocked, they will normally be blocked so that the number of records in each block remains constant. Blocks containing fewer than the normal number of records will be processed with equal facility by GEFRC.

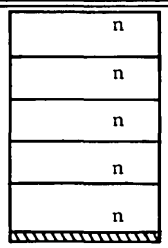
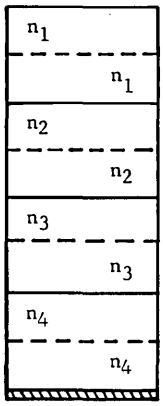
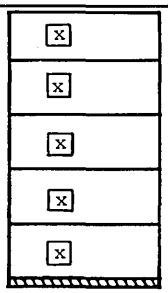
Record Type	Structure	GEFRC gets size by: Read	GEFRC gets size by: Write
FIXED		Record size field in FCB	Record size field in FCB
VARIABLE		First word of record	User inserts in record size field of FCB prior to CALL to write
MIXED		Enter user routine that tests "x" and inserts in record size field of FCB	User inserts in record size field of FCB prior to CALL to write

Figure 1. GEFRC Locates Record Size

Variable-Length Records

Variable-length records produced by GEFRC always include a record size control word as the first word of the record. Variable-length records created by GEFRC are limited to those files recorded in the binary mode.

Note: On disc or drum, linked files are always made up of variable-type records with block serial numbers. The block serial number contains the block size. An EOF for disc or drum has an extra word (included in block size) after the last logical record in the last block. This word has a record size of zero followed by the character 17_8 and two zero characters.

Mixed-Length Records

Mixed-length records are normally associated with files that contain several different record types each identified by a single procedure.

Having identified a record as to its type, the size is also known to the program. When a file is defined as containing this type of record, the size of each input record must be obtained from a user-supplied routine. The location of this routine is defined by the file control block macro-instruction, and is entered from GEFRC using the following calling sequence:

CALL size (fcb)

Where:

size = the symbolic location of the user-supplied routine to determine the record size
fcb = the symbolic location of the file control block

The user routine will determine the size of the record using the current record index (word 0 of the file control block) to locate the record. Once the size is determined, the value must be stored into word 1 of the file control block prior to return via the RETURN macro-instruction.

For output records of this type, the user must initialize the "record size" of the file control block with a binary integer equal to the number of data words in his output record. This must be done prior to execution of the "write" calling sequence.

Figure 1 (page 7) illustrates the three types of records and shows how GEFRC obtains the record size for each.

SPECIAL FILE MARKS

A GE-625/635 computer can write any character as a single-character record. When read, this is interpreted as a file mark. GEFRC has been designed with certain assumptions regarding the use of this file mark flexibility.

Only the normal end-of-file mark character (octal 17) is interpreted by GEFRC as a file delimiter. Any request to position a file forward or backward to a file mark results in a search for a file mark (octal 17) character.

When a checkpoint sentinel file mark (octal 00) is encountered, it and its checkpoint records are automatically bypassed by GEFRC. When any other file mark is read by GEFRC, control is transferred to the user's routine to allow that routine to determine the correct action required.

4. CALLING SEQUENCES

Files are opened, processed, and closed by the execution of calling sequences to the various routines of GEFRC. This chapter gives a detailed discussion of each calling sequence that may be used with GEFRC. Also included are the exception conditions that may arise with each calling sequence and the action taken by GEFRC. The user also has corrective action options under certain error conditions. These are discussed in the Error Procedures section of this chapter.

All requests from the object program for input/output activity are written in the format of the CALL system macro-instruction as defined by GMAP. The two types of calling sequences are as follows:

1. Record Processing--The record processing calling sequences include those required to open a file, read a logical record, write a logical record, and close a file. A file must be opened before it can be read from or written on. When a file is opened, GEFRC performs the required label checking, medium positioning, and initialization of buffers and the file control block.

   The user may read logical records from input files and write logical records to output files. The GEFRC routines which perform these logical reads and writes also accomplish the necessary blocking, deblocking, and the physical record reading and writing in accordance with information in the file control block. As an option, the logical read and write requests may cause the logical records to be physically moved between the buffers and specified working storage locations. After a file has been processed, it must be closed. When a file is closed, the buffers are emptied and label processing and repositioning occur as specified by the calling sequence and the file control block.

2. Nondata Transmitting--The nondata transmitting calling sequences are provided to allow activities such as backspace file, backspace block, force end of reel, rewind, and write tape mark. These functions are actually the software equivalents of the hardware commands executed by GEFRC in such a manner as to maintain the logical record control of the file.

For ease of discussion, the GEFRC calling sequences have been divided into five logical groups. They are (1) File Preparation commands, (2) Logical Record Processing commands, (3) Device Positioning commands, (4) Physical Record Processing commands, and (5) Input/Output Editor functions.

Included with the description of each calling sequence are the exception conditions that may occur with the use of the command.

These exception conditions are listed in tabular form giving the reason for the error and the action taken by the system. In most cases the action is ABORT with a reason code. The reason code is a number between 1 and 7. The reason codes for each calling sequence are defined in the Error Procedures section at the end of this chapter.

Also mentioned in this chapter are file designator words. These words supply data to GEFRC for opening and closing files. File designator words are supplied by the programmer using the VFD (variable-field definition) pseudo-operation. A complete description of the file designator words may be found in Chapter 7.

FILE PREPARATION COMMANDS

Open File

Function: To initialize (open) files so that they may be properly accessed by other GEFRC functions.

Calling Sequence: CALL OPEN (list, number)

list = location of first of n consecutive file designator words for the n files that are to be initialized (opened).

number = n, the number of files to be opened.

Normal Return: Return is to the instruction which follows the CALL OPEN command. Each of the n files mentioned in the list will have been linked to each other and to all previously opened files via the file control block pointer field in word -1 of their file control blocks. The actions performed on each file will vary according to the conditions stated in its file designator word, the contents of its file control block, and the type of device which has been allocated for the current execution of the program.

If a file designated in the list is already open due to its presence in the list of a previous CALL OPEN command, then no action is performed for that file.

The OPEN subroutine calls upon GECOS via the MME GEFCON command to supply information concerning the device allocated to each currently unopened and unlocked file given in the list. GECOS will initialize the following fields in the file control block of those files:

1. Device type in word 0. Included is the bit which identifies the file as the SYSOUT file.
2. Physical device address in word -1.
3. File serial number in word -7 if the file is assigned to a non-SYSOUT magnetic tape.
4. Reel sequence number in word -8 if the file is assigned to a non-SYSOUT magnetic tape.
5. File present indicator in word -5. For files which are not present, no further action is performed by the OPEN subroutine.

GECOS obtains this data from file control cards supplied at execution time and relates this data to the proper file control block via the file code given on the control card and in word -4 of the file control block.

Each file is opened as an input or as an output file according to the specification given in the file designator word. For those files which are opened as input files with one or two buffers, the OPEN subroutine initiates reading of the first physical record into a buffer unless priming of the buffer is inhibited by bit 22 in the file designator word. In addition, it performs the following according to the type of device allocated:

Card Reader: Override label and block serial number bits in word -5 of the file control block by setting them for no labels and no block serial numbers.

Card Punch: Override label and block serial number bits in word -5 of the file control block by setting them for no labels and no block serial numbers. Set unit record device bit in word -5 of file control block so that records will not be blocked. Set record type field of word 0 of file control block to "fixed" so that record control words will not be created.

Perforated Tape Reader/Punch: No special action.

Printer: Override label and block serial number bits in word -5 of file control block by setting them for no labels and no block serial numbers. Set recording mode in word 0 of file control block for BCD. Set unit record device bit in word -5 of file control block so that records will not be blocked. Set record type field of word 0 of file control block to "fixed" so that record control words will not be created.

Magnetic Drum/Disc Storage Unit: Issue Rewind command if specified in file designator word. Set recording mode in word 0 of file control block to binary. Set block count in word -6 of file control block to zero. If the file is to be processed as a random file, abort the program if buffers are supplied. If the file is to be processed as a linked file, insure that the block size in word +4 of the file control block is not greater than 320 (if greater than 320, set value at 320). Also for linked files, abort the program if the record type in word 0 is not variable, if there are no buffers, or if the bit in word -5 indicates no block serial numbers.

Magnetic Tape: The actions performed for tape vary according to the data in the file designator word and the label bit in word -5 of the file control block. The following table shows the possible actions:

File Designator word says			FCB says						
Rewind	File #	Input/Output	Labeled	Rewind	Space Forward Files	Perform Label Routine	Set File Count (-6)	Set Block Count (-6)	Set Density as per FCB
No	0	output	--	No	0	OUTHX	Add 1	0	Yes
No	0	input	Yes	No	0	INHDX	Add 1	0	Yes
No	0	input	No	No	0	--	Add 1	0	Yes
Yes	0	output	--	Yes	0	OUTHX	1	0	Yes
Yes	0	input	Yes	Yes	0	INHDX	1	0	Yes
Yes	0	input	No	Yes	0	--	1	0	Yes
-	n	output	--	Yes	n-1	OUTHX	n	0	Yes
-	n	input	Yes	Yes	n-1	INHDX	n	0	Yes
-	n	input	No	Yes	n-1	--	n	0	Yes

SYSOUT: Insure that the block size in word +4 of the file control block is not greater than 320 (if greater than 320, set value at 320). Abort the program if record type in word 0 of file control block is not variable, if bit in word -5 indicates no block serial number, or if there are no buffers.

EXCEPTION CONDITIONS	ACTION
Illegal device for GEFRC (such as: typewriter)	Abort, Reason #1
File to be opened is designated as locked	Abort, Reason #2
The device is printer or punch, but the file is designated as an input file.	Abort, Reason #3
Illegal format for disc or drum (such as: buffers given for a random file)	Abort, Reason #4
Device is linked disc or drum, but record type is not variable	Abort, Reason #5
The file is assigned to SYSOUT, but its format is illegal (such as: no block serial numbers)	Abort, Reason #6
The file is required for this activity, but it is not present.	Abort, Reason #7

OPEN)

Alt pg 43

Close File

Function: To disconnect (close) files when no further activities are to be performed on them.

Calling Sequence: CALL CLOSE (list, number)

list = location of first of n consecutive file designator words for the n files that are to be disconnected (closed).

number = n, the number of files to be closed.

Normal Return: Return is to the instruction which follows the CALL CLOSE command. Each of the n files mentioned in the list will have been removed from the chain of currently open files (maintained in the file control block pointer field in word -1 of the file control blocks). Word -5 of their file control blocks will have been set to indicate that these files are closed, not present and that an end-of-file status does not exist on these files. No action is taken on files which were closed by a previous CALL CLOSE command.

Additional actions are performed on individual files according to data supplied in the file designator words, the contents of the file control blocks, and the type of device assigned. The CLOSE subroutine uses the input/output bit in word -5 of the file control block rather than the bit in the file designator word to determine whether a particular file is being used for input or for output.

For output files, the following actions are performed according to the device:

Non-SYSOUT linked disc/drum: Enter a single word end-of-file record (file mark = 17_8) after the last logical record of the file, and write the physical record from the current buffer.

Non-SYSOUT magnetic tape with buffers: Write the physical record from the current buffer, write an end-of-file on the tape (file mark = 17_8), and, if labeled, execute the OUTLX label routine.

Non-SYSOUT magnetic tape without buffers: Write an end-of-file on the tape (file mark = 17_8) and if labeled, execute the OUTLX label routine.

Other: If buffered, write the logical record from the current buffer.

The rewind and lock options given in the file designator words are honored according to device in the following ways:

REWIND AND LOCK

Non-SYSOUT disc/drum: Set bit for locked, and release device via MME GERELS

Non-SYSOUT multifile reel magnetic tape: Set bit for locked

Non-SYSOUT single-file reel magnetic tape: Set bit for locked, and release device via MME GERELS

SYSOUT file: No action

Any other: Set bit for locked, and release device via MME GERELS

REWIND WITHOUT LOCK

Non-SYSOUT disc/drum: Rewind

Non-SYSOUT magnetic tape: Rewind

Any other: No action

For buffered output files having the appropriate bit set in the file designator word, the CLOSE subroutine issues a CALL PUTSZ command with record size equal to zero prior to writing the physical record from the current buffer. However, the PUTSZ routine will be called only if PUTSZ has been requested elsewhere in the program.

EXCEPTION CONDITIONS	ACTIONS
End-of-file status on file designated as an output file	Abort, Reason #1
Unrecoverable I/O error and user error routine provided via word -5 in file control block	Error routine, codes 4, 5
Unrecoverable I/O error and no user error routine provided	Abort, Reason #2
File to be closed is not in chain of currently open files	Abort, Reason #3
Illegal status received from I/O on disc or drum	Abort, Reason #4

Please note: If a program terminates abnormally with files still open, it is the function of the ERTN (TERM) modules of GECOS to determine if any abort subroutine is to be performed on any of the files (if so, it will load those files open at the time of the program) and to close all files still open. (TERM)/ERTN will disposition all closed files per the disposition code designated on the respective file control cards.

Set as Input File

Function: To set a currently open file to an input file regardless of its current mode.

Calling Sequence: CALL SETIN (fcb)

fcb = symbolic location of the file control block.

Normal Return: Return is to the instruction which follows the CALL SETIN command. The actions performed depend upon the current state of the file. If the file is already an input file, no action is taken. If the file is an output file without buffers, then the input/output bit in word -5 of fcb is changed to indicate input.

If the file is an output file with buffers, then the input/output bit in word -5 of fcb is changed to indicate input. In addition, reading of the next physical record into the buffer is initiated.

Note that an output file must be backspaced (with REWND, BSREC, or BSTFM) between the last Write command and the CALL SETIN command.

Set as Output File

Function: To set a currently open file to an output file regardless of its current mode.

Calling Sequence: CALL SETOUT (fcb)

fcb = symbolic location of the file control block.

Normal Return: Return is to the instruction which follows the CALL SETOUT command. The actions performed depend upon the current state of the file. If the file is already an output file, no action is taken.

If the file is an input file without buffers, then the input/output bit in word -5 of fcb is changed to indicate output and any end-of-file indications are removed.

In addition to the above, if an input file has buffers the file and all its pointers are set to rewrite the next logical record.

LOGICAL RECORD PROCESSING COMMANDS

Read Logical Record

Function: To obtain the next logical input record from a designated input file.

Calling Sequence: CALL GET (fcb, eof, stor)

fcb = symbolic location of the file control block.

eof = symbolic location of user's end-of-file routine.

stor = symbolic location of the first cell of a working storage area into which the record is to be moved (this field may be omitted).

Normal Return: The normal return is to the instruction which follows the CALL GET. The following fields will have been set in the file control block:

1. The current record index (word 0 of fcb) contains the location of the first data word of the requested logical record as it resides in the buffer. Even though the record may have been moved to working storage (stor), this value will still point to its position in the buffer.

2. The record size field (word +1 of fcb) will contain the number of data words in the requested logical record.

End-of-file Return: Whenever an end-of-file condition is encountered on the input device, return will be to the location given as eof in the calling sequence. The file mark character will be in bits 6-11 of word -3 of the file control block. If after receiving an end-of-file indication a subsequent CALL GET command is issued, then the resulting action will be shown in the table below.

Device Type	File Mark	Labeled File	Multi or Single File	Action
ANY	ANY OTHER $\neq 17_8$ THAN	--	--	Obtain next logical record
Linked DSU/drum	17_8	--	--	Repeat end-of-file exit
Magnetic Tape	17_8	yes	--	Repeat end-of-file exit
Magnetic Tape	17_8	no	~~single~~ MULTI	Perform unit switch and obtain first logical record on the new tape
Magnetic Tape	17_8	no	~~multi~~ SINGLE	Obtain first logical record of the next file

EXCEPTION CONDITIONS	ACTION (Cont'd)
File was opened as an output file	Abort, Reason #1 GET 1
File is not present	EOF Return
Illegal conditions in file control block (such as: no buffers)	Abort, Reason #2 GET 2
Unrecoverable I/O error and user error routine provided via word -5 in file control block	User error routine, codes 2, 3, 6 (Go p. 1/2)
Unrecoverable I/O error and no user error routine provided	Abort, Reason #3 GET 3
Block serial number in input physical record does not agree with block count in file control block and user error routine has been provided	User error routine code 1 (Go p. 1/2)
Block serial number in input physical record does not agree with block count in file control block and user error routine has not been provided	Abort, Reason #4 GET 4
Record type is "fixed" or "mixed" but the record size field in the file control block contains zero	Abort, Reason #5 GET 5

Read Logical Record from Next Block

Function: To obtain the first logical record in the next physical record from a designated input file.

Calling Sequence: CALL GETBK (fcb, eof, stor)

fcb = symbolic location of the file control block.

eof = symbolic location of user's end-of-file routine.

stor = symbolic location of the first cell of a working storage area into which the record is to be moved (this field may be omitted).

The CALL GETBK command performs the same functions as CALL GET except that all logical records remaining in the last accessed physical record are ignored. See description of GET for return and exception condition information.

CALL GETBK(fcb, eof) will cause next physical record (block) to be made available to the program without transmission of the first logical record.

Write Logical Record

Function: To allocate space within a buffer of the designated output file for inserting the next logical record of that file and, if desired, move that logical record to the allocated area.

Calling Sequence: CALL PUT (fcb, stor)

fcb = symbolic location of the file control block.

stor = symbolic location of the first data word of the logical record as it resides in working storage. This field may be omitted, but when given, it indicates that the record is to be moved by GEFRC to the allocated buffer area.

The record size field (word +1 of fcb) must contain the number of data words in the logical record prior to execution of the CALL PUT command. The value in this field is used to indicate the amount of space to allocate and the number of words to move when that option is employed. The value of record size is not modified by GEFRC; therefore, it is not necessary to reinitialize this value for successive equal-length records.

Normal Return: Return is to the instruction which follows the CALL PUT. The current record index field (word 0 of fcb) will contain the location in the buffer where the first data word of the logical record does or will reside.

EXCEPTION CONDITIONS	ACTION
Status on designated file is end of file	Abort, Reason #1 PUT 1
Unrecoverable I/O error and user error routine provided via word -5 in file control block	User error routines, codes 4, 5
Unrecoverable I/O error and no user error routine provided	Abort, Reason #2 PUT 2
Illegal conditions in file control block (such as: no buffers)	Abort, Reason #3 PUT 3
Current logical record as defined by the record size field in the file control block is larger than the buffer	Abort, Reason #4 PUT 4

Write Logical Record in Next Block

Function: To allocate space at the beginning of a buffer of the designated output file for inserting the next logical record of that file, and if desired, move that logical record to the allocated area

Calling Sequence: CALL PUTBK (fcb, stor)

fcb = symbolic location of the file control block

stor = symbolic location of the first data word of the logical record as it resides in working storage. This field may be omitted, but when given, it indicates that the record is to be moved by GEFRC to the allocated buffer area.

The CALL PUTBK command performs the same function as CALL PUT except that the referenced logical record will be the first logical record in the next physical record. This implies that the physical record which had been under construction in the buffer may be shorter than the usual physical record created for this file. See description of PUT for record size, return and exception condition information.

Write Logical Record from Input Buffer

Function: To move the last accessed logical input record from the designated input file to the next available position in the designated output file.

Calling Sequence: CALL COPY (fcb-out, fcb-in)

fcb-out = symbolic location of the file control block of the output file.

fcb-in = symbolic location of the file control block of the input file.

The CALL COPY command performs the same function as CALL PUT except that the current record index of the input file (word 0 of fcb-in) is used in place of the usual working storage location (stor in CALL PUT). The size of the record is determined by the record size field of the output file control block (word +1 of fcb-out). The input file control block (fcb-in) is not modified or checked in any manner. See description of PUT for return and exception condition information.

Update File Control Block

Function: To update the file control block of the designated output file to reflect the true size of the last logical record placed in that file.

Calling Sequence: CALL PUTSZ (fcb, size)

fcb = symbolic location of the file control block.

size = symbolic location of a word whose contents in bits 0-17 provide the number of words in the last logical output record to this file.

This command is generally used in the case where an output record of unknown length is to be constructed in the buffer. Either CALL PUT or CALL PUTBK is issued with the record size field of the file control block (word +1 of fcb) set to some maximum record size value. Space for this maximum size record is thus reserved. After the record has been constructed and its actual length determined, the CALL PUTSZ command is issued to update the file control block with the appropriate pointers.

The original value of record size in the file control block (word +1 of fcb) is not modified by the CALL PUTSZ command. It will retain the maximum value placed there prior to execution of the CALL PUT or CALL PUTBK command. If the new record size indicated by the value in size is zero, then the last request via CALL PUT or CALL PUTBK is effectively deleted. Note that successive CALL PUTSZ commands must not be given without intervening CALL PUT or CALL PUTBK commands.

EXCEPTION CONDITIONS	ACTION
Illegal conditions in file control block (such as: no buffers)	Abort, Reason #1
New record size is larger than the original request via PUT or PUTBK	Abort, Reason #2

Release Current Buffer

Function: To cause the next referenced logical record of the designated file to be the first logical record of the next physical record.

Calling Sequence: CALL RELSE (fcb)

fcb = symbolic location of the file control block.

Normal Return: The normal return is to the instruction which follows the CALL RELSE. If the file designated by fcb is an input file, then all, if any, logical records remaining in the current physical record will be ignored. The next logical record request will obtain the first logical record in the next physical record.

If the file designated by fcb is an output file, then the physical record currently under construction in a buffer of that file will be written to the assigned device. This physical record may be shorter than the usual record created for this file. The next logical record on this file will begin a new physical record.

DEVICE POSITIONING COMMANDS

Force End of Reel

Function: To force an end-of-reel condition on a magnetic tape file.

Calling Sequence: CALL FORCE (fcb)

fcb = symbolic location of the file control block.

Normal Return: Return is to the instruction which follows the CALL FORCE command. The actions performed depend upon the description of the file. If the file is an output file, an end of file (file mark = 17_8) will be written on the current tape. If buffered, the physical record under construction in the buffer will be written prior to writing the file mark. If labeled, the trailer label will be written on the current tape. Unit switching will then be performed. The header label on the new tape will be checked for an expired retention period and the new label, if so indicated, will be written.

If the file is an input file, unit switching will be performed. If labeled, the header label on the new tape will be checked.

EXCEPTION CONDITIONS	ACTION
Illegal request for the FORCE routine (such as: the device is not magnetic tape or the file is not open)	Abort, Reason #1

Forward Space to File Mark

Function: To forward space an unlabeled multifile magnetic tape file to a position immediately following the n^{th} succeeding standard end of file (file mark = 17_8).

Calling Sequence: CALL FSTFM (fcb, n)

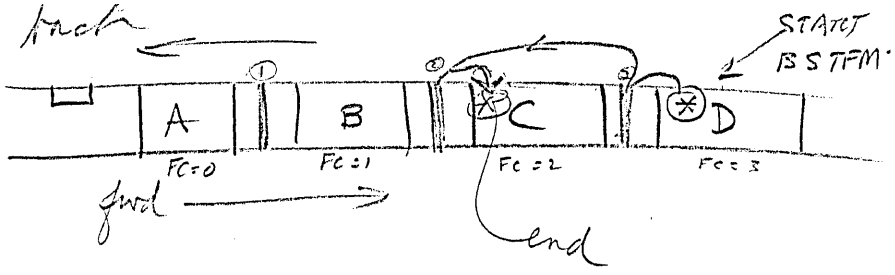
fcb = symbolic location of the file control block.

n = number of standard file marks to bypass.

Normal Return: Return is to the instruction which follows the CALL FSTFM command. The tape will be positioned immediately after the n^{th} standard file mark which followed the initial position of the tape. The next input/output request for this file will reference the first physical record which follows the designated file mark. The block count is reset to zero, the file count is incremented by n-1 in word -6, and the EOF bit is set in word -5 of the file control block.

EXCEPTION CONDITIONS	ACTION
Illegal request for FSTFM routine (such as: file is labeled, single-file reel, not open, or not magnetic tape)	Abort, Reason #1
The request exceeded the limit of the reel--n is greater than the number of files remaining	Abort, by GEPR

Backspace to File Mark



Backspace ~~to~~ over File Mark

Function: To backspace an unlabeled magnetic tape file to a position immediately following the n^{th} preceding standard end of file (file mark = 17_8).

Calling Sequence: CALL BSTFM (fcb, n)

fcb = symbolic location of the file control block.

n = number of standard file marks to bypass.

Normal Return: Return is to the instruction which follows the CALL BSTFM command. The tape will be positioned immediately after the n^{th} standard file mark which preceded the initial position of the tape. The next input/output request for this file will reference the first physical record which follows the designated file mark in a forward direction. The block count is reset to zero and the file count is decremented by n-1 in word -6 of the file control block.

If this command results in reaching the magnetic tape leader, a normal return is made if this event occurred when backspacing the n^{th} file. If it occurs prior to the n^{th} file, the program is aborted.

EXCEPTION CONDITIONS	ACTION
Illegal request for BSTFM routine (such as: file is labeled, not open, or not magnetic tape)	Abort, Reason #1
The load point on the tape was reached before n files have been spaced	Abort, Reason #2

Forward Space n number of Blocks (Physical Records)

Function: To space over the next n physical records on the designated magnetic tape or linked disc/drum file in a forward direction.

Calling Sequence: CALL FSREC (fcb, n, eof)

fcb = symbolic location of the file control block.

n = number of physical records to bypass ($n \leq 63_{10}$).

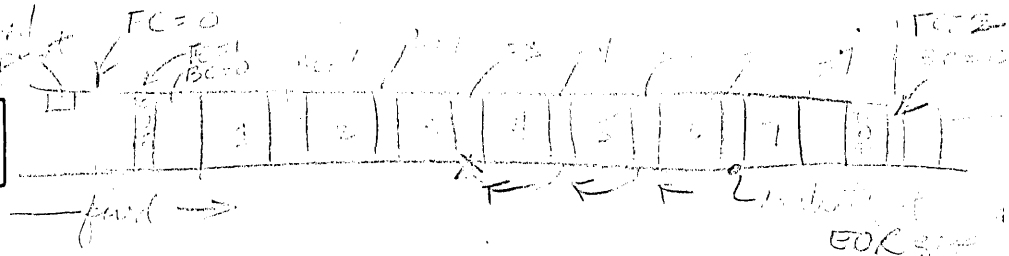
eof = symbolic location of user's end-of-file routine.

Normal Return: The normal return is to the instruction which follows the CALL FSREC command. The file is positioned immediately after the n^{th} physical record which follows the initial position of the file. For a buffered file, if the last command issued for that file referenced a logical record, then the initial position of the file is assumed to be immediately after the physical record that contained that logical record. The block count in word -6 of the file control block is increased by n.

End-of-file Return: If a file mark (any single character record) is encountered before n physical records have been bypassed, then return is to the location given as eof in the calling sequence. The file mark character is in bits 6-11 of word -3 of the file control block, and bits 30-35 of that word give the number of records not spaced. For example, if n = 5 and the third record is a file mark, then the value in bits 30-35 is 2 (5-3 = 2). The block count in word -6 of the file control block is increased by the number of records skipped (3 in our example) unless the file mark is the standard file mark (17_8). In that case the block count is reset to zero and the file count is increased by one in word -6 of the file control block.

EXCEPTION CONDITIONS	ACTION
The file is not present	Abort, Reason #1
A previous end-of-file condition exists on the file such that forward spacing is not possible	Abort, Reason #2
File is closed, locked or not present	Abort, Reason #4
Status returned from request to space file was not end of file or channel ready, and no user error routine provided via word -5 in file control block	Abort, Reason #5
Device is not magnetic tape, linked disc or drum	Abort, Reason #6
Status returned from request to space file was not end of file, channel ready, or blank tape on read	Abort, Reason #7
Status returned from request to space tape was blank tape on read and user error routine provided via word -5 in file control block	User error routine, code 6

Backspace n number of Blocks (Physical Records)



Backspace n number of Blocks (Physical Records)

Function: To space over the n last accessed physical records on the designated magnetic tape or linked disc/drum file in a backward direction.

Calling Sequence: CALL BSREC (fcb, n, eof)

fcb = symbolic location of the file control block.

n = number of physical records to bypass ($n \leq 63_{10}$).

eof = symbolic location of the user's end-of-file routine.

Normal Return: The normal return is to the instruction which follows the CALL BSREC command. The file is positioned immediately ahead of the n^{th} physical record which preceded the initial position of the file. For a buffered file, if the last command issued for that file referenced a logical record, then the initial position of the tape is assumed to be immediately after the physical record that contained that logical record. The block count in word -6 of the file control block is reduced by n.

End-of-file Return: If a file mark (any single character record) is encountered before n physical records have been bypassed, then return is to the location given as eof in the calling sequence. The file mark character is in bits 6-11 of word -3 of the file control block, and bits 30-35 of that word give the number of records not spaced. For example, if n = 5 and the third record is a file mark, then the value in 30-35 is 2(5-3 = 2). The block count in word -6 of the file control block is reduced by the number of records skipped (3 in our example) unless the file mark is the standard file mark (17_8). In that case the file count is reduced by one and the block count is incorrectly set to zero. (Additional backspacing will not further reduce the zero block count.) Note that a request to read following this end-of-file return will also result in an end-of-file condition.

EXCEPTION CONDITIONS	ACTION
Illegal request for BSREC command (such as: device is not tape, linked disc or drum, file is closed)	Abort, Reason #1
Status returned from request to space tape was not end of file, channel ready, or tape on load point	Abort, Reason #2
Status returned from request to space tape was tape on load point and user error routine provided via word -5 in file control block	User error routine code 7

Rewind

Function: To rewind a file whose device is unlabeled magnetic tape, linked magnetic drum or linked disc storage unit.

Calling Sequence: CALL REWND (fcb)

fcb = symbolic location of the file control block.

Normal Return: Return is to the instruction which follows the CALL REWND command. If the file is recorded on more than one reel of magnetic tape, then only the current reel is rewound. If the file is recorded on a multifile reel of tape, the entire reel is rewound.

EXCEPTION CONDITIONS	ACTION
Illegal request for REWND routine (such as: labeled tape, illegal device, or file closed)	Abort, Reason #1

Write End-of-File

Function: To write a file mark on an output file whose device is magnetic tape, linked magnetic drum or linked disc storage unit. Note: The WEF routine may not be used when outputting to Tape-SYSOUT.

Calling Sequence: CALL WEF (fcb, file mark)

fcb = symbolic location of the file control block.

file mark = symbolic location of word whose bits 30-35 contain the file mark to be written.

Normal Return: Return is to the instruction which follows the CALL WEF command. The actions performed depend upon the device assigned to the file. If the device is linked drum/disc, the file mark is inserted into the current buffer following the last logical record. Since files on these media conform to standard system format, the file mark becomes a special case of the record control word. The record size field (bits 0-17) is zero and the file mark occupies bits 18-23 of that word. If the device is magnetic tape and buffers are used, then the physical record under construction in the current buffer is written prior to writing the file mark.

There are several restrictions on the choice of the file mark character when the device is magnetic tape. They are:

1. The file mark cannot be zero or all ones (77_8).
2. A standard file mark (17_8) may not be written on labeled tapes.

EXCEPTION CONDITIONS	ACTION
An end-of-file status exists on this file	Abort, Reason #1
Unrecoverable I/O error or illegal status for these devices	Abort, Reasons #2 and #4
Illegal request for WEF routine (such as: not tape or drum/disc, not open, or illegal file mark)	Abort, Reason #3

PHYSICAL RECORD PROCESSING COMMANDS

Read Physical Record

Function: To initiate reading of a physical record from the designated file under user control.

Calling Sequence: CALL READ (fcb, cont, cc)

fcb = symbolic location of the file control block.

cont = symbolic location of the first control word for reading the record.

cc = symbolic location of a "courtesy call" routine supplied by the user (this field may be omitted).

The control words specified by cont must be the data control words (DCW's) describing where the data in the record is to be stored. In addition, for files recorded on random drum/disc, the first word must contain the relative block address of the record to be read. For example, to read a 200-word record with 100 words into each of the areas A and B would require:

From block 25 of drum/disc

```
cont    DEC   25
        IOTP  A,100
        IOTD  B,100
```

From any other device

```
cont    IOTP  A,100
        IOTD  B,100
```

These words must not be modified until the transmission is known to be complete.

Normal Return: Return is to the instruction which follows the CALL READ command. GEFRC will have requested IOS to read the record and the status return words (-3 and -2 of fcb) will contain zero until the request is satisfied (correct or not).

Special action is required to ascertain when the requested record has been read and whether or not it was read correctly. The usual, and preferred, method is to proceed as far as possible without knowing or caring about the status of the request. At that point in the program beyond which it is impossible to proceed without the requested record, a CALL WAIT command is issued for the same file. A normal return from the WAIT routine guarantees proper termination of the transmission.

If a courtesy call is specified (by cc entry), then the routine located at cc will be executed on termination of the read request. Knowledge of this execution (when it is done or the fact that it has been done) is not imparted to the main program unless the courtesy call routine itself specifically sets some signal that the main program can test. The following rules for courtesy call routines must be observed:

1. Return must be accomplished by the MME GEENDC instruction.
2. Execution time must not exceed 200 or 400 microseconds on the GE-635 or GE-625, respectively. Time spent by IOS in issuing another I/O command need not be considered in this limit.

3. Master mode entries (MME) to the following routines are not permitted: GEBORT, GECALL, GECHEK, GEFCON, GEFILS, GEFINI, GERELC, GEROAD, GEROLL, GERSTR, GESAVE.

4. The WAIT routine may not be requested within the courtesy call. In the case of an end-of-file or an error procedure the time limits would be exceeded.

If a courtesy call results in an additional I/O request on the same file, then the main program cannot use the WAIT routine to determine the accurate completion of the request (original or within courtesy call).

EXCEPTION CONDITIONS	ACTION
Illegal request for this command (such as: file is not open; file has buffers)	Abort, Reason #1

Write Physical Record

Function: To initiate the writing of a physical record on the designated file under user control.

Calling Sequence: CALL WRITE (fcb, cont, cc)

fcb = symbolic location of the file control block.

cont = symbolic location of the first control word for writing the record.

cc = symbolic location of a "courtesy call" routine supplied by the user (this field may be omitted).

The control words specified by cont must be the data control words (DCW's) describing the location of the data to be written. In addition, for files recorded on random drum/disc, the first word must contain the relative block address of the record to be written. For example, to write a 200-word record with 100 words from each of the areas A and B would require:

To block 25 of drum/disc

```
cont    DEC    25
        IOTP   A,100
        IOTD   B,100
```

To any other device

```
cont    IOTP   A,100
        IOTD   B,100
```

These words and the data area must not be modified until the transmission is known to be complete.

Normal Return: Return is to the instruction which follows the CALL WRITE command. GEFRC will have requested IOS to write the record, and the status return words (-3 and -2 of fcb) will contain zero until the request is satisfied (correct or not).

Special action is required to ascertain when the requested record has been written and whether or not it was written correctly. The usual, and preferred, method is to proceed as far as possible without knowing or caring about the status of the request. At that point in the program beyond which it is impossible to proceed without completion of the request, a CALL WAIT command is issued for the same file. A normal return from the WAIT routine guarantees proper termination of the transmission.

If a courtesy call is specified (by cc entry), then the routine located at cc will be executed on termination of the WRITE request. Knowledge of this execution (when it is done or the fact that it has been done) is not imparted to the main program unless the courtesy call routine itself specifically sets some signal that the main program can test. The following rules for courtesy call routines must be observed:

1. Return must be accomplished by the MME GEENDC instruction.
2. Execution time must not exceed 200 or 400 microseconds on the GE-635 or GE-625, respectively. Time spent by IOS in issuing another I/O command need not be considered in this limit.

3. Master mode entries (MME) to the following routines are not permitted: GEBORT, GECALL, GECHEK, GEFCON, GEFILS, GEFINI, GERELC, GEROAD, GEROLL, GESTR, GESAVE.

4. The WAIT routine may not be requested in the courtesy call. In the case of an error procedure the time limits would be exceeded.

If a courtesy call results in an additional I/O request on the same file, then the main program cannot use the WAIT routine to determine the accurate completion of the request (original or within courtesy call).

EXCEPTION CONDITIONS	ACTION
File is not present	Abort, Reason #1
An end-of-file status exists on this file	Abort, Reason #2
Illegal request for this command (such as: file has buffers; file is not open)	Abort, Reason #3

WRITE

Wait for Physical Record

Function: To delay processing until the satisfactory completion of the last requested I/O activity on the designated file.

Calling Sequence: CALL WAIT (fcb, eof)

fcb = symbolic location of the file control block.

eof = symbolic location of user's end-of-file routine.

Normal Return: The normal return is to the instruction which follows the CALL WAIT command. This will occur only when the status for the designated file is effectively "channel ready" or the user has been informed of an irregularity (via user error routine) which he chose to ignore.

End-of-file Return: If an end-of-file condition is encountered on the device for the designated file, return will be to the location given as eof in the calling sequence. The file mark character will be in bits 6-11 of word -3 of the file control block.

Note that if the device is "random disc" and if the READ or WRITE request accessed the "last consecutive block" (a relative block address whose last five bits are 11111), then an end-of-file condition (file mark = 00) results. Transmission beyond this "last consecutive block" will not have been performed. For example, if the relative block address is 31_{10} (37_8) and if 35 words are to be transmitted, then transmission will be complete but an end-of-file condition will be given. But if the address is again 31_{10} and 50 words are to be transmitted, then the end-of-file condition will be given with only 40 words transmitted (the number of words before the end of the "last consecutive block"). It is the user's responsibility to determine whether or not his requested transmission was completed.

Note: The WAIT routine must not be used within a courtesy call. If it should require extensive testing of the status returned by IOS, then the time allowed for a courtesy call could be exceeded. Also, the WAIT routine must not be used if the READ or WRITE asked for a courtesy call which also requested I/O on the same file. The reason is that there is no guarantee which request the WAIT routine will handle--the original or the one in the courtesy call.

EXCEPTION CONDITIONS	ACTION
Unrecoverable I/O error and user error routine provided via word -5 in file control block	User error routine, Codes 2,3,4,5,6
Unrecoverable I/O error and no user error routine provided	Abort, Reason #1
End-of-file condition but no eof entry in calling sequence	Abort, Reason #1

WAIT

INPUT/OUTPUT EDITOR FUNCTIONS

The calling sequences in this section are specifically designed to provide certain special purpose requirements of the language processors. These include providing a limited output formatting capability for both printed and punched output. It includes the ability to convert input from COMDEK to Hollerith format, to merge an ALTER file with the primary source language input and to create an updated COMDEK output file from the merged input. In addition, the output routines included here provide an accurate interface with the standard output collector--SYSOUT.

Input/Output Editor Initialization

Function: To initialize the edit functions such as PRINT and PUNCH with parameters which do not usually vary with each call for these functions.

Calling Sequence: CALL IOEDIT (list, n)

list = location of the first entry in the list of control parameters.

n = the number of entries in the list.

An entry in the list has the form

ZERO control, code

control = location of the parameter identified by the code.

code = a value expressing the meaning of the parameter.

The permissible entries are:

Code	Control
1	Location of the first of ten consecutive words to be used as the first heading line on each page of output produced by the PRINT routine.
2	Location of the first of twenty consecutive words to be used as the second heading line on each page of output produced by the PRINT routine. If not given, there will be no subheading line.
3	Location of the first of two consecutive words whose first eight characters are to be used as the label in columns 73-80 of the card deck produced by the PUNCH routine. If the last i characters of this eight character field are numeric, then the cards are sequence numbered in the last i columns beginning with the value given in those i characters.

For example:

Parameter	1st Card	2nd Card	3rd Card
(4) A000	(4) A000	(4) A001	(4) A002
ABCD9999	ABCD9999	ABCD0000	ABCD0001
LABEL (3)	LABEL (3)	LABEL (3)	LABEL (3)

Code	Control
4	Location of the first of two consecutive words whose first eight characters are to be used as the label in columns 73-80 of the card deck produced by the COMDEK routine. Rules for sequencing are as stated above for code 3.
5	Location of a word which contains a number (right adjusted) expressed in BCD to be used as the first page number on the report produced by the PRINT routine. Succeeding pages will be numbered consecutively from this value. If not given, the value one is assumed.
6	Location of the file control block for a file (usually P*) on which incorrect ALTER cards will be recorded in standard system format for subsequent printing. The report code used is that of the execution report. If not given, ALTER errors will not be reported.
7	Location of a word which contains a value (right justified) to be used as the report code for the output produced by the PRINT routine. If not given, a report code of 74_8 (execution report) is assumed.

The IOEDIT routine is normally called only once within a program to initialize the necessary parameters. However, it may also be used whenever a change is desired in any of the parameters. For example, if a new second heading line is desired, then IOEDIT would be called with a code 2 parameter.

The values supplied to the edit routines via the IOEDIT parameters are never changed by these routines. For example, given an entry ZERO ABC,3 where ABC and ABC+1 contain (4) A000 (4), the contents of ABC and ABC+1 will not change as the card sequence number progresses from A000 to A001 to A002, etc. However, if the user changes the contents of ABC and ABC+1, the sequencing will proceed from the new value found therein even though a new IOEDIT entry of code 3 was not made.

If one or more of the n entries in the list do not contain a legal code, then such entries will be ignored.

Read Input Record

Function: To obtain the next logical input record from the designated file (with decompression from COMDEK format, if necessary) or from an Alter file of changes to the designated file; and, if required, to compress this logical record into the COMDEK format and insert it into the file designated as K*.

Calling Sequence: CALL RDREC (fcb, eof)

fcb = symbolic location of the file control block.

eof = symbolic location of the user's end-of-file routine.

Normal Return: The normal return is to the instruction which follows the CALL RDREC. The only meaningful field in the file control block is the current record index (word 0 of fcb). It will contain the location of the first data word of the requested logical record. There are three possible locations for this record:

1. In the buffer of the designated file if it is recorded as a Hollerith image.
2. In an internal GEFRC working area if it is recorded as a compressed image in COMDEK format--the record in the working area will have been decompressed.
3. In the buffer of the Alter file (A*) if appropriate.

Note: The only permissible binary card image handled by the RDREC routine is the COMDEK binary card.

End-of-File Return: If an end-of-file condition is encountered on the designated file (by fcb), return is to the location given as eof in the calling sequence. If the device is magnetic tape, then this exit is taken only if the file mark is 17_8.

EXCEPTION CONDITIONS	ACTION
A binary card record in the input file is not a COMDEK card	Abort, Reason #1
Sequence number on incoming COMDEK card record is incorrect	Abort, Reason #2
Error occurred in decompressing a COMDEK card	Abort, Reason #3

Note: In performing its numerous functions, RDREC uses the GET, PUT, OPEN and CLOSE routines. Therefore, exception conditions reported by these routines may result from a CALL RDREC command.

Write Output Record

Function: To insert a logical record in the next available position in the designated output file if the record is to be a printed line or a punched card.

Calling Sequence: CALL WTREC (fcb, image, media, report)

fcb = symbolic location of the file control block.

image = symbolic location of the first word of the logical record (line or card) to be inserted.

media = symbolic location of a word containing a code in bits 30-35 that describes the ultimate form of the record. The possible codes are:

1. Column binary card
2. Hollerith card
3. Print line

report = symbolic location of a word containing a report code in bits 30-35.

In addition to the data supplied to the calling sequence, the record size field in the file control block (word+1 of fcb) must contain the number of data words in the record to be inserted.

If the ultimate form of the record is to be a printed line (media code 3), then printing will be performed on the on-line printer in the edit mode wherein the control characters are embedded in the data record itself. Thus each line must contain its own slew controls and they must be inserted by the user prior to the CALL WTREC command.

The function of the WTREC routine is identical to the PUT routine except that data supplied by media and report parameters are inserted into the record control word when the record type is "variable." These codes are required for the standard output media conversion (SYSOUT) which is capable of accepting and subsequently sorting out up to eight reports for a single activity. Thus it uses the media data to select the proper device and the report data to sort these records into discrete reports.

EXCEPTION CONDITIONS
The PUT routine is used in performing the WTREC function and thus the exception conditions reported by the PUT routine may result from a CALL WTREC command.

Write Print Line Image

Function: To insert a line into the one current printed report whose pages are automatically

1. Titled and subtitled
2. Numbered
3. Controlled by an internal line counter.

Calling Sequence: CALL PRINT (fcb, image, slew)

fcb = symbolic location of the file control block for the file containing this report.

image = symbolic location of the first word of the print line image to be inserted.

slew = symbolic location of a word whose contents in bits 30-35 will control the movement of the paper. If this value is in the range of 0 to 15, then the paper will be moved the indicated number of lines (0 to 15) after printing the line. If this value is greater than 15, then the paper will be ejected to the top of page and headings will be printed before printing the line. Such a request results in automatically spacing one line after printing.

In addition to the data supplied in the calling sequence, the record size field of the file control block (word +1 of fcb) must contain the number of data words in the print line image. Upon return from the PRINT routine this value will be unchanged. (It may be changed temporarily during execution of the PRINT routine but it will be restored before returning to the user program.)

The actual printing of the report produced by PRINT on the on-line printer will be performed in the edit mode wherein the control characters are embedded in the data record itself. The PRINT routine inserts the slew codes at the end of the image according to the request in slew. Their position in the image depends upon the user's value of record size in the file control block. The possibilities are:

1. If record size is 23, then the slew characters replace the last two characters in word 23.
2. If record size is n where n is less than 23, then the record size becomes n+1 temporarily and the slew characters are the first two characters of word n+1.
3. If record size is greater than 23, then only the first 22 words of the line image will be used and the slew characters are the first two characters of word 23.

The report produced by the PRINT routine is automatically assigned the report code of 74_8. In the event that the file is assigned to SYSOUT, this has the same report code as the execution report produced by GECOS. The 74_8 report code may be overridden by an IOEDIT parameter (Code 7).

Each page of the report produced by PRINT contains a maximum of 54 lines. The page format is as follows:

Line 1 - The first heading line has the following nonblank print positions:

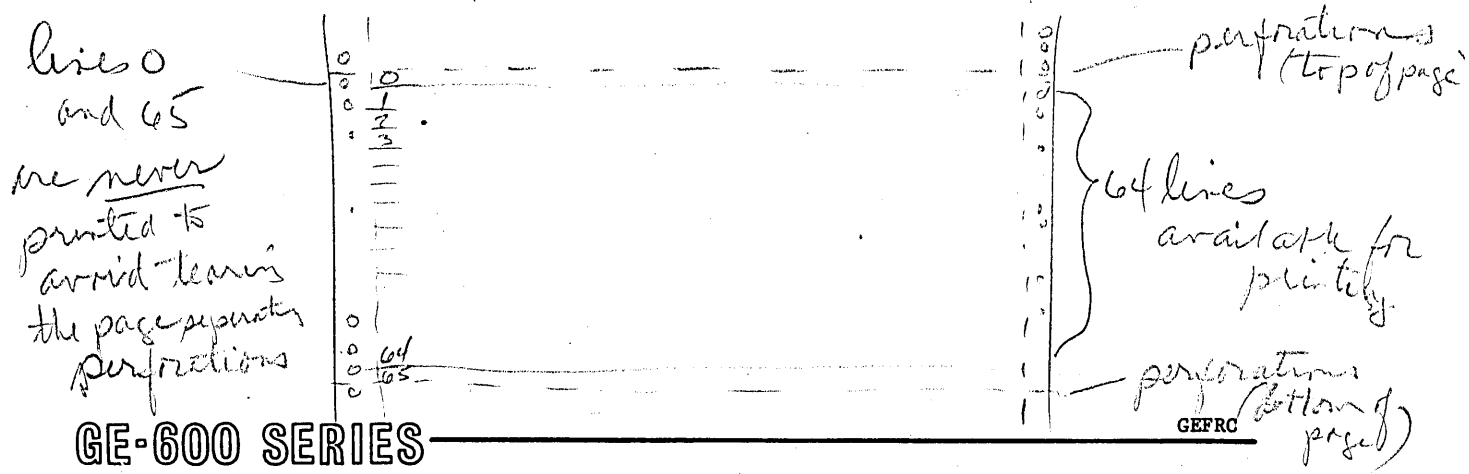
1-5	Contents of Job Identifier field of $ SNUMB control card.
7-8	Activity number within the job.
11-18	Current date; for 4 July 1776, this field would contain 07-04-76.
31-90	The 60 characters of the first heading line as provided by a Code 1 parameter to IOEDIT. If not supplied, this field is blank.
104-107	The word "PAGE."
110-114	The sequential page number right justified with leading blanks. An initial page number other than one may be supplied by a Code 5 parameter to IOEDIT.

Line 2 Blank

Line 3 The 120 characters of the second heading line as provided by a Code 2 parameter to IOEDIT. If a Code 2 parameter is not given, then this line is the first data line on the page.

Line 4 Blank, if line 3 contains a second heading line. Otherwise, its form and content is under user control.

EXCEPTION CONDITIONS
The Print routine uses the WTREC routine which in turn calls PUT. Thus, exception conditions reported by the PUT routine may result from a CALL PRINT command.



Edit and Write Print Line Image

Edit and Write Print Line Image

Function: To edit a print line image prior to writing via the PRINT routine.

Calling Sequence: CALL EPRINT (fcb, image, slew, n)

fcb = symbolic location of the file control block for the file containing the report.

image = symbolic location of the first word of the print line image to be inserted.

slew = symbolic location of a word whose contents in bits 30-35 will control the movement of the paper. If this value is in the range of 0 to 15, then the paper will be moved the indicated number of lines (0-15) after printing the line. If this value is greater than 15, then the paper will be ejected to the top of page and headings will be printed before printing the line. Such a request results in automatically spacing one line after printing.

n = the number of initial words in image which do not require editing. (This field may be omitted.)

All records destined for the on-line printer are eventually printed in the edit mode of the printer. In this mode, the character 17_8 (?) is an ignore character wherein the character and its space are deleted. Also, the character 77_8 (!) is a control character which causes either character spacing or line slewing.

The EPRINT routine scans the line image located by image looking for 17_8 and 77_8 characters in the last m-n words, where m equals the record size given in word +1 of fcb. If any of these characters are found, they will be preceded by two 77_8 characters which then allow the ? and ! to be printed. The new image is built in a special line image buffer within the EPRINT routine. EPRINT will CALL PRINT when the line has been edited and is ready for printing.

The original unedited line (in image) will be truncated to 132 characters (22 words) if the record size (m) is greater than 22. Also, if the slew code indicates a slew to the top of page before printing, then the line will be truncated to an edited length of 132 characters.

Write Punch Card Image

Function: To insert a punched card image in the next available position in the designated output file.

Calling Sequence: CALL PUNCH (fcb, image, kind)

fcb = symbolic location of the file control block.

image = symbolic location of the first word of the card image to be inserted.

kind = symbolic location of a word whose content designates the type of card to be produced--if zero, column binary card; if not zero, Hollerith card.

The length of the record inserted is determined by the value in the word referenced by kind and not by the value in the record size field of the file control block. Upon return from the PUNCH routine, word +1 of fcb (record size) will be identical to its value on entry.

Column binary card records (kind points to zero word) will contain 24 words (image through image +23). If they are defined as variable record type, the report code is 76 8 and the media code is ~~one~~. 01 (8)

Hollerith card records (kind points to nonzero word) will contain 12 words (image through image +11). If they are defined as variable record type, the report code is 76 8 and the media code is ~~two~~. 02 (8)

The card deck produced will be labeled and/or sequenced numbered in columns 73-80 if a code 3 parameter was previously provided via the IOEDIT routine.

EXCEPTION CONDITIONS
The PUNCH routine uses the WTREC routine which in turn calls PUT. Thus exception conditions reported by the PUT routine may result from a CALL PUNCH command

The calls and the types of devices they may handle are shown in Figure 2. Also shown are any restrictions that may affect the use of the calls.

Call	Magnetic Tape	Paper Tape	Printer	Card Punch	Card Reader	Linked Disc/Drum	Random Disc/Drum
GET	1,4	1,4			1,4	1,3,4,5	
GETBK	1,4	1,4			1,4	1,3,4,5	
PUT	2,4	2,4	2,4	2,4		2,3,4,5	
PUTSZ	2,4	2,4	2,4	2,4		2,3,4,5	
PUTBK	2,4	2,4	2,4	2,4		2,3,4,5	
COPY	2,4	2,4	2,4	2,4		2,3,4,5	
RELSE	0	0	2	2	1	3,4,5	
FORCE	0						
FSTFM	8,9						
BSTFM	8						
FSREC	0						
BSREC	0						
REWND	8					3,4,5	
WEF	2,7					2,3,4,5	
READ	1,6	1,6			1,6		6
WRITE	2,6	2,6	2,6	2,6			6
RDREC	1,4	1,4			1,4	1,3,4,5	
PRINT	2,4	2,4	2,4	2,4		2,3,4,5	
PUNCH	2,4	2,4	2,4	2,4		2,3,4,5	
WTREC	2,4	2,4	2,4	2,4		2,3,4,5	
EPRNT	2,4	2,4	2,4	2,4		2,3,4,5	

Codes
0 Legal with no restrictions RELSE; FORCE; FSREC; BSREC
1 Must be input file RELSE; READ; RDREC; GET; GETBK
2 Must be output file PUT; PUTBK; PUTSZ; COPY; WRITE; PRINT; EPRINT; PUNCH; WT.
3 Must have variable record type
4 Must have at least one buffer
5 Must have block serial numbers
6 Must not have buffers (MUST HAVE DCW'S) READ; WRITE
7 If FM = 17_8, must not have labels WEF
8 Must not have labels FSTFM; BSTFM; REWND
9 Must be multifile reel FSTFM

Figure 2. GEFRC Call vs. Device Legality

June E. Kim
10-6-64

ERROR PROCEDURES

User-Supplied Routine

GEFRC gives the user the option of providing his own routine for taking corrective action on certain types of errors. The location of this routine is specified by an entry in the file control block.

GEFRC enters the user's error routine by the calling sequence

CALL user-routine (loc-of-fcb, code)

where

user-routine is the location of the error routine as given in subfield 9 in the file control block macro for the file on which the error occurred

loc-of-fcb is the location of the file control block for the file on which the error occurred

code is a number indicating the type of error (in bits 0-17 of the calling sequence word).

If, after performing the desired action within the error routine, the user desires to have GEFRC continue with its normal posterror procedure, then the user routine must not call upon the GEFRC routine that found the error. Since many of the routines use common subroutines the user may be returning to the routine that found the error if he is not careful in his selection. For example, if a call to the GET routine results in an error condition, the user error recovery routine should not call GET if the error routine also makes a normal return to the routine which called it. Such action would cause the program to be aborted.

The error routine is called as a result of a request for the first logical record for which the error applies. For example, an unrecoverable lateral parity error on reading tape causes the error routine to be called when GEFRC receives a request for the first logical record in the incorrect physical record.

Figure 3 shows the possible codes, their meanings, and the GEFRC action if the user routine returns via the RETURN macro-instruction.

When a user's error routine is not supplied, GEFRC will normally abort the program if one of these errors occurs. The only exception to this rule is that the top-of-page status on the printer is handled by GEFRC without informing the user.

Note: The contents of the registers at the time that the user error routine is called are not guaranteed to be the same as at the time GEFRC was called by the user initially. The reason is that GEFRC consists of many levels of subroutines in which the lowest level usually encounters the error. It will call the user error routine directly without proceeding backward through the higher level routines.

Code	Meaning	If user routine returns, GEFRC will:
1	The block serial number recorded in the input file does not equal the block count maintained in the file control block.	Take the value in the input file and place it in the file control block, and then continue processing.
2	A physical record has been read but the device status is not "channel ready."	Proceed as though the error had not occurred.
3	A request to read a physical record was issued; the record has not been read.	Reissue the request to read and continue.
4	A physical record has been written but the device status is not "channel ready."	Proceed as though the error had not occurred.
5	A request to write a physical record was issued; the record has not been written.	Reissue the request to write and continue.
6	Blank tape encountered on read request.	Take EOF exit given in the call to GEFRC (file mark = 00).
7	In a request to backspace a tape the load point was found before the requested number of records had been spaced.	Take EOF exit given in the call to GEFRC (file mark = 00).

Figure 3. Error Codes to User Routine and Meaning of Returns

GEFRC Abort Messages

In the preceding section of this chapter the error conditions were listed as they applied to each calling sequence. If the error condition caused the program to abort, the following message format is printed by SYSOUT:

ABORTED BY GEFRC ROUTINE ----- CODE - FILE CODE --

At the time of the abort, index register 2 contains the location of the file control block of the file which caused the error.

As an additional debugging aid, bits 0-17 of word 15 of the slave program will contain the location of the file control block of one of the files which are currently open. Bits 18-35 of this word will give the count of the currently open files.

The following list of error messages defines the abort reasons noted in the discussion of the calling sequences in the preceding sections of this chapter. (See Figure 4.)

Figure 4 lists the GEFRC abort messages and provides an explanation of codes.

```
ABORTED BY GEFRC ROUTINE ------ CODE - FILE CODE --
                                    L[FCB] FOR ↑ IS IN XR2
BSREC     1   ILLEGAL REQUEST FOR THIS ROUTINE
          2   STATUS NOT TAPE ON LOAD POINT OR OK
BSTFM     1   ILLEGAL REQUEST FOR THIS ROUTINE
          2   REACHED LOAD POINT BUT MORE FILES TO SKIP
CLOSE     1   EOF STATUS ON OUTPUT FILE
          2   UNRECOVERABLE I/O ERROR, NO USER ROUTINE
          3   FILE TO BE CLOSED IS NOT IN CHAIN
          4   ILLEGAL STATUS FOR DISC OR DRUM
FORCE     1   ILLEGAL REQUEST FOR THIS ROUTINE
FSREC     1   FILE IS NOT PRESENT
          2   EOF ON DEVICE FROM PRIOR COMMAND
          3   IMPOSSIBLE RETURN FROM SYSTEM ROUTINE
          4   ILLEGAL FILE DEFINITION IN FCB
          5   UNRECOVERABLE I/O ERROR, NO USER ROUTINE
          6   ILLEGAL REQUEST FOR THIS ROUTINE
          7   I/O STATUS OTHER THAN BLANK TAPE ON READ
FSTFM     1   ILLEGAL REQUEST FOR THIS ROUTINE
GET       1   FILE DESIGNATED AS OUTPUT FILE
          2   ILLEGAL FILE DEFINITION IN FCB
          3   UNRECOVERABLE I/O ERROR, NO USER ROUTINE
          4   BLOCK SERIAL NUMBERS DO NOT AGREE
          5   FIXED OR MIXED REC. SIZE FOR A FILE IS ZERO, OR VAR. REC. SIZE IS ZERO FOR TAPE FILE
GF200     1   EOF STATUS ON OUTPUT FILE
          2   UNRECOVERABLE I/O ERROR, NO USER ROUTINE
GF980     1   TRIED TO CREATE AN ILLEGAL I/O REQUEST
OPEN      1   ILLEGAL DEVICE CODE FOR GEFRC
          2   FILE IS LOCKED
          3   DEVICE IS PRINTER OR PUNCH, NOT OUTPUT FILE
          4   ILLEGAL DISC OR DRUM FORMAT
          5   LINKED FILE BUT NOT VARIABLE RECORD TYPE
          6   ILLEGAL FORMAT FOR SYSOUT FILE
          7   FILE DESIGNATED AS REQUIRED IS NOT PRESENT
          8   TWO FILE DESIGNATORS POINTING TO SAME FILE
PUT       1   EOF STATUS ON OUTPUT FILE
          2   UNRECOVERABLE I/O ERROR, NO USER ROUTINE
          3   ILLEGAL FILE DEFINITION IN FCB
          4   CURRENT LOGICAL RECORD LARGER THAN BUFFER
PUTSZ     1   ILLEGAL REQUEST FOR THIS ROUTINE
          2   NEW SIZE LARGER THAN OLD RECORD SIZE
RDREC     1   BINARY CARD IS NOT A COMDEK CARD
          2   THE COMDEK CARDS ARE OUT OF SEQUENCE
          3   DATA ERROR IN DECOMPRESSING COMDEK CARD
READ      1   ILLEGAL FILE DEFINITION IN FCB
REMOT     1   REMOTE TERMINAL OUTPUT FILE HAS NO BUFFER
REWND     1   ILLEGAL REQUEST FOR THIS ROUTINE
USERR     1   ROUTINE CALLING ERROR WAS USED BY SAME
WAIT      1   UNRECOVERABLE I/O ERROR, NO USER ROUTINE
WEF       1   EOF ON OUTPUT FILE
          2   UNRECOVERABLE I/O ERROR, NO USER ROUTINE
          3   ILLEGAL REQUEST FOR THIS ROUTINE
          4   ILLEGAL STATUS FOR DISC, DRUM OR TAPE
WRITE     1   FILE IS NOT PRESENT
          2   EOF STATUS ON OUTPUT FILE
          3   ILLEGAL FILE DEFINITION IN FCB
```

Figure 4. GEFRC Abort Messages and Explanation of Codes

5. FILE CONTROL BLOCKS

GENERAL

The file control block is the primary communication between the user program and the various GEFRC routines. Its contents serve to define the physical characteristics of the data file and indicate which of the various processing options are to be used for this file. One file control block must be provided for each file used in the program.

FILE CONTROL BLOCK MACRO-INSTRUCTION

The contents of the file control block are supplied by using a special GMAP Macro, as shown in Figure 5. In this example the various subfields of the macro-instruction are shown on separate lines of the form to facilitate explanation. As a general rule, however, they may be presented as multiple comma-separated subfields, using only as many lines as required.

GENERAL ELECTRIC — SYMBOLIC CODING FORMS

PROBLEM FORMAT OF FILE CONTROL BLOCK MACRO-INSTRUCTION

PROGRAMMER | DATE | PAGE | OF

LOCATION (1 2 6)	E/O (7)	OPERATION (8 14)	ADDRESS, MODIFIER (15 16 32)	COMMENTS (32 72)		IDENTIFICATION (73 80)
		FILCB		LOCSYM OF FILE CONTROL BLOCK	1	
		ETC		2-CHARACTER FILE CODE	2	
		ETC		LOCSYM OF FIRST BUFFER	3	
		ETC		LOCSYM OF SECOND BUFFER	4	
		ETC		MAXIMUM BLOCK SIZE	5	
		ETC		RECORD FORM: 0 = VARIABLE, 1 = FIXED, 2 = MIXED	6	
		ETC		RECORD SIZE: NO. IF FIXED, LOCSYM IF MIXED	7	
		ETC		1 IF STANDARD BLOCK SERIAL NOS. NOT USED	8	
		ETC		LOCSYMBOL OF ERROR ROUTINE	9	
		ETC		1 IF NO STANDARD LABELS	10	
		ETC		RECORDING MODE: 0 = BINARY, 1 = BCD, 2 = MIXED, 3=9-track tape		
		ETC		RECORDING DENSITY: 0 = HIGH, 1 = LOW	12	
		ETC		1 IF FILE IS ON MULTIFILE REEL	13	
		ETC		RETENTION PERIOD IN DAYS	14	
		ETC		LOCSYM OF PREHEADER LABEL ROUTINE	15	
		ETC		LOCSYM OF POSTHEADER LABEL ROUTINE	16	
		ETC		LOCSYM OF PRETRAILER LABEL ROUTINE	17	
		ETC		LOCSYM OF POSTTRAILER LABEL ROUTINE	18	
		ETC		FILENAME	19	
		ETC		GEPR OVERRIDE	20	

Figure 5. File Control Block Macro-Instruction

The number of subfields required depends on the nature of the file and varies from a possible minimum of two to a maximum of all 19. The subfields must appear in the order shown below and must be separated by n-1 commas, where the nth subfield is the last one required for definition of the file control block. The operation column of the form must contain FILCB on the first line of the call and ETC on each subsequent line used. The contents of the various subfields must be as follows:

1. Location symbol of the file control block--This symbol is used as the argument for each of the various GEFRC calling sequences in the program. The rules for symbol structure defined for GMAP apply.

2. File code--This is a two-character alphanumeric code unique to this file within the program. This is the same code contained in the GECOS file card and serves to link this file with its allocated physical device. This code must be supplied. If found to be not equal to a code supplied via the file card, the program assumes the file is not present.

3. Location symbol of the first buffer--This field contains the symbolic location of the first word of the first buffer to be used by this file. This must be supplied as a valid GMAP symbol whenever the logical record processing capabilities of GEFRC are to be used. When not supplied or a null field is implied, it is assumed that the program will reference this file using the physical record processing calling sequences exclusively.

4. Location symbol of the second buffer--This field contains the symbolic location of the first word of the second buffer to be used by this file. When supplied as a valid GMAP symbol, it is assumed that the double buffering, logical record processing capabilities of GEFRC are to be used. When not supplied it is assumed that buffering is as indicated by subfield 3, above.

5. Maximum block size--This field contains a decimal value equal to the number of words contained in the largest block of the file. The value must include the word occupied by the block serial number if that option is used (see subfield 8). When the value is not supplied, a GE-625/635 system standard of 320 words is assumed. If buffers have not been defined, the field will be ignored. The maximum value for block size is 4095.

6. Record form indicator--This field contains an indicator of the format of logical records contained within the file. Valid alternatives are:

   0 = Variable-length records
   1 = Fixed-length records
   2 = Mixed-length records

   When this field is not supplied, form 0 is assumed.

7. Record size--The content of this field depends upon the record form indicator specified as subfield 6. If the records are fixed-length (form 1), this field contains a decimal value equal to the number of words contained in each record of the file. If the records are mixed-length (form 2), this field contains a symbolic location of a user-supplied routine which supplies the size of each mixed-length record. This field must be supplied for either of these types of records. When not supplied, it is assumed that the file contains only variable-length records.

8. Block serial numbers--This field contains a 1 if block serial numbers are not included in the file. If the field contains a 0 or is not supplied, it is assumed that the first word of each data block is the standard block serial number word. Block serial numbers must not be used for those files defined by subfield 11 as BCD mode.

9. Error routine exit--If this field is supplied, it contains the symbolic location of the user-supplied error analysis and/or correction procedure. If this field is not supplied, or contains zeros, it is assumed that the standard operational procedures will be executed when an error occurs.

10. Standard labels--This field contains a 1 if standard label processing is not to be performed on this file. If the field contains a 0 or is not supplied, it is assumed that standard labels are used within this magnetic tape file. For files assigned to media other than magnetic tape this field is ignored by GEFRC.

11. Recording mode--This field contains an indicator of the recording mode of the file. For magnetic tape and punched cards the alternatives are:

    0 = Binary mode
    1 = BCD (Hollerith) mode
    2 = Mixed modes (card input files only)
    3 = 9-track tape mode (labels for a file in this mode will be written in binary)

    For perforated tape, they are:

    0 = Normal mode
    1 = Edit mode
    2 = Single-character mode
    3 = Double-character mode

    When this field is not supplied, the binary mode is assumed. When the file is assigned to a medium other than magnetic tape, perforated tape, or punched cards, this field is ignored by GEFRC.

12. Recording density--This field contains an indicator of the density of recording when the file is assigned to magnetic tape. A 1 indicates low density and a 0 or field not supplied indicates high density. Mixed density is not allowed within a file processed by GEFRC.

13. Multifile reel--This field contains a 1 if this file is one of several contained on a single magnetic tape. If the field contains a 0 or is not supplied, it is assumed that only one file is contained within the tape. When the file is assigned to media other than magnetic tape this field is ignored.

14. Retention period--This field contains a decimal value not exceeding 999 to specify the number of days this output magnetic tape file is to be retained. The field is required only for output files which include standard labels. A value of 999 is interpreted to mean permanent retention. If the field is not supplied, a zero value is assumed.

15. Preheader label exit--This field contains the symbolic location of a user-supplied routine used to modify the contents of the header label. The exit will be made on input, after the header label has been read but before it has been checked, on output before the header label has been built. If this field is not supplied, no exit will be made.

16. Postheader label exit--This field contains the symbolic location of a user-supplied routine used to modify the contents of the header label. The exit will be made on input after the standard label check has been completed but while it is still available, on output after the label is built but before it is written. When this field is not supplied, no exit is made.

17. Pretrailer label exit--This field contains the symbolic location of a user-supplied routine used to modify the contents of the trailer label. The exit will be made on input after the label has been read but before it has been checked, on output before it has been built. When this field is not supplied, no exit is made.

18. Posttrailer label exit--This field contains the symbolic location of a user-supplied routine used to modify the contents of the trailer label. The exit will be made on input after the standard label check but while the label is still available, on output after the label has been built but before it has been written. When this field is not supplied, no exit will be made.

19. File name--This field contains a 12-character alphanumeric file name which may be used for label checking or writing. When specified, this field will be used to test the file name contained in the input header label or will be placed into the output label. If the field is not supplied, the label check on file name will be bypassed and the output label file name field will contain blanks. The following examples illustrate the effect of blanks within the file name field (b = blank):

On FILCB macro-instruction card	In file control block
FUNNYbFILEbC	FUNNYbbbbbbbb
[FUNNYbFILEbC]	FUNNYFILECbb
(FUNNYbFILEbC)	FUNNYbFILEbC

20. GEPR override--This field contains an indicator selected by the user to negate a GEPR override option. The user may use this field to exercise special control over the types of exception or error actions in response to a non-ready status on input from a peripheral device. The indicator values that may be entered in this field are 0-7. The indicators control the GEPR override options as follows:

Indicator	Action
0 or blank	No options are set
1	Sets LOC-SYM -3 to -1
2	Sets LOC-SYM -3 to -2
3	Sets LOC-SYM -3 to -3
4	Sets LOC-SYM -3 to -4
5	Sets LOC-SYM -3 to -5
6	Sets LOC-SYM -3 to -6
7	Sets LOC-SYM -3 to -7

(See GE-625/635 Comprehensive Operating Supervisor reference manual for meaning of GEPR override options.)

Note: If the FILCB macro-instruction is continued onto ETC cards, then all except the last must end with a comma. For example, a standard system format file with two buffers might be described by the two cards

```
FILCB    FILEA,AA,
ETC      BUFA1,BUFA2
```

6. WORKING FILE CONTROL BLOCK FORMAT

The expansion of the macro call by the Macro Assembler will result in a variable-size file control block. The file control block is dependent upon the subfields specified within the macro-instruction. As a result, the file control block defining the characteristics of a given file is only as large as necessary for that particular file.

The format of the generated file control block is shown in Figure 6. The right margin indicates the five different sections which may or may not be present for a given file. In the left margin are the word numbers relative to the location symbol of the file control block. Shaded areas are reserved for system use and may be used without any previous warning being given to users.

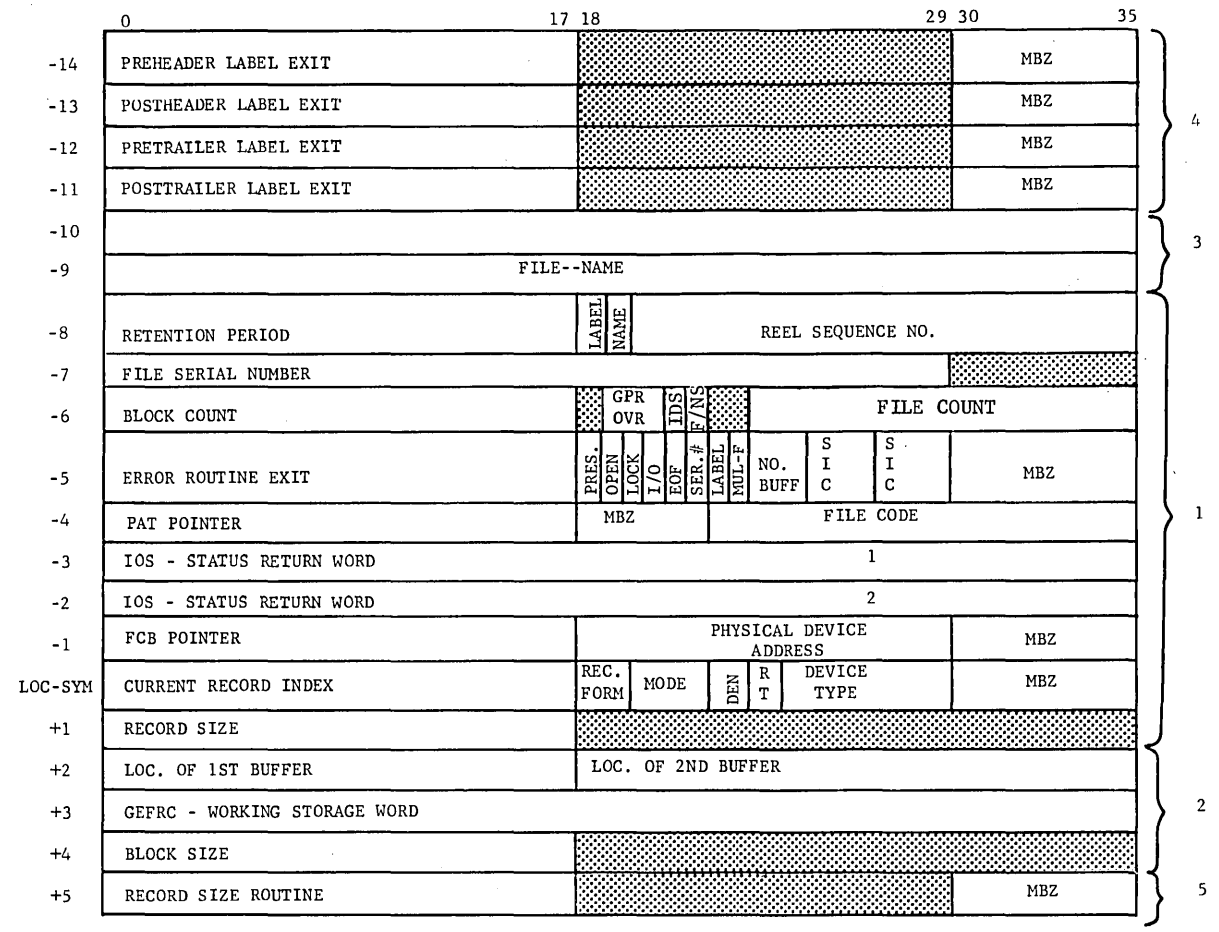


Figure 6. File Control Block Format

The following list describes when the various sections of the expanded file control block must be present:

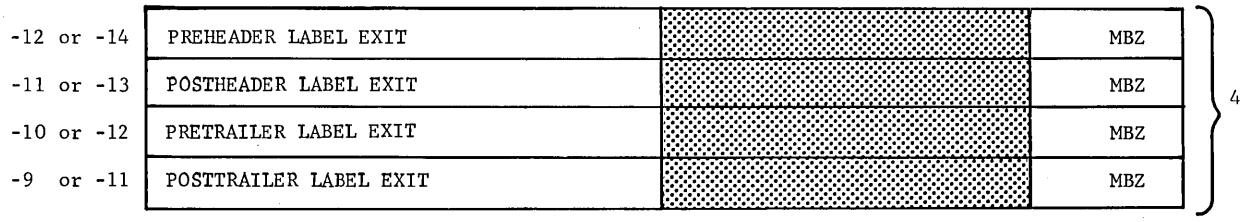
Section No.	Required for following files
1	All files.
2	Files with buffers.
3	Tape files having a file name.
4	Tape files requiring special label processing routine(s) in addition to the standard label routines.
5	Files having mixed-length record type.

The column labeled SOURCE in the lists on the following pages contains a letter code indicating the source of the information in the field being described. The code is defined below:

U -- by user via FILCB macro-instruction
G -- maintained or set by GEFRC
C -- supplied via GECOS
I -- supplied via IOS
V -- varies with usage

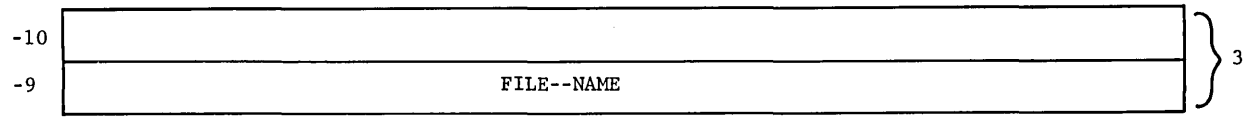
Section 4 of the file control block will be generated in total when any one or all of the special label processing exits have been specified. When present, they occupy words -9 through -12 or words -11 through -14 depending upon the presence of section 3.

The format of these words is as follows:



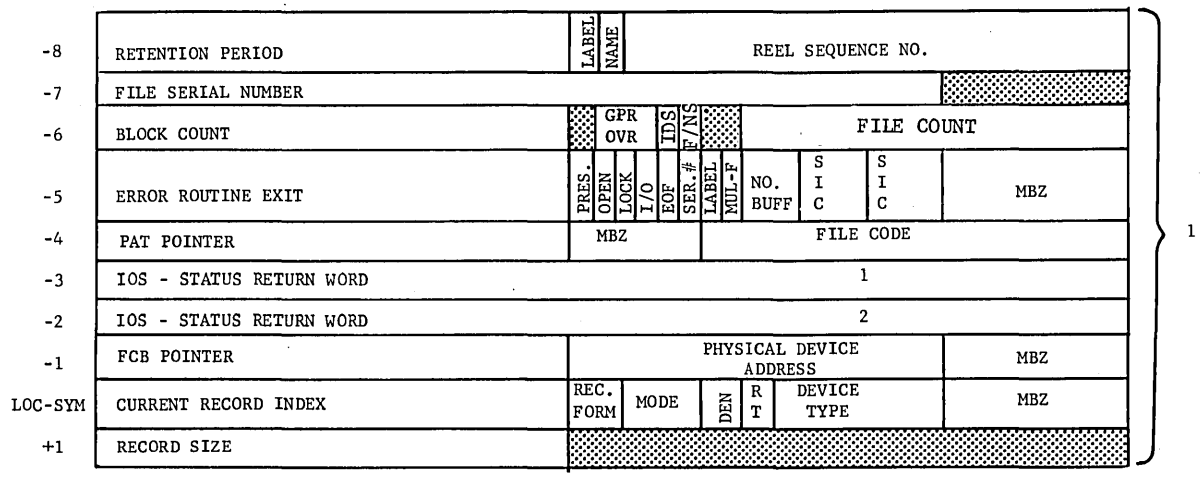
Word Number	Bits	Source	Contents
-12 or -14	0-17	U	Location of the user-supplied routine to be executed as specified by subfield 15 of the file control block macro-instruction; or zero if subfield 15 is not supplied.
	18-29		Zero
	30-35		MBZ
-11 or -13			Same as word -14 but for subfield 16
-10 or -12			Same as word -14 but for subfield 17
-9 or -11			Same as word -14 but for subfield 18

Section 3 of the file control block will be generated when file name is specified as subfield 19 of the file control block macro-instruction. When present, the file name will be contained in the following format:



Word Number	Bits	Source	Contents
-10	0-35	U	The first 6 characters of the 12-character file name
-9	0-35	U	The last 6 characters of the 12-character file name.

Section 1 is the constant portion of the block, which is always present regardless of the file specifications. The details of the fields contained in this section are shown below and on the following pages.



The top two words of section 1 are used for various purposes depending on the physical device allocated to this file. The following shows the configuration of these words when the file contains standard labels and is assigned to magnetic tape.

Word Number	Bits	Source	Contents
-8	0-17	U	Retention period--A 3-character BCD number which is the user-supplied value required for label checking/writing as defined by subfield 14 of the file control block macro-instruction.

Word Number	Bits	Source	Contents
-8 cont.	18	G	Label procedure flag--A switch indicating whether or not any or all of the user label procedure exits are present as section 4 of the generated file control block. 0 Exits not present 1 Exits present
	19	G	File name flag--A switch indicating whether or not the file name has been specified as subfield 19 of the file control block macro instruction. 0 File name not present 1 File name present
	20-35	C	Reel sequence number--The number of the reel within the file. The field is initialized as a binary value during the OPEN routine from data supplied by the file card.
-7	0-29	C	File serial number--A 5-character BCD number which is the reel serial number of the first reel of this labeled magnetic tape file. The field is initialized during the OPEN routine from data supplied by the file card.
	30-35		Zero

When the device assigned to the file is the drum or disc storage unit, these two words will contain a configuration required to accomplish the seek portion of the input/output request. In all other cases these words (-8 and -7) will not be used.

Word Number	Bits	Source	Contents
-8	0-17	G	DCW address, which is the location of word -7.
	18-23		MBZ
	24-35	G	DCW word count of 1.
-7	0-35	U	Will be initialized with the relative seek address obtained from the user in the case of random file processing. When the random file is opened, the first and last link numbers will be found here prior to the first CALL READ or WRITE. For linked file processing this word is used by IOS.
-6	0-17	G	Block count--A binary value equal to the number of physical records read or written within this file on the current reel. On input files it is the number of the current record; on output files it is the number of the last record written.
	19-21	U	GEPR override value on input 0 None 1 Set LOCSYM-3 to -1 2 Set LOCSYM-3 to -2 3 Set LOCSYM-3 to -3

Word Number	Bits	Source	Contents
-6 cont.	22	U	I-D-S common journal indicator: 0 This file not I-D-S journal file 1 This file I-D-S journal file
	23	U	FORTRAN noslew option indicator: 0 Noslew option not present 1 Noslew option present
	24-25		Zero
	26-35	G	File count--A binary value indicating the number of the current file. This value is initialized by the file number specified to the OPEN routine and is adjusted for each standard file mark (17_8) encountered.
-5	0-17	U	The location of the user's error procedure routine as defined by subfield 9 of the file control block macro-instruction, zero when subfield 9 is omitted.
	18	C	File present indicator--The switch indicating whether or not this file has been assigned to a device by GECOS based on the presence of a system card: 0 Not present 1 Present
	19	G	File open indicator--The switch indicating whether this file is open or closed: 0 Closed 1 Open
	20	G	File lock indicator--The switch indicating whether or not the lock option was exercised by the CLOSE calling sequence: 0 Not locked 1 Locked
	21	G	Input/output indicator--The switch indicating whether this file has been opened as input or as output: 0 Input 1 Output
	22	G	End-of-file indicator--The switch indicating whether or not a logical end-of-file condition exists for this file: 0 Not end of file 1 End of file

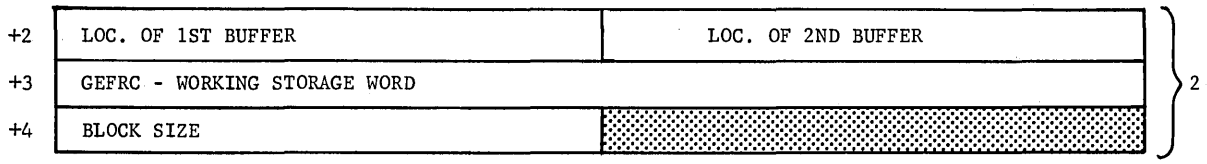
Word Number	Bits	Source	Contents
	23	U	Block serial number indicator--The switch indicating whether or not the data blocks of this file contain the standard block serial numbers: 0 Includes block serial numbers 1 No block serial numbers
-5 cont.	24	U	Standard label indicator--The switch indicating whether or not the file contains the standard labels: 0 Standard labels 1 Not standard labels
	25	U	Multifile tape indicator--The switch indicating whether or not this file is one of several contained on the tape: 0 Not multifile tape 1 Multifile tape
	26-27	U	Numbers of buffers assigned to file. 00 No buffers 01 One buffer 11 Two buffers
	28	G	Special internal control bit. 0 Normal operation 1 For output files, physical device is a unit record device, (printer, card punch) 1 For input files, anticipated physical record is not in memory due to special activity such as record spacing
	29	C	Special internal control bit. 1 if device is magnetic tape which may require dismounting (disposition code C or D)
	30-35		MBZ
-4	0-17	I	Peripheral assignment table pointer--Initially zero and subsequently set by IOS, this provides the actual physical device address of the device assigned to this file by GECOS.
	18-23		MBZ
	24-35	U	File code--The user-supplied code as defined by subfield 2 of the file control block macro-instruction.

Words -3 and -2 are the status words returned by IOS. The fields within these two words of general interest to the GEFRC user are:

Word Number	Bits	Source	Contents
-3	0	I	Termination indicator
	2-5	I	Major status
	6-11	I	Substatus
	30-35	I	Record count residue
-2	0-17	I	Address +1 of last data word transmitted.
	24-35	I	Word count residue.
-1	0-17	G	File control block pointer--The location of another file control block in this program. This provides a link in the chain connecting all files currently open.
	18-29	C	Physical device address.
	30-35		MBZ
LOCSYM (0)	0-17	G	Current record index--The location of the first data word of the last referenced logical record in the buffer area.
	18-19	U	Record form: 00 Variable-length records 01 Fixed-length records 10 Mixed-length records
	20-21	U	Recording mode: 00 Binary mode 01 BCD mode 10 Mixed (card files only) 11 9-track tape mode (labels in binary) Perforated tape: 00 Normal 01 Edit 10 Single character 11 Double character
	22	U	Recording density: 0 High density 1 Low density

Word Number	Bits	Source	Contents
0	23	C	Special internal control bit: 1 if output file assigned to remote terminal device
		G	1 if input file causing COMDK output
	24	C	If disc or drum: 1 = random 0 = linked If tape: 1 = 9-track 0 = 7-track
	25	C	1 if file assigned to SYSOUT
	26-29	C	Code for type of device (octal): 2 Magnetic tape 4 Disc 6 Drum 10 Card reader 12 Printer 14 Perforated tape 16 Card punch
	30-35		MBZ
+1	0-17	V	Record size--Contains a binary value equal to the number of data words contained in the last referenced logical record of the file. For files without buffers, this field contains the location of the last CALL READ or CALL WRITE request.
	18-35		Zero For files without buffers, this field contains the location of the last control words of the last CALL READ or CALL WRITE request.

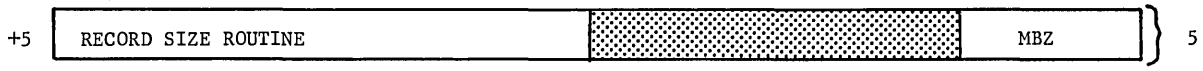
Section 2 of the generated file control block will be present when the number of buffers (word -5, bits 26-27) is nonzero. When present, its contents are as follows:



Word Number	Bits	Source	Contents
+2	0-17	U	Location of first buffer assigned to this file.
	18-35	U	Location of second buffer assigned to this file, or location of first buffer if only one assigned. Note that for two buffers this word has the form A,B and for one buffer it has the form A,A.

Word Number	Bits	Source	Contents
+3	0-35	G	Reserved for internal GEFRC control: for input files, it contains the location plus one of the last data word in the current buffer (bits 0-17). For output files, it contains the number of words still available for assignment in the current buffer (bits 0-17).
+4	0-17	U	Block size--A binary value equal to the maximum number of words in the largest physical block contained in this file. In other words, the size of the buffer(s) less one.
+4	18-35		Zero

Section 5 of the generated file control block will be present when the record form indicator (subfield 6) indicates mixed-length records (form 2). When present, its format is as follows:



Word Number	Bits	Source	Contents
+5	0-17	U	Location of user-supplied routine required when the input record form is specified as mixed.
	18-29		Zero
	30-35		MBZ

7. FILE DESIGNATOR WORDS

The OPEN and CLOSE routines obtain their data from file designator words supplied by the user. These words are used for both opening and closing files, and they indicate the options to be exercised. The format for the file designator word is as follows:

bits 0-17 Location of the file control block.

18 Input/output indicator:

- 0 Input file
- 1 Output file

19 Rewind-on-OPEN option:

- 0 Either do not rewind the file or this option is not appropriate to the device
- 1 Rewind the file when opened

20-21 Rewind-on-CLOSE option:

- 0 This option is not appropriate to this device
- 1 Lock the file when closed. Release via MME GERELS
- 2 Do not rewind and do not lock the file when closed
- 3 Rewind but do not lock the file when closed

22 Prime input buffer option:

- 0 Normal operation
- 1 This is a buffered input file and the buffer is not to be filled when opened

23 Put-size on output option:

- 0 Normal operation
- 1 This is a buffered output file for which a CALL PUTSZ command with record size equal zero is to be issued when closed

24 Optional/required status on OPEN:

- 0 Normal operation
- 1 Abort if file is not present

25-35 File positioning value for OPEN:

0 Honor the rewind option as given in bit 19 but do not otherwise position the device

n This file is the nth file on a multifile tape reel. Rewind the tape and space forward over n-1 files

Note that "rewind" for a linked file on drum/DSU means that the file is to be positioned at the beginning of the assigned area for that file.

The file designator words may be specified in assembly language with the VFD operation of the form:

VFD 18/fcb, 1/io, 1/open, 2/close, 1/prime, 1/size, 1/req, 11/file

fcb = Symbolic location of the file control block
io = Input/output indicator value
open = Rewind-on-OPEN option value
close = Rewind-on-CLOSE option value
prime = Prime input buffer option value
size = Put-size on output option value
req = Required file option
file = File number

8. BUFFERS

Whenever the logical record blocking and/or deblocking facilities of GEFRC are to be used, one or two buffers must be defined and reserved. These buffers may be reserved in any portion of allocated memory and each must be one word larger than the largest physical record it is to hold. As an example, suppose we wish to describe a file whose code is AA, which is to be double buffered with physical records up to 200 words long. The file control block macro-instruction might be:

```
FILCB   FILA,AA,BUFA1,BUFA2,200
```

The program should also contain the following commands to reserve the buffer space:

```
BUFA1   BSS     201
BUFA2   BSS     201
```

The placement of these commands in relation to each other is unimportant.

The additional word required for each buffer is used by GEFRC as a buffer control word. This word, which is the first word of the buffer, is used to control the data flow to or from the buffer during data transmission. It is also used during the processing of the records within the buffer. For input files, it contains a pointer (bits 0-17) to the word which follows the last data word of the current logical record. For output files, it contains (bits 0-17) the number of words of the buffer which have been used.

When only one buffer is provided for a file, GEFRC relinquishes control to some other activity while transmission to or from that buffer is taking place. With double buffering, one buffer is used for processing and the other for transmission. GEFRC then relinquishes control only when it wishes to exchange the roles of the buffers and finds that the transmission was not complete when processing terminated on the other buffer. Thus the choice between these two forms of buffering usually depends upon the memory which is available, together with a consideration of the activity to be expected from the file.

9. SYSTEM INPUT AND OUTPUT FILES

SYSTEM INPUT FILES (VIA GEIN)

Input files created by GEIN (namely, I* and any other file designated by a $ DATA control card) are recorded in standard system format. Therefore, the file control blocks to process these files must specify:

1 - block serial numbers present
2 - variable record type
3 - block size = 320 words

In these cases, the end-of-file is automatically appended to the created file by GECOS Input Media Conversion (GEIN).

SYSTEM OUTPUT FILES (VIA SYSOUT)

The file designated by the file code P* is automatically assigned to the "device" SYSOUT unless the file code P* appears on a $ file control card assigning it to some specific device. Also, any other file may be assigned to SYSOUT at execution time by a $ SYSOUT control card. All files assigned to SYSOUT must have the following requirements.

1 - block serial numbers present
2 - variable record type
3 - block size $\leq$ 320 words

Every physical record of a SYSOUT file will contain an additional two-word logical record to identify the activity for which the record is being written. Therefore, the block size must be large enough to accommodate this record. For example, if a SYSOUT file is to contain output records consisting of 20 data words, then the block size must be at least 24 (20 data words + 2 words for identification + 1 control word for the data record + 1 block serial number word). The identification record is inserted by GEFRC automatically.

The logical records must contain media codes and report codes in their control word. These codes are inserted by CALL's to PRINT, PUNCH and WTREC.

The number of different report codes must not exceed eight and the codes must be values in the range $0 \leq code \leq 55_{10}$.

The total number of logical records in the complete set of files assigned to SYSOUT for the activity must not exceed the number specified in the $ LIMITS control card.

10. ALTER AND COMPRESSED DECK FORMATS

ALTER PROCEDURE

An Alter file is created by the input phase of GECOS based on the occurrence of the $ UPDATE control card. The presence of this file is made known to the RDREC routine of the I/O Editor when a bit in the "switch word" has been set "ON" by GECOS. If such a file has not been created, the Alter routine will make no attempt to open the file and the RDREC routine will subsequently ignore the Alter file for that activity.

When present, the Alter file will consist of control cards and source or symbolic language cards to be merged with the cards of the primary input file. The format of these control cards is as follows:

1	8	16
$	ALTER	n,m

where n and m are Alter numbers that define whether the following cards are to be added to or replace cards in the primary input file. The Alter numbers used on the control card are taken from the previous listing of the program deck now being processed and must not be larger than the last Alter number of that deck. These numbers are simply consecutive card numbers starting with 00001 and increasing by 1 for each source input card.

When it is desired to insert cards into a deck, the m subfield is not used. In this case, the cards following this $ ALTER card up to, but not including, the next $ ALTER card, will be inserted just prior to the card corresponding to Alter number n. To delete and/or replace one or more cards from a deck the m subfield is given as shown above. When n and m are equal, card n will be deleted. When m identifies a card following n, all cards n through m will be deleted. In addition, any cards following this $ ALTER card up to, but not including, the next $ ALTER card will be inserted in place of the deleted cards. The end of an Alter file is designated by the normal end-of-file convention appropriate to the medium containing the file.

Note: The Alter file must be prepared in ascending Alter number order. That is, the $ ALTER card

```
$   ALTER 100,102
```

must precede the $ ALTER card

```
$   ALTER 105,107
```

COMDEK FORMAT

The COMDEK format is produced by compressing any Hollerith-coded card image by removing sequences of 3 or more blanks and packing the information in standard column binary form.

A = # of characters in field (up to next string of 2 blanks)
B = # of characters in string

To accomplish the compression, the Hollerith card is considered as being made up of a series of fields and strings. A "field" is defined as a segment of the card containing no sequences of more than two consecutive blanks except at the beginning. A "string" is that portion of a field obtained by deleting any leading blanks.

Each field specification starts with the octal value of A($0 \leq A \leq 67_8$) followed by the octal value of B($0 \leq B < 67_8$) followed by the B characters constituting the string. (A=the number of characters in the field; B=the number of characters in the string.)

The size of A and B is limited, as indicated above, in order to reserve a set of codes to serve as flags when found in a position in which a count had been expected. If a given length exceeds the maximum length, it is segmented into separate fields. For example, given 70 (decimal) consecutive nonblank characters, it is necessary to treat this as two fields with:

Field 1 A = 67, B = 67 (octal values)
Field 2 A = 17, B = 17 (octal values)

The field specifications (A,B,string) are packed sequentially on a binary card in the format indicated below. A field specification may be started on a COMDEK card (X) and may be completed on the following card (X+1); but in doing so, the field is split into two parts. The first two characters of card X+1 will be the A and B characters for the second part.

The following codes for A are used to designate specific conditions. The B character is not present in such cases.

A = 0	End of a compressed card. Continue decoding on the next card
$A = 77_8$	End of encoded string for a given Hollerith card image
$A = 76_8$	End of the compressed deck segment
$A = 70_8 - 75_8$	Available for extension

The COMDEK card layout consists of:

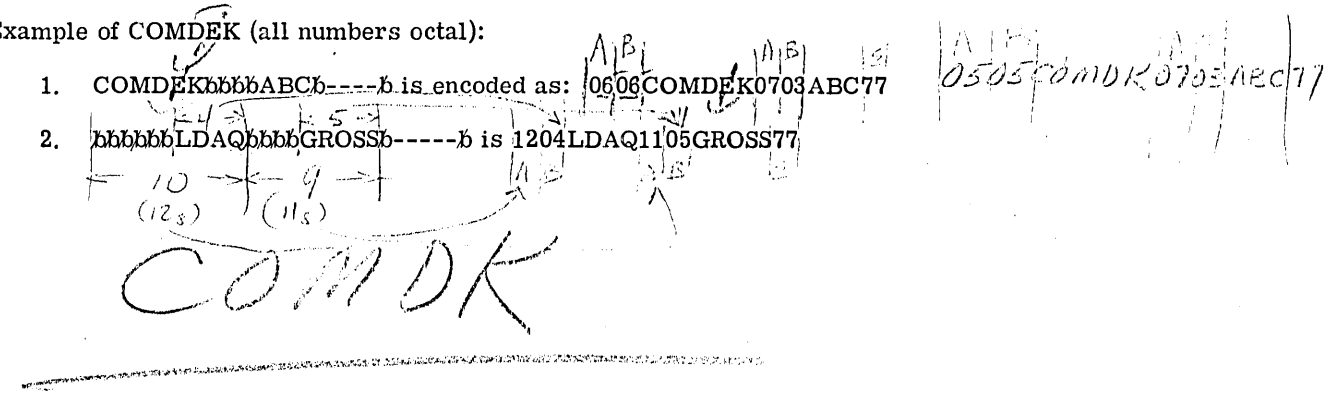
Word 1:	0-2	Column binary card type 5
	3-8	Zeros
	9-11	101 (7-9 punches)
	12-35	Binary sequence number
Word 2:		Checksum of word 1 and words 3 - 24
Words 3-24:		Compressed card image
Words 25-27:		Hollerith-coded label or zeros

The binary sequence number is maintained when a COMDEK output is produced and is checked when the deck is used as input. When a sequence error is found in an input COMDEK file the activity will be terminated.

The label portion of the COMDEK format will be supplied by the Input/Output Editor when the appropriate initialization is accomplished.

Example of COMDEK (all numbers octal):

1. COMDEKbbbbbABCb----b is encoded as: 0606COMDEK0703ABC77
2. bbbbbbLDAQbbbbGROSSb-----b is 1204LDAQ1105GROSS77



11. SYSTEM FILE CONTROL CARDS

The system file control cards are supplied by the user at execution time for each file that is to be processed by GEFRC. In general, the file control cards are used by GECOS to assign physical devices to the various files used by the program. In addition, the file cards are used in certain cases to provide object-time label-checking parameters to GEFRC.

When a file control card is not present for a file referred to by the program, GEFRC assumes that the file is not present and treats all input/output requests as if the file was at an end-of-file condition.

A description of the various file control card formats as they apply to GEFRC follows:

MAGNETIC TAPE FILE CONTROL CARD

The file control card used for files on 7-track magnetic tape is supplied as follows. The file control card used for files on 9-track magnetic tape is identical except that columns 8 through 12 contain TAPE9. (See also GE-625/635 Comprehensive Operating Supervisor, CPB-1195.)

	1	8	16
Symbolic Example	$	TAPE	File Code, Logical Unit Designator, Secondary Logic Unit Designator, File Serial No., Reel Sequence No., File Name
Actual Example	$	TAPE	AB,A1S,,24,0001,BIGFILE

Where:

1. File code is a 2-character alphanumeric code assigned by the programmer to uniquely identify a given file within his program. This is the same file code which is given as subfield 2 in the file control block macro-instruction.

2. Logical unit designator is a three- or four-character designation of the initial channel and unit assigned to the file. The first character is the symbolic channel designator. This designator can be any symbolic letter. No attempt is made to separate letter channel requests unless the letter is "X." Thus, all "A" requests must be satisfied on a single channel. If the channel designator is "X," however, the device request will be allocated cyclically, one device to a channel until the request is satisfied.

   The second portion of the Logical Unit Designator is numeric (one or two digits $\leq$ 62). The numerals have no specific meaning except that, for any one channel, the quantity of different numerals is equal to the number of files requested on that channel.

The last character of the channel unit designator is used to indicate the disposition of the unit following execution of the current activity. This is referred to as the disposition code and as such is used to:

a. Assign a physical device containing a file created in a prior activity to the proper input file for the current activity.

b. Ensure that a magnetic tape file to be removed from the computer is rewound with standby so that it may not be written on and inadvertently reallocated.

c. Allow a magnetic tape which has been written on to be temporarily dismounted, its write permit ring pulled, remounted, and read.

d. Inform the system when a file is to be released for use by other activities/jobs.

The specific disposition codes are as follows:

Code	Meaning	One or more of the following actions occur:
S	Save	1. No specific termination action. 2. Unit is assigned to files with the same channel unit designator in subsequent activities of this job.
C	Continue	1. Tape is rewound using rewind and standby. 2. Operator is told to remove the write permit ring and remount the tape. 3. Unit is assigned to files with the same channel unit designator in subsequent activities of this job.
D (or Code Omitted)	Dismount	1. Tape is rewound using rewind and standby. 2. Operator is issued dismounting instructions. 3. Tape is made available for assignment as a utility tape.
R	Release	1. Tape is rewound using rewind instructions. 2. Dismounting instructions are not to be issued. 3. Tape is made available for assignment as utility tape.

3. Secondary Logical Unit Designator is a nonblank combination of characters used to denote a second tape channel unit which is assigned to a job. This enables alternate use of the two, thus achieving more efficient utilization of the units. When a multifile reel is not used, this variable is omitted. However the comma denoting end-of-field must be included if other fields follow.

4. File Serial Number is the tape serial number of the first reel of the file. This value will normally be used as the primary input file label checking parameter when using GEFRC. When this value cannot be specified in advance for a labeled input file, the user may set this field to 99999, in which case the label checking procedure will ignore this field. This field is not required and may be omitted for output files and unlabeled input files. Mounting instructions will be issued to the operator if this field is present and not 99999.

5. Reel sequence number is the number of the reel within a given file at which processing is to begin. This value will be used as a label checking parameter for all labeled input files processed by GEFRC. This field is not required and may be omitted for output files and unlabeled input files. However, the comma denoting end-of-field must be included if other fields follow.

6. File Name is a 12-character or less literal name given to the file for external identification and is used to issue mounting instructions to the operator.

FILE CONTROL CARD for N UTILITY MAGNETIC TAPES

The file control card used for assigning N utility magnetic tapes is as follows:

	1	8	16
Symbolic Example	$	NTAPE	File code, Channel Designator Number of Tapes
Actual Example	$	NTAPE	U1,A,3

The NTAPE control card requests a number of utility tapes to be assigned to the tape channel specified. File codes are assigned by GECOS, beginning with the file code specified and increasing by one (binary addition) for each tape assigned.

File Code--The first tape assigned will be given this file code; each subsequent tape will be assigned a file code equal to one (1) plus the file code of the previous tape.

Channel Designator--This entry specifies the symbolic tape channel on which the tapes are to be allocated.

Number of Tapes--This entry specifies the number of utility tapes to be allocated.

FILE CONTROL CARD FOR DRUM/DISC

The file control card used for files assigned to the drum or disc storage unit is as follows:

	1	8	16
Symbolic Example	$	DISC	File Code, Logical Unit Designator, Access
Actual Example	$	DISC	C1,A1,4R

Where:

1. File Code and Logical Unit Designator are as described for $ TAPE control cards.

2. Access is a coded designator indicating whether the file is to be processed randomly or sequentially (linked) as a pseudo tape.

When a file is stored on the drum or disc as a random file, all references to elements of the file by the user program must be made relative to the beginning of the area allocated to the file. When a WRITE operation is performed, a test is made to insure that a specified relative address is within the allocated bounds of the file.

When a file is stored on the drum or disc as a linked file, appropriate chaining is provided from one block of the file to the next.

For a random file, this designator must contain an "R" preceded by the number of 3840 word blocks that the user wishes to have assigned to this file. For a linked file, this designator must contain an L preceded by the number of 3840 word blocks required.

Disposition codes for nonmagnetic tape files are as follows:

Code	Meaning	Action Taken at Activity Termination
S	Save	This device is assigned to files with the same channel unit designator in subsequent activities of this job.
R	Release	Make available for use by another activity/job.
If a D (Dismount) or a C (Continue) is inadvertently included, they will be interpreted as an R (Release) since these codes do not apply to disc or drum files but only to tape files.		

OTHER FILE CONTROL CARDS

The file cards used for the other types of assignable peripherals are

1	8	16
$	PPTP	File code, Logical Unit Designator
$	PPTR	File code, Logical Unit Designator
$	PRINT	File code, Logical Unit Designator
$	PUNCH	File code, Logical Unit Designator

1	8	16
$	READ	File code, Logical Unit Designator
$	TYPE	File code, Logical Unit Designator

where the assignable types are indicated as:

PPTP	Perforated tape punch
PPTR	Perforated tape reader
PRINT	High-speed printer
PUNCH	Low-speed punch
READ	High-speed card reader
TYPE	On-line console typewriter

Note: For a more complete discussion of the file control cards, consult the manual entitled GE-625/635 Comprehensive Operating Supervisor, CPB-1195.

12. LABEL PROCESSING AND UNIT SWITCHING

GEFRC includes a complete facility for processing standard labels and for performing associated unit switching at the end of magnetic tape reels. The procedures included are specifically designed for the standard case and as such will not perform label functions on nonstandard labels. Exits from GEFRC to user-supplied routines allow the programmer to supplement these procedures but will not provide for the omission or substitution of any label or format. If an alternate standard is desired, the label processing routines of GEFRC will have to be replaced within the GEFRC system. Standard label procedures apply only to magnetic tape files.

FORMAT OF LABELED FILES

The format of a labeled file is as follows:

1. Beginning of medium indicator (load point reflective foil)
2. BTL header label
3. End-of-file mark (octal 17 character)
4. Data blocks
5. End-of-file mark (octal 17 character)
6. EOR or EOF trailer label
7. End-of-file mark (octal 17 character)

If the file is multireel, item 6 is an end-of-reel (EOR) label for every reel but the last one, on which it is an end-of-file (EOF) label. If a given unit contains more than one file, items 2-7 are repeated for each file. GEFRC assumes that all elements of a file are recorded in a single density and mode.

STANDARD LABEL FORMATS

All standard magnetic tape files must contain the following 14-word labels:

1. Header Label

Word Number	Format	Description
1-2	GEbb600bBTLb	Label identifier
3	xxxxxx	Installation identification
4	bxxxxx	Tape reel serial number
5	bxxxxx	File serial number
6	bbxxxx	Reel sequence number
7	byyddd	Creation date yy = Year ddd = Day of year (001-365)
8	bbbxxx	Retention days
9-10	xxxxxx	File name
11-14	(arbitrary)	Not used. Available to user program

Installation identification is constant information for a given user installation. This field is ignored by label checking routines.

Tape reel serial number is the serial number of the physical tape reel. This number is also recorded externally on the reel itself.

File serial number is the "tape reel serial number" of the first reel of the file.

Reel sequence number is the number of the reel within a given file. For example, the first reel of a file is reel 0001, the second is 0002, etc.

2. Blank Reel Label

Word Number	Format	Description
1-4	Identical to that described under "Header Label"	
5-10	000000	All zero
11-14	(arbitrary)	Not used

3. Trailer Label

Word Number	Format	Description
1	bEORbb or bEOFbb	End-of-reel label or End-of-file label
2	xxxxxx	Block count
3-14	(arbitrary)	Not used

Block count is the number of data blocks or physical records contained on this reel of this file. In the case of a multifile reel the block count is the number of data blocks for the current file only. This number is recorded as a 36-bit binary number or a 6-character BCD number, depending upon the recording mode of the file.

Figure 7 shows information carried on standard tape header labels and describes where GEFRC gets the information.

FIELDS	FIXED	VARIES PER REEL	VARIES PER USAGE	WHERE FOUND
BTL	X			
INSTALLATION ID	X			
REEL SERIAL #		X		IN LABEL
FILE SERIAL #			X	FCB VIA FILE CARD
REEL SEQ. #			X	FCB VIA FILE CARD
CREATION DATE			X	CURRENT/LABEL
RETENTION DAYS			X	FCB VIA PROGRAM
FILE NAME			X	FCB VIA PROGRAM

Figure 7. Standard Tape Header Label Fields

LABEL CHECKING

Input Header Label

When a file is opened, and at the beginning of each new reel (if it is a multireel file) the label is checked to ensure that the correct reel is being processed. The following conditions must be met for an input label to be valid:

1. The file serial number of the file control block must agree with the file serial number in the label. This test may be bypassed by supplying a file serial number of 99999 in the corresponding file control card. Bypassing of this test will be desired when a file has been created in one activity and used as input to the next activity. In this case the file serial number is not known at the time the job is submitted.

2. The reel sequence number of the file control block must agree with that contained in the label. When reel switching occurs, the reel sequence number contained in the corresponding file control block is incremented by GEFRC.

3. The file name in the file control block must agree with the file name contained in the label. If no file name is contained in the file control block, this test is bypassed.

If any of the above checks fail, a message to the operator will be typed. This message will include the SNUMB, unit code, and all necessary information from the label and the file control block. The operator may then accept the reel as valid or mount the correct reel to be processed.

In the event that a check does fail and the operator accepts the reel as valid, any remaining tests of the label contents are bypassed. For details of bypassing labels refer to the GE-625/635 Operator's Reference Manual, CPB-1045.

Figure 8 shows the fields described under Input Header Label checking, and illustrates the information GEFRC uses for the label checks. Also described are the output label checks for the same fields, which are discussed on the following pages.

FIELDS	INPUT	OUTPUT
REEL SERIAL #		GET FROM OLD LABEL OR OPERATOR
FILE SERIAL #	FCB = 99999, OK; FCB = LABEL, OK	IF IN FCB, USE IT; IF NOT, USE REEL SER. #
REEL SEQUENCE #	LABEL = FCB, OK	USE FCB VALUE
FILE NAME	MISSING IN FCB, OK; LABEL = FCB, OK	IF IN FCB, USE IT; IF NOT, BLANKS

Figure 8. Input/Output Label Checks

Input Trailer Label

Each labeled file is assumed to be a multireel file. The occurrence of an EOR label causes tape switching to the next reel. The occurrence of an EOF label, or any other record where an EOR label is expected, will cause an exit to the EOF address of the read routine. The block count which appears in each trailer label created by GEFRC will be checked and a sequence error exit will be taken if the count does not agree with the block count maintained in the file control block.

Output Header Label

Every reel upon which a labeled output file is to be written must have a label on that reel. This may be a blank reel label or a header label on which the retention period had expired. (Retention period = date created + retention days.) If these conditions are not met, a message is typed to the operator indicating the invalid condition. The operator may accept the reel as valid by entering the correct reel serial number or remove the reel and mount a new labeled reel.

Output Trailer Label

When the end-of-tape foil is encountered during the course of writing on an output reel, an EOF mark, an EOR label, and another EOF mark will be written on the tape. The reel will then be rewound and unit switching will take place. The output label written will include the block count from the file control block which indicates the number of data blocks recorded on the reel. When the new reel is provided, GEFRC will continue processing.

USER-SUPPLIED LABEL ROUTINES

GEFRC provides four points within the label-processing routines at which an exit may be made to the user-supplied routines. These four points are:

1. Before beginning label--On input, after the label has been read but before it has been checked; on output, before the label has been built.

2. After beginning label--On input, after the standard label check has been completed but while it is still available; on output, after the label is built but before it is written.

3. Before ending label--On input, after the label has been read but before it has been checked; on output, before it has been built.

4. After ending label--On input, after the standard label check but while the label is still available; on output, after the label has been built, but before it has been written.

GEFRC will exit to the user's routine at each of these points for which a symbolic location has been supplied in the file control block. This exit is performed by the macro-instruction.

CALL user-routine (fcb, buffer)

Where:

user-routine is the symbolic location of the user-supplied label routine obtained from the file control block

fcb is the symbolic location of the file control block

buffer is the symbolic location of the 14-word tape buffer contained within the GEFRC label routine.

The user routine then performs the desired processing of the label, using buffer to locate the label. The user routine must return to GEFRC via the RETURN macro-instruction.

The user routine must not issue a logical record request for the file involved in the label procedure. To do so could result in a nonstandard file and will cause an error procedure to be entered when control is returned to GEFRC.

UNLABELED FILE PROCEDURES

While the processing of labeled files is completely automatic, GEFRC is also capable of handling unlabeled files. The difference lies in the EOF procedures.

When an EOF tape mark is encountered on an unlabeled input file, the return from the read is always to the EOF location specified. It is the user's responsibility to determine the true end of his data file. A subsequent request to read will indicate that unit switching is required.

When an EOF tape mark is encountered on an unlabeled multifile reel, a subsequent read request is interpreted as a request for the first data block following the tape mark. It is the user's responsibility to determine the true end of his data. If an attempt is made to read beyond the EOF mark of the last file of the reel, an erroneous tape condition may result.

For unlabeled output files, the detection of the end-of-tape foil will cause an EOF mark to be written and unit switching to occur. In the case of multifile reels, however, only the first reel will be acceptable as input to processing with use of GEFRC.

APPENDIX A. PROGRAMMING EXAMPLES

This appendix contains information to aid the programmer in the use of GEFRC.

PROGRAM SIZE

One of the more important problems to a programmer is estimating the size of the program. Figure 9 shows all the calling sequences a programmer may use and also gives the size of each GEFRC subroutine that a particular call may include.

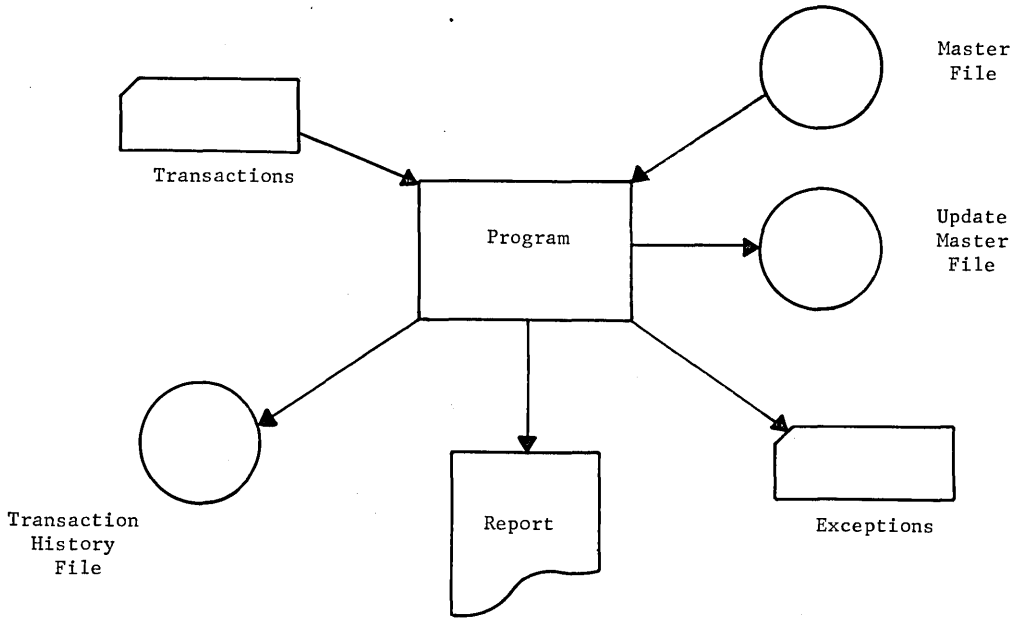
USER'S CALLS																						SUBROUTINES	
WTREC	PUNCH	EPRINT	PRINT	RDREC	IOEDIT	WAIT	WRITE	READ	WEF	REWND	BSREC	FSREC	BSTFM	FSTFM	FORCE	SETOUT	SETIN	CLOSE	OPEN	RELSE	COPY	PUTBK	PUTSZ	PUT	GETBK	GET		NUMBER OF WORDS
X	X	X	X	X		X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	BASIC (OPEN,CLOSE,etc.=998 plus label check, unit switch =330	1328
				X			X	X				X													X	X	.GR250	36
				X																					X	X	GET	218
X	X	X	X	X																	X	X		X			PUT	150
																							X				PUTSZ	62
											X	X			X					X							RELSE	26
																	X										SETIN	22
																X											SETOUT	56
															X												FORCE	56
														X													FSTFM	60
													X														BSTFM	80
												X															FSREC	150
											X																BSREC	126
										X																	REWND	44
									X																		WEF	114
								X																			READ	56
							X																				WRITE	54
						X																					WAIT	50
	X	X	X	X	X																						IOEDIT	60
	X			X																							.GILLB	208
	X	X	X	X																							.GBCD,.GBNRY	44
				X																							RDREC,.GCMDK,.GALTR	1344
		X	X																								PRINT,.GIOPG	228
		X																									EPRINT	160
	X																										PUNCH	72
X	X	X	X	X																							WTREC	34

Figure 9. GEFRC Calling Sequences and Subroutine Sizes

FILE CONTROL BLOCK MACRO USAGE

The usage of file control block macro-instructions can best be shown by giving some input or output file parameters, then coding a file control block macro as an example to describe each file. Therefore on the following pages an example of a file update problem is presented with the input and output file parameters listed. Using these parameters, some examples of file control block macro coding are shown. The detail coding of the problem is not presented as it would vary according to the programmers interpretation.

A top level flowchart of the program is shown below.



General parameters:

1. The input transactions are Hollerith cards to be read from the GECOS data file I*.

2. The transaction history file is a printed report to be written on SYSOUT.

3. The master file is in the binary mode, with record size fixed at 60 words.

4. The report shows the complete master file with the transactions that were matched, and it is to be printed in a subsequent bulk media conversion activity.

5. The output cards would be any transactions that did not match the master file. These cards would be punched as duplicates of the transactions and used for checking back the transactions that did not match the master file. They would be written on SYSOUT.

Master file parameters:

Fixed-length records
Record size--60 words
Block size--300 words
Two buffers
Standard labels
Multireel file
High density
Binary mode
File name is INV.MASTER
Location of input file control block--MASTF
Location of output file control block--UPMAST
GEFRC CALLS for this file: OPEN, CLOSE, GET, COPY

The file control block macro-instructions used to define both the input and output master files are described below. Also described are the VFD pseudo-operations that define the file designator words which provide data for opening and closing of the files and the BSS pseudo-operation that allocates the buffer areas.

PROBLEM

PROGRAMMER DATE

	LOCATION	E/O	OPERATION	ADDRESS, MODIFIER	COMMENTS
	1 2 6	7	8 14	15 16	32
I	MFDES		VFD	18/MASTF,1/0,1/1,2/1,14/0	
Ø	MFDES		VFD	18/UPMAST,1/1,1/1,2/1,14/0	
			FILCB	MASTF,MF,INBUF1,INBUF2,300,1,60,1,,,Ø,,,,,,,,INV.MASTER	
			FILCB	UPMAST,UP,OUTPT1,OUTPT2,300,1,60,1,,,Ø,,,7,,,,,INV.MASTER	
I	NBUF1		BSS	301	
I	NBUF2		BSS	301	
	OUTPUT1		BSS	301	
	OUTPUT2		BSS	301	

Transaction file parameters (card input):

Location of file control block--TRANS

File code--I*
Block size--320 words
One buffer
Variable record length with block serial numbers
GEFRC CALLS for this file: OPEN, CLOSE, GET

The file designator word and the file control block macro-instruction that define the input transaction file are described on the following page along with the buffer area allocation.

PROBLEM

PROGRAMMER DATE PAGE OF

LOCATION	E/O	OPERATION	ADDRESS, MODIFIER	COMMENTS	IDENTIFICATION
1 2 6	7	8 14	15 16 32	72	73 80
CDDES		VFD	18/TRANS,1/0,1/0,2/0,14/0		
		FILCB	TRANS,I*,CDBUF		
CDBUF		BSS	321		

Punched error card parameters and transaction history file parameters:

Variable-length records
Block size--320 words
One buffer
Location of the file control block--PUNPR
File code is P*
GEFRC CALLs for these files: OPEN, CLOSE, PRINT, PUNCH

The file designator word and the file control block macro-instruction that define the transaction history file are described below along with the buffer area allocation.

PROBLEM

PROGRAMMER DATE PAGE OF

LOCATION	E/O	OPERATION	ADDRESS, MODIFIER	COMMENTS	IDENTIFICATION
1 2 6	7	8 14	15 16 32	72	73 80
TRDES		VFD	18/PUNPR,1/1,1/1,2/3,14/0		
		FILCB	PUNPR,P*,PBUF		
PBUF		BSS	321		

Printed report file parameters:

Variable-length records
Block size--100 words
Two buffers
Mode--binary
File name--REPORT1
Location of file control block--REPRT1
File code--R1
GEFRC CALLs for this file: OPEN, CLOSE, PRINT

The file designator word and the file control block macro-instruction that define the printed report file are described below along with the buffer area allocation.

PROBLEM

PROGRAMMER DATE PAGE OF

LOCATION	E/O	OPERATION	ADDRESS, MODIFIER	COMMENTS	IDENTIFI-CATION
1 2 6	7	8 14	15 16	32 72	73 80
R PTDES		VFD	18/REPRT1,1/1,1/0,2/0,14/0		
		FILCB	REPRT1,R1,PRT1,PRT2,100,,,,,,,,,,,,REPORT1		
PRT1		BSS	101		
PRT2		BSS	101		

APPENDIX B. SUBROUTINE NAMES

This Appendix describes the GEFRC subroutines as having meaningful names without special heading characters (for example, OPEN,CLOSE). Names in this form can produce duplicate symbols when used within a language processor environment where reserved names do not exist (such as FORTRAN). Therefore, each GEFRC subroutine for which a CALL is described in this manual has at least two names:

1. The name as described.

2. A name whose first two characters are ".G"

All system software and all GEFRC internal CALLs use the name in the .G form.

Following is a three-column list of subroutine names. Column one contains the names as described herein. Column two is the .G name for the same subroutine. The third column gives still another name for many of the subroutines and it has a .GA as the first three characters. The purpose of this name is to provide for optional extra-processing routines. For example, suppose that program X must produce output which has been edited under one of several possible rules. For each editing rule there is a subroutine called .GAPUT which performs the editing and calls PUT to output the edited record. The proper edit routine can then be included at execution time and it will initiate loading of the PUT routine. However, for those executions in which no editing is to be performed, the user CALL .GAPUT will call the PUT routine directly since .GAPUT is just another name for PUT and it has not been otherwise defined.

BSREC	.GBSRC	.GABSR
BSTFM	.GBSFM	.GABSF
CLOSE	.GCLSE	.GACLS
COPY	.GCOPY	.GACOP
EPRINT	.GEPRN	-
FORCE	.GFRCE	.GAFRC
FSREC	.GFSRC	.GAFSR
FSTFM	.GFSFM	.GAFSF
GET	.GGET	.GAGET
GETBK	.GGTBK	.GAGTB
IOEDIT	.GEDIT	.GAEDI
OPEN	.GOPEN	.GAOPE
PRINT	.GPRNT	.GAPRN
PUNCH	.GPNCH	.GAPNC
PUT	.GPUT	.GAPUT
PUTBK	.GPTBK	.GAPTB
PUTSZ	.GPTSZ	.GAPTS
RDREC	.GRDRC	.GARDR
READ	.GREAD	.GAREA
RELSE	.GRLSE	.GARLS
REWND	.GRWND	.GARWN
SETIN	.GSTIN	-
SETOUT	.GSTOT	-
WAIT	.GWAIT	.GAWAI
WEF	.GWEF	.GAWEF
WRITE	.GWRIT	.GAWRI
WTREC	.GWTRC	.GAWTR

APPENDIX C. LIST OF SUBROUTINE NAMES

INDEX

The key-word index lists as entries each key-word subject arranged alphabetically by significant words in the titles of subject matter, program control words, language words, and the titles of both figures and tables.

To locate a subject, think of significant words that might describe the subject, then look in the index for the word being sought. When a title containing the word is found to be of interest, the full abstract can be found on the page number given to the right of the line. Titles are truncated to the left of the significant word; and those words truncated appear to the right of the title line, beginning with an asterisk.

DOCUMENT REVIEW SHEET

TITLE: GE-625/635 File and Record Control

CPB #: 1003E

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: _____

Please cut along this line

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
COMPUTER EQUIPMENT DEPARTMENT
13430 NORTH BLACK CANYON HIGHWAY
PHOENIX, ARIZONA - 85029

ATTENTION: DOCUMENTATION STANDARDS AND PUBLICATIONS B-90

FOLD

Progress Is Our Most Important Product

GENERAL ELECTRIC

INFORMATION SYSTEMS DIVISION

LITHO U.S.A.